GLOBE Data Architecture - Domains, Tables, Views, Procedures

August 7, 2002

Table of Contents

1 Domain Datatype..... 24
2 AOTdomain..... 32
3 AOTmethodDomain..... 33
4 AciditvDomain..... 33
5 AciditvMethodDomain..... 33
6 ActionTextDomain..... 34
7 ActivityRoleDomain..... 34
8 AddressLineDomain..... 34
9 AirTempDomain..... 34
10 AlkalConstantDomain..... 35
11 AlkalDropsDomain..... 35
12 AlkalRangeDomain..... 35
13 AlkalinitvDomain..... 35
14 AndersonDomain..... 36
15 AnnualGroupDomain..... 36
16 AppNumDomain..... 36
17 AugerDiamDomain..... 37
18 AvailabilitvDomain..... 37
19 BankMaterialDomain..... 37
20 BarometricPressureDomain..... 38
21 BedrockDomain..... 38
22 BioSquaresDomain..... 38
23 BiomassDomain..... 39
24 BulkDensitvDomain..... 39
25 CCcodeDomain..... 39
26 CableNbrDomain..... 39
27 CanopyCoverDomain..... 40
28 CarbonateDomain..... 40
29 ChannelIdDomain..... 40
30 CitvDomain..... 41
31 ClientTypeDomain..... 41
32 CloudCoverDomain..... 41
33 CloudPresenceDomain..... 42
34 Comment500Domain..... 42
35 CommentDomain..... 43
36 CompasAngleDomain..... 43
37 ConductivitvDomain..... 43
38 CongressBranchDomain..... 43
39 CongressDistrictDomain..... 44
40 CongressNbrDomain..... 44
41 ContactCodeDomain..... 44
42 ContactMethodDomain..... 45
43 ContainerIdDomain..... 45
44 Countrv Domain..... 45
45 CoverClassDomain..... 46
46 CoverTypeDomain..... 46
47 CreditTypeDomain..... 46
48 CreditsCodeDomain..... 47
49 DBmasterSiteDomain..... 47
50 DatabaseActionDomain..... 47
51 DateDomain..... 48
52 DateHourDomain..... 48
53 DatetimeDomain..... 48
54 DensitvMethodDomain..... 48
55 Description..... 49
56 DewPointDomain..... 49
57 DirectSourceDomain..... 49
58 DirectionAngleDomain..... 49
59 DissolvedOxyDomain..... 50
60 Dist500mLDomain..... 50

61 DocumentDomain.... 50
62 DuffLitterDomain.... 51
63 DyingStateDomain.... 51
64 ETconstantDomain.... 52
65 ElevationDomain.... 52
66 EmailDomain.... 52
67 EvalAgreeDomain.... 52
68 EvalCategoryDomain.... 53
69 EvalEffectiveDomain.... 53
70 EvalNumberDomain.... 54
71 EvalPreparedDomain.... 54
72 EvalRatingDomain.... 55
73 EvalScoreAvgDomain.... 55
74 FIPScountyCode.... 55
75 FertilityEstDomain.... 56
76 FlagDomain.... 56
77 FranContactTypeDomain.... 56
78 FranchiseDomain.... 57
79 FuelCountDomain.... 57
80 FullName2Domain.... 57
81 FullNameDomain.... 57
82 GDSdomain.... 58
83 GenusDomain.... 58
84 GlobePhaseDomain.... 58
85 GoalTextDomain.... 59
86 GradeLevelDomain.... 59
87 GradeListDomain.... 59
88 GrantNumberDomain.... 60
89 GrantScoreDomain.... 60
90 GrantStatusDomain.... 60
91 GrantTrainingDomain.... 61
92 GravOffsetDomain.... 61
93 GreeningCycleDomain.... 61
94 GreeningIdDomain.... 61
95 GreeningStateDomain.... 62
96 GroupCodeDomain.... 62
97 GroupIdDomain.... 63
98 HRnbrDomain.... 63
99 HazeEstDomain.... 63
100 HazeSensitivityDomain.... 64
101 HazeTransmissionDomain.... 64
102 HeightAngleDomain.... 64
103 HeightCmDomain.... 64
104 HorizonDomain.... 65
105 HourDomain.... 65
106 HumidityDomain.... 65
107 HumidityMethodDomain.... 66
108 HydroWeightDomain.... 66
109 IPpayerDomain.... 66
110 ImgDataSourceDomain.... 67
111 ImgDatumDomain.... 67
112 ImgLinesDomain.... 67
113 ImgLocatorDomain.... 67
114 ImgProjDistanceDomain.... 68
115 ImgProjUnitsDomain.... 68
116 ImgProjValueDomain.... 68
117 ImgProjectionDomain.... 69
118 ImgSamplesDomain.... 69
119 ImgSensorDomain.... 69
120 ImgStatusDomain.... 69
121 InfiltrateDepthDomain.... 70
122 InfiltrationDomain.... 70

123 InfiltrationMinDomain.... 70
124 InfiltrationSecDomain.... 71
125 InvestigationDomain.... 71
126 InvoiceNbrDomain.... 71
127 IssuesTextDomain.... 72
128 KitMfgNameDomain.... 72
129 LanguageCodeDomain.... 72
130 LanguageDomain.... 72
131 LatitudeDomain.... 73
132 LeafColorDomain.... 73
133 LeafIdDomain.... 73
134 LeafLengthDomain.... 74
135 LevelTypeDomain.... 74
136 LifeCycleDomain.... 74
137 LilacHeightDomain.... 78
138 LilacIdDomain.... 78
139 LilacTypeCodeDomain.... 78
140 LocationMethodDomain.... 79
141 LongitudeDomain.... 79
142 MUCcodeDomain.... 79
143 MUCdescriptionDomain.... 80
144 MUCnameDomain.... 80
145 MUCsourceDomain.... 80
146 MailFormatDomain.... 81
147 MailStatusDomain.... 81
148 MailTopicDomain.... 81
149 MailingNameDomain.... 81
150 MeasurementDomain.... 82
151 MinuteDomain.... 82
152 ModelNameDomain.... 83
153 MoistureEstDomain.... 83
154 MoistureProtocolDomain.... 83
155 MoneyDomain.... 84
156 MotivationTextDomain.... 84
157 MovingWaterDomain.... 84
158 NameDomain.... 85
159 NameTypeDomain.... 85
160 NbrItemsDomain.... 85
161 NbrSchoolsDomain.... 85
162 NitrateDomain.... 86
163 ObscuredDomain.... 86
164 ObstacleDistDomain.... 86
165 OrderNumberDomain.... 87
166 OrderTypeDomain.... 87
167 OzoneMethodDomain.... 87
168 OzonePPBDomain.... 88
169 PETdomain.... 88
170 PaperDocLocDomain.... 88
171 PartnerDomain.... 89
172 PercentDomain.... 89
173 PersonIdDomain.... 89
174 PhotoSubmittedDomain.... 89
175 PhotomVoltageDomain.... 90
176 PhotometerDomain.... 90
177 PlantedFlagDomain.... 90
178 PostActionItemDomain.... 91
179 PostalDomain.... 91
180 PreActionItemDomain.... 91
181 PrecipDaysDomain.... 92
182 PrefixDomain.... 92
183 PressureMethodDomain.... 92
184 PrevalenceDomain.... 92

185 ProductCodeDomain..... 93
186 Protocol ListDomain..... 93
187 QualityOfLangDomain..... 93
188 QuantityDomain..... 94
189 QueryStatusDomain..... 94
190 QuestAnswerDomain..... 94
191 QuestOrderDomain..... 95
192 QuestTargetDomain..... 95
193 QuestVersionDomain..... 95
194 QuestionTextDomain..... 96
195 RainfallDomain..... 96
196 RecruitCodeDomain..... 96
197 RecruitTextDomain..... 96
198 RegTypeCodeDomain..... 97
199 RegionCategoryDomain..... 97
200 RegionTypeDomain..... 98
201 RegisterCodeDomain..... 98
202 ResponsibilityDomain..... 98
203 RingDiamDomain..... 99
204 RockEstDomain..... 99
205 RockVolumeDomain..... 99
206 RoleDomain..... 99
207 RootEstDomain..... 100
208 RotClassDomain..... 100
209 SalinityDomain..... 100
210 SalinityMethodDomain..... 100
211 SaltOrFreshDomain..... 101
212 SalutationDomain..... 101
213 SchoolFundingDomain..... 101
214 SchoolIdDomain..... 102
215 SchoolNameDomain..... 102
216 SchoolTypeDomain..... 102
217 SeasonDomain..... 103
218 ServiceTypeDomain..... 103
219 SessionDomain..... 103
220 SiteCodeDomain..... 104
221 SiteDistanceDomain..... 104
222 SiteProximityDomain..... 104
223 SkyColorEstDomain..... 105
224 SlopeDomain..... 105
225 SnowDailyDomain..... 105
226 SnowTotalDomain..... 106
227 SoilColorDomain..... 106
228 SoilConsistenceEstDomain..... 106
229 SoilDepositDomain..... 107
230 SoilDepthDomain..... 107
231 SoilDryMthdDomain..... 107
232 SoilDryTimeDomain..... 108
233 SoilHorDepthDomain..... 108
234 SoilMeterDomain..... 108
235 SoilSourceDomain..... 109
236 SoilStructEstDomain..... 109
237 SoilTempDomain..... 109
238 SoilTempUsageDomain..... 110
239 SoilTextureDomain..... 110
240 SoilThermomDomain..... 110
241 SoilVolumeDomain..... 111
242 SoilWeightDomain..... 111
243 SolarDatetimeDomain..... 111
244 SolarDayDomain..... 112
245 SolarHourDomain..... 112
246 SolarMinDomain..... 112

247 SolarNoonDiffDomain.... 113
248 SourceStateDomain.... 113
249 SpeciesDomain.... 113
250 SpecificGravityDomain.... 114
251 SplitSessionDomain.... 114
252 StandingWaterDomain.... 114
253 State(US)Domain.... 115
254 StateProvDomain.... 115
255 StockNbrDomain.... 115
256 StrategyText.... 115
257 StudySiteDomain.... 116
258 SuffixDomain.... 116
259 SurfaceCoverDomain.... 116
260 SurfaceTypeDomain.... 117
261 TableNameDomain.... 117
262 TableTypeDomain.... 117
263 TaskIdDomain.... 118
264 TeacherRoleDomain.... 118
265 TeacherStatusDomain.... 118
266 TelPrefixUSDomain.... 118
267 TelephoneDomain.... 119
268 ThermomTypeDomain.... 119
269 TideTypeDomain.... 120
270 TimeZoneUSDomain.... 120
271 TimezoneDomain.... 120
272 TraceMissDomain.... 120
273 TrackingIdDomain.... 121
274 TrainerLevelDomain.... 122
275 TrainerRoleDomain.... 122
276 TrainerTypeDomain.... 123
277 TrainingOptionDomain.... 124
278 TransectIdDomain.... 125
279 TreeCircumDomain.... 125
280 TreeDiamDomain.... 126
281 TreeGrassShrubDomain.... 126
282 TreeHeightDomain.... 126
283 TreeIdDomain.... 126
284 TurbidityEstDomain.... 127
285 TurbiditySecchiDomain.... 127
286 TurbidityTubeDomain.... 127
287 URLdomain.... 128
288 UToffsetDomain.... 128
289 UrbanCodeDomain.... 128
290 VendorDomain.... 129
291 VisualDayDomain.... 129
292 WaterAreaDomain.... 129
293 WaterContentDomain.... 130
294 WaterDepthDomain.... 130
295 WaterDiffDomain.... 130
296 WaterLocationDomain.... 131
297 WaterSizeDomain.... 131
298 WaterTempDomain.... 131
299 WaterWidthDomain.... 131
300 WavelengthDomain.... 132
301 WindSpeedDomain.... 132
302 WkshpCategoryDomain.... 132
303 WorkshopDomain.... 133
304 YearDomain.... 133
305 ZipCodeUS5Domain.... 133
306 ZoomLevelDomain.... 134
307 Table ACIDITY_WS.... 135
308 Table ACIDITY_WS_CNT.... 135

309 Table ADMIN_ACTIVITY 135
310 Table AIR_TEMP 135
311 Table AIR_TEMP_CNT 135
312 Table ARCHIVEMESSAGELOG 135
313 Table ATMOSPHERE_SITE 135
314 Table ATMOSPHERE_SITE_CNT 136
315 Table BAROM_PRESSURE 137
316 Table BAROM_PRESSURE_CNT 137
317 Table BUDBURST_METADATA 137
318 Table BUDBURST_META_CNT 137
319 Table BULLETINS 137
320 Table CERTMERGE_DIRECTOR 137
321 Table CERTMERGE_RUNGONLY 137
322 Table CERTMERGE_TEACHER 137
323 Table CHECKLIST_ITEM 137
324 Table CLOUD_OBS 137
325 Table CLOUD_OBS_CNT 137
326 Table CONGRESS 137
327 Table CONGRESS_PEOPLE 137
328 Table CONTACT_LOG 137
329 Table CONTACT_TOPIC 137
330 Table COORDINATED_SITES 138
331 Table COORDINATIONS 138
332 Table COUNTRIES 138
333 Table COUNTRY_ATTENDANCE 138
334 Table COUNTRY_CODES 138
335 Table COUNTRY_CONTACTS 138
336 Table COUNTRY_IMAGES 139
337 Table COUNTRY_LINKS 139
338 Table COUNTRY_REGION 139
339 Table COVER_WS 139
340 Table CURRENT_TEMP_NONNOON 139
341 Table DAILY_DIXON_OBS 139
342 Table DAILY_DIXON_REP 139
343 Table DATA_ACTIVITY 139
344 Table DATA_ENTRY_OPTIONS 140
345 Table DB_MASTER_SITES 140
346 Table DEADLOG_WS 140
347 Table DENSITY_WS 141
348 Table DENSITY_WS_CNT 141
349 Table DIXON_SUMMARY_CNT 141
350 Table DIXON_SUMMARY_GROUPED 141
351 Table DIXON_SUMMARY_MC 141
352 Table DUFFLITTER_WS 141
353 Table EVALUATION_FORM 141
354 Table EVAL_QUESTIONS 141
355 Table EXPERIMENTLOG 141
356 Table FERTILITY_WS 141
357 Table FERTILITY_WS_CNT 141
358 Table FIREFUEL_CNT 141
359 Table FIREFUEL_SITE 141
360 Table FIREFUEL_TRANSECT 141
361 Table FLOWERLIST_BYSCHOOL 141
362 Table FLOWERSPECIES_COUNTS 141
363 Table FLOWERSPECIES_COUNTS_CNT 141
364 Table FOLLOWUP_TARGET 142
365 Table FRANCHISE_CONTACT 142
366 Table FRANCHISE_HOMEPAGE 142
367 Table GDS_MOISTCHG_30DAYS 142
368 Table GEOGRAPHIC_REGION 142
369 Table GLOBE1EXPERIMENTS 142
370 Table GLOBEDB_USERS 142

371 Table GLOBE_CAMPAIGN....142
372 Table GLOBE_CONCEPT....143
373 Table GLOBE_COUNTRY....143
374 Table GLOBE_FRANCHISE....144
375 Table GLOBE_MAILING....144
376 Table GLOBE_ORDER....144
377 Table GLOBE_PRODUCT....144
378 Table GLOBE_SCHOOL....145
379 Table GLOBE_STARS....147
380 Table GLOBE_TASK....147
381 Table GLOBE_TEACHER....147
382 Table GLOBE_TRAINER....149
383 Table GLOBE_VENDOR....149
384 Table GPS_METADATA....149
385 Table GPS_WORKSHEET....149
386 Table GRASSBIOM_WS....149
387 Table GRASSBIOM_WS_CNT....149
388 Table GRASS_BIOMETRY....149
389 Table GRASS_BIOMETRY_CNT....149
390 Table GRASS_TYPES....149
391 Table GREENDOWN_WS....149
392 Table GREENDOWN_WS_CNT....150
393 Table GREENING_METADATA....150
394 Table GREENING_METADATA_CNT....150
395 Table GREENUP_WS....150
396 Table GREENUP_WS_CNT....150
397 Table GROUP_USES_COUNT....150
398 Table HAZE....150
399 Table HAZE_CNT....150
400 Table HAZE_WS....150
401 Table HAZE_WS_CNT....151
402 Table HR_ATMOSADV_CURRENT....151
403 Table HR_ATMOSADV_PREVIOUS....151
404 Table HR_ATMOS_CURRENT....151
405 Table HR_ATMOS_PREVIOUS....151
406 Table HR_CLIMATE_CURRENT....151
407 Table HR_CLIMATE_PREVIOUS....151
408 Table HR_CLOUDS_CURRENT....151
409 Table HR_CLOUDS_PREVIOUS....151
410 Table HR_EARTHSYS_CURRENT....151
411 Table HR_EARTHSYS_PREVIOUS....151
412 Table HR_HYDRO_CURRENT....151
413 Table HR_HYDRO_PREVIOUS....151
414 Table HR_LCBIO_CURRENT....151
415 Table HR_LCBIO_PREVIOUS....151
416 Table HR_OVERALL_CURRENT....151
417 Table HR_OVERALL_PREVIOUS....151
418 Table HR_PANGLOBE_CURRENT....151
419 Table HR_PANGLOBE_PREVIOUS....151
420 Table HR_SOILCHAR_CURRENT....151
421 Table HR_SOILCHAR_PREVIOUS....151
422 Table HR_SOILMT_CURRENT....151
423 Table HR_SOILMT_PREVIOUS....151
424 Table HUMIDITY....151
425 Table HUMIDITY_CNT....152
426 Table HUMMINGBIRD_COUNTS....152
427 Table HUMMINGBIRD_COUNTS_CNT....152
428 Table HUMMINGBIRD_SITE....152
429 Table HUMMINGBIRD_SITE_CNT....152
430 Table HUMMINGBIRD_SPECIES....152
431 Table INFILTRATION_WS....152
432 Table INFILTRATION_WS_CNT....152

433 Table INTERNATIONAL LOG....152
434 Table INVESTIGATION CODES....152
435 Table LANDCOVER CODES....153
436 Table LAND COVER....153
437 Table LAND COVER CNT....153
438 Table LEAF COLOR....153
439 Table LEARNING STANDARD....153
440 Table LETTERMERGE DIRECTOR....153
441 Table LETTERMERGE RUNGONLY....153
442 Table LETTERMERGE TEACHER....153
443 Table LIFECYCLE CODES....153
444 Table LILAC METADATA....154
445 Table LILAC METADATA CNT....155
446 Table LILAC PHENOPHASE....155
447 Table LILAC PHENOPHASE CNT....155
448 Table LISTLEVEL....155
449 Table MAILING EFFECT....155
450 Table MAILING STATUS....155
451 Table MALE BEHAVIOR....155
452 Table MASSMAIL....155
453 Table MEASURE AT SITE....155
454 Table MEASURE CODES....155
455 Table MEASURE COUNTS....156
456 Table MEASURE GROUP....156
457 Table MESSAGELOG....156
458 Table MICRO CLIMATE....156
459 Table NESTING REPORT....157
460 Table NEW PARTNER CHECKLIST....157
461 Table NWS MEASUREMENTS....157
462 Table NWS SITES....157
463 Table ORDER ITEMS....158
464 Table OZONE....158
465 Table OZONE CNT....158
466 Table PALS INFO....158
467 Table PALS PROTOCOLS....158
468 Table PARTICLE96 WS....158
469 Table PARTICLE96 WS CNT....159
470 Table PARTICLE WS....159
471 Table PARTICLE WS CNT....159
472 Table PARTNER GOALS....159
473 Table PARTNER IMPLEMENTATION PLAN....159
474 Table PARTNER LIAISON....159
475 Table PARTNER SCHOOLTYPES....159
476 Table PARTNER STRATEGIES....159
477 Table PARTNER TO AFFILIATE....159
478 Table PET CHART....159
479 Table PHENGARDEN CONTENT....159
480 Table PHENGARDEN SITE....159
481 Table PHENOLOGICAL GARDEN....159
482 Table PHENOLOGY CNT....160
483 Table PHENOLOGY SHRUB....160
484 Table PHENOLOGY SITE....160
485 Table PHENOLOGY SITE CNT....160
486 Table PHENOLOGY WORKUP....160
487 Table PHENOLOGY WS....161
488 Table PHENOLOGY WS CNT....161
489 Table PHOTOMETER CALIB....161
490 Table PHOTO ENVELOPE....162
491 Table PHOTO INFO....162
492 Table POST CHECKLIST....162
493 Table PRECIP RAIN....162
494 Table PRECIP RAIN CNT....162

495 Table PRECIP_SOLID....162
496 Table PRECIP_SOLID_CNT....163
497 Table PRECIP_SOLID_WS....163
498 Table PRE_CHECKLIST....163
499 Table PROTOCOL_TO_CONTACT....163
500 Table PROTOCOL_USES_GROUPS....163
501 Table REGISTRATION....163
502 Table REGISTRATION_FOR_PROTOCOL....163
503 Table S2S_ACTIVITY....163
504 Table S2S_PARTICIPANT....164
505 Table SALINITY_LOOKUP....164
506 Table SCHLTCHR_MINCWDATE....164
507 Table SCHOOL7_A....164
508 Table SCHOOL7_CGTS....164
509 Table SCHOOL7_CP....164
510 Table SCHOOL7_IO....164
511 Table SCHOOL7_M....164
512 Table SCHOOL7_N_CGTS....164
513 Table SCHOOLFLAGS....164
514 Table SCHOOLGRADE_CODES....165
515 Table SCHOOLZOOM....165
516 Table SCHOOL_CONGRESS....165
517 Table SCHOOL_EQUIP....165
518 Table SCHOOL_GRANTINFO....165
519 Table SCHOOL_HTTPINFO....165
520 Table SCHOOL_MINCWDATE....166
521 Table SCHOOL_NBRCGTS....166
522 Table SCHOOL_TRAINEDSTAFF....166
523 Table SCHOOL_ZOOMLEVEL....166
524 Table SESSION_PROTOCOLS....166
525 Table SESSION_TO_PARTNER....166
526 Table SITE_CODES....166
527 Table SITE_LOCATION....166
528 Table SITE_LOCATION_CNT....167
529 Table SITE_OBSTRUCTION....167
530 Table SITE_PHOTO....167
531 Table SITE_PHOTO_CNT....167
532 Table SITE_PROXIMITY....167
533 Table SITE_PROXIMITY_CNT....168
534 Table SKY_OBSCURED....168
535 Table SKY_OBSCURED_CNT....168
536 Table SMGRAV_WS....169
537 Table SMGRAV_WS_CNT....169
538 Table SMGYPS_WS....169
539 Table SMGYPS_WS_CNT....169
540 Table SNAP$_PHENOLOGY_WORKUP....169
541 Table SNOWPACK_WATER....169
542 Table SOILCHAR_SITE....169
543 Table SOILCHAR_SITE_CNT....170
544 Table SOILMOISTURE_SITE....170
545 Table SOILMOISTURE_SITE_CNT....171
546 Table SOILTEMP_DAILYINFO....172
547 Table SOIL_CHAR....172
548 Table SOIL_CHAR_CNT....172
549 Table SOIL_INFILTRATION....172
550 Table SOIL_INFILTRATION_CNT....173
551 Table SOIL_MOISTURE....173
552 Table SOIL_MOISTURE_CNT....173
553 Table SOIL_TEMP....173
554 Table SOIL_TEMP_CNT....173
555 Table STAFF_PROTOCOLS....173
556 Table STATE_LEA_CODE....173

557 Table STATE_PROV ... 173
558 Table STATE_PROV_REGION ... 174
559 Table STRATEGY_ITEM ... 174
560 Table STUDY_SITE ... 174
561 Table SURFACEWATER_SITE ... 174
562 Table SURFACEWATER_SITE_CNT ... 176
563 Table SURFACE_WATER ... 176
564 Table SURFACE_WATER_CNT ... 176
565 Table SW_CALIBRATION ... 176
566 Table T2T_MESSAGE ... 176
567 Table TASK_TO_CONCEPT ... 176
568 Table TASK_TO_LEARNING ... 177
569 Table TEACHER_AGT ... 177
570 Table TEACHER_EVALUATION ... 177
571 Table TEACHER_TO_SCHOOL ... 177
572 Table TEMPERATURE_LOGGER ... 177
573 Table TEMPERATURE_LOGGER_CNT ... 177
574 Table TEMP_EXTREMES ... 177
575 Table THERMOMCALIB_WS ... 177
576 Table THERMOM_CALIB ... 177
577 Table THERMOM_RESET ... 177
578 Table TM_IMAGE ... 177
579 Table TRAINER_EVALUATION ... 178
580 Table TRAINER_LANGUAGES ... 178
581 Table TRAINER_PROTOCOLS ... 178
582 Table TRAINER_TRAINING ... 178
583 Table TRAINING_TARGET ... 178
584 Table TREEBIOM_WS ... 178
585 Table TREEBIOM_WS_CNT ... 178
586 Table TREE_BIOMETRY ... 178
587 Table TREE_BIOMETRY_CNT ... 178
588 Table TREE_TYPES ... 178
589 Table TURBIDITY_WS ... 178
590 Table TURBIDITY_WS_CNT ... 178
591 Table URBANIZATION_CODES ... 179
592 Table USREGIONS ... 179
593 Table WEEKLY_ADDDROP_RPT ... 179
594 Table WORKSHOP ... 179
595 Table WORKSHOP_CONTACTS ... 180
596 Table WORKSHOP_DROPADD ... 180
597 Table WORKSHOP_SESSION ... 180
598 Table WORKSHOP_STAFF ... 181
599 Table WWW_USERS ... 181
600 Table ZIP_INFO ... 181
601 View ACIDITY_WS_DR1 ... 182
602 View ACIDITY_WS_DR2 ... 182
603 View ACIDITY_WS_DU1 ... 182
604 View ACIDITY_WS_DU2 ... 182
605 View ACIDITY_WS_MC1 ... 182
606 View ACIDITY_WS_MC2 ... 182
607 View ACIDITY_WS_RPRT_VU ... 182
608 View ACIDITY_WS_VU1 ... 182
609 View ACIDITY_WS_VU2 ... 183
610 View AIR_TEMP_DR1 ... 183
611 View AIR_TEMP_DR2 ... 183
612 View AIR_TEMP_DU1 ... 183
613 View AIR_TEMP_DU2 ... 183
614 View AIR_TEMP_MC1 ... 183
615 View AIR_TEMP_MC2 ... 183
616 View AIR_TEMP_RPRT_VU ... 183
617 View AIR_TEMP_VU1 ... 183
618 View AIR_TEMP_VU2 ... 184

619 View ATMOSADV_HR1 184
620 View ATMOSADV_HR2 184
621 View ATMOSADV_HR3 184
622 View ATMOSPHERE_SITE_DR1 185
623 View ATMOSPHERE_SITE_DR2 185
624 View ATMOSPHERE_SITE_DU1 185
625 View ATMOSPHERE_SITE_DU2 185
626 View ATMOSPHERE_SITE_MC1 185
627 View ATMOSPHERE_SITE_MC2 185
628 View ATMOSPHERE_SITE_RPRT_VU 185
629 View ATMOSPHERE_SITE_VU1 186
630 View ATMOSPHERE_SITE_VU2 186
631 View ATMOS_HR1 186
632 View ATMOS_HR2 187
633 View ATMOS_HR3 187
634 View ATMOS_HRDX 187
635 View ATMOS_PRECIP_HR1 188
636 View ATMOS_PRECIP_HR2 188
637 View ATMOS_PR_HR1 188
638 View ATMOS_PS_HR1 188
639 View ATMOS_TEMP_HR1 188
640 View ATMOS_TEMP_HR2 189
641 View AciditvRain_HR1 189
642 View AciditvSnow_HR1 189
643 View Aciditv_HR2 190
644 View Aciditv_HR3 190
645 View Acidity_HR4 191
646 View Aerosol_HR1 191
647 View Aerosol_HR2 191
648 View Aerosol_HR3 191
649 View Aerosol_HR4 192
650 View BAROM_PRESSURE_VU1 192
651 View BAROM_PRESSURE_VU2 192
652 View BBatmosLink_HR0 193
653 View BBatmosLink_HR1 193
654 View BBrain_HR0 193
655 View BBsnow_HR0 193
656 View BBtemp_HR1 194
657 View BBzPrecip_HR1 194
658 View BUDBURST_META_DR1 194
659 View BUDBURST_META_DR2 194
660 View BUDBURST_META_DU1 194
661 View BUDBURST_META_DU2 194
662 View BUDBURST_META_VU1 195
663 View BUDBURST_META_VU2 195
664 View BudBurst_HR1 195
665 View BudBurst_HR2 195
666 View CERTMERGE_DIRECTOR_VU 197
667 View CERTMERGE_RUNGONLY_VU 197
668 View CERTMERGE_TEACHER_VU 198
669 View CLIMATE_1_HR1 198
670 View CLIMATE_1_HR2 199
671 View CLIMATE_2_HR1 200
672 View CLIMATE_2_HR2 201
673 View CLIMATE_3_HR1 202
674 View CLIMATE_3_HR2 203
675 View CLIMATE_4_HR1 204
676 View CLIMATE_4_HR2 205
677 View CLIMATE_4_HR3_2parms 205
678 View CLIMATE_4_HR3_4parms 206
679 View CLIMATE_4_HR3_6parms 208
680 View CLIMATE_4_HR4 210

681 View CLIMATE_4_HR5 210
682 View CLIMATE_5_HR1 211
683 View CLIMATE_5_HR2 212
684 View CLIMATE_6_HR1 213
685 View CLIMATE_6_HR2 214
686 View CLIMATE_7_HR1 215
687 View CLIMATE_7_HR2 216
688 View CLIMATE_7_HR3 216
689 View CLIMATE_7_HR4 217
690 View CLIMATE_7_HR5 218
691 View CLIMATE_7_HR6 219
692 View CLIMATE_8_HR1 220
693 View CLIMATE_ALL_HR0 220
694 View CLIMATE_ALL_HR1 220
695 View CLIMATE_ALL_HR2 220
696 View CLOUDS_HR0 221
697 View CLOUDS_HR1 221
698 View CLOUDS_HR2 221
699 View CLOUDS_HR3 221
700 View CLOUDS_HR4 222
701 View CLOUD_OBS_DR1 222
702 View CLOUD_OBS_DR2 222
703 View CLOUD_OBS_DU1 222
704 View CLOUD_OBS_DU2 222
705 View CLOUD_OBS_MC1 222
706 View CLOUD_OBS_MC2 222
707 View CLOUD_OBS_NOON 222
708 View CLOUD_OBS_RPRT_VU 223
709 View CLOUD_OBS_VU1 223
710 View CLOUD_OBS_VU2 223
711 View CURRENT_TEMP_NONNOON_VU1 224
712 View CURRENT_TEMP_NONNOON_VU2 224
713 View Canopy_HR1 224
714 View Canopy_HR2 224
715 View Canopy_HR3 226
716 View CloudCover_HR1 226
717 View CloudCover_HR2 226
718 View DAILY_DIXON_OBS_VU 227
719 View DAILY_DIXON_REP_VU 228
720 View DENSITY_WS_DR1 229
721 View DENSITY_WS_DR2 229
722 View DENSITY_WS_DU1 230
723 View DENSITY_WS_DU2 230
724 View DENSITY_WS_MC1 230
725 View DENSITY_WS_MC2 230
726 View DENSITY_WS_RPRT_VU 230
727 View DENSITY_WS_VU1 230
728 View DENSITY_WS_VU2 230
729 View DISSOLVED_OXYGEN_VU 231
730 View DIURNAL_LOG_HR0 231
731 View DIURNAL_LOG_HR1 231
732 View DIURNAL_LOG_HR2 231
733 View DIURNAL_TEMP_HR0 232
734 View DIURNAL_TEMP_HR1 232
735 View DIURNAL_TEMP_HR2 233
736 View DIXON_SUMMARY_VU1 233
737 View DIXON_SUMMARY_VU2 237
738 View DIXON_SUMMARY_VU3 237
739 View EARTH_SYSTEM_1_HR0 238
740 View EARTH_SYSTEM_1_HR1 238
741 View EARTH_SYSTEM_2_HR0 238
742 View EARTH_SYSTEM_2_HR1 238

743 View EARTH_SYSTEM_3_HR0....238
744 View EARTH_SYSTEM_3_HR1....239
745 View EARTH_SYSTEM_4_HR0....239
746 View EARTH_SYSTEM_4_HR1....239
747 View EARTH_SYSTEM_5_HR0....239
748 View EARTH_SYSTEM_5_HR1....239
749 View EARTH_SYSTEM_6_HR1....239
750 View EARTH_SYSTEM_ALL_HR1....240
751 View EARTH_SYSTEM_ALL_HR2....240
752 View FERTILITY_WS_DR1....240
753 View FERTILITY_WS_DR2....240
754 View FERTILITY_WS_DU1....241
755 View FERTILITY_WS_DU2....241
756 View FERTILITY_WS_MC1....241
757 View FERTILITY_WS_MC2....241
758 View FERTILITY_WS_RPRT_VU....241
759 View FERTILITY_WS_VU1....241
760 View FERTILITY_WS_VU2....241
761 View FIPS_TO_COUNTY....242
762 View FIRECOVER_SUMMARY....242
763 View FIREFUEL_SUMMARY....242
764 View FIREFUEL_VU1....242
765 View FIREFUEL_VU2....242
766 View FLOWERSPECIES_COUNTS_VU1....242
767 View FLOWERSPECIES_COUNTS_VU2....242
768 View GDS_MOISTCHG_30DAYS_VU....242
769 View GDS_PRECIP_30DAYS_VU....243
770 View GDS_PRECIP_DAILY_VU....243
771 View GDS_SNOWMELT_30DAYS_VU....243
772 View GLOBE_CAMPAIGN_LIST....243
773 View GLOBE_PI_SCIENTISTS....243
774 View GRASSBIOM_HR1....244
775 View GRASSBIOM_HR1_LocP....244
776 View GRASSBIOM_HR2....244
777 View GRASSBIOM_HR3....246
778 View GRASSBIOM_WS_DR1....246
779 View GRASSBIOM_WS_DR2....246
780 View GRASSBIOM_WS_DU1....246
781 View GRASSBIOM_WS_DU2....246
782 View GRASSBIOM_WS_MC1....246
783 View GRASSBIOM_WS_MC2....246
784 View GRASSBIOM_WS_RPRT_VU....247
785 View GRASSBIOM_WS_VU1....247
786 View GRASSBIOM_WS_VU2....247
787 View GRASS_BIOMETRY_DR1....247
788 View GRASS_BIOMETRY_DR2....247
789 View GRASS_BIOMETRY_DU1....247
790 View GRASS_BIOMETRY_DU2....248
791 View GRASS_BIOMETRY_MC1....248
792 View GRASS_BIOMETRY_MC2....248
793 View GRASS_BIOMETRY_RPRT_VU....248
794 View GRASS_BIOMETRY_VU1....248
795 View GRASS_BIOMETRY_VU2....248
796 View GREENDOWN_COLOR....248
797 View GREENDOWN_FALLENLEAF....249
798 View GREENDOWN_STATES....249
799 View GREENDOWN_WS_VU1....250
800 View GREENDOWN_WS_VU2....250
801 View GREENING_METADATA_VU1....250
802 View GREENING_METADATA_VU2....251
803 View GREENUP_GROWTH....251
804 View GREENUP_LOSTLEAF....251

805 View GREENUP_MAXLENGTH....252
806 View GREENUP_STATES....252
807 View GREENUP_SUMMARY....253
808 View GREENUP_WS_VU1....253
809 View GREENUP_WS_VU2....254
810 View GrassTrees_HR1....254
811 View GrassTrees_HR2....254
812 View Grass_HR0....255
813 View GreenDown_HR1....256
814 View GreenUp_HR1....256
815 View HAZE_VISUALIZATION....257
816 View HAZE_VU1....257
817 View HAZE_VU2....258
818 View HAZE_WS_CALC....258
819 View HAZE_WS_SUMMARY....258
820 View HAZE_WS_VU1....259
821 View HAZE_WS_VU2....259
822 View HONOR_ROLL_OVERALL....259
823 View HONOR_ROLL_SUMMARY_VU1....259
824 View HONOR_ROLL_SUMMARY_VU2....260
825 View HONOR_ROLL_SUMMARY_VU3....260
826 View HONOR_ROLL_SUMMARY_VU4....260
827 View HR_ALLCNT_SUMS....260
828 View HR_CATEGORIES_ALL....261
829 View HR_CATEGORIES_ALL_CNT....262
830 View HR_CATEGORIES_CERT_MONTHS....262
831 View HR_CATEGORIES_DISTINCT....263
832 View HR_CATEGORIES_NEW....263
833 View HR_CATEGORIES_NEW_MONTHS....264
834 View HR_CATEGORIES_PREV_MONTHS....265
835 View HR_CATEGORIES_YEAR_CNT....265
836 View HR_CATEGORIES_YEAR_MONTHS....265
837 View HR_CERTIFICATES....266
838 View HR_CERTIFICATES_APRIL....266
839 View HR_CERTIFICATES_NEW....266
840 View HR_CERTIFICATES_PREV....266
841 View HR_CERTIFICATE_INFO_VU1....267
842 View HR_CERTIFICATE_INFO_VU2....268
843 View HR_CERTIFICATE_INFO_VU3....268
844 View HR_CERTIFICATE_TEACHERS....269
845 View HR_LETTER_INFO_VU1....269
846 View HR_LETTER_INFO_VU2....270
847 View HR_LETTER_INFO_VU3....271
848 View HR_LETTER_INFO_VU4....271
849 View HR_LTRSCHOOLS....272
850 View HR_LTRSCHOOL_TEACHERS....272
851 View HR_OVERALL_VU....272
852 View HR_RUNGCHG....273
853 View HR_RUNGONLY_CERTIFICATE....273
854 View HR_RUNGONLY_LETTER....273
855 View HUMIDITY_VU1....273
856 View HUMIDITY_VU2....274
857 View HUMMINGBIRD_COUNTS_VU1....274
858 View HUMMINGBIRD_COUNTS_VU2....274
859 View HUMMINGBIRD_FIRST_LAST....274
860 View HUMMINGBIRD_FREQUENCIES....274
861 View HUMMINGBIRD_SITE_VU1....274
862 View HUMMINGBIRD_SITE_VU2....275
863 View Humidity_HR1....275
864 View Humidity_HR2....275
865 View Humidity_HR3....275
866 View Humidity_HR4....276

867 View INFILTRATION_WS_DR1 276
868 View INFILTRATION_WS_DR2 276
869 View INFILTRATION_WS_DU1 276
870 View INFILTRATION_WS_DU2 276
871 View INFILTRATION_WS_MC1 276
872 View INFILTRATION_WS_MC2 276
873 View INFILTRATION_WS_RPRT_VU 277
874 View INFILTRATION_WS_VU1 277
875 View INFILTRATION_WS_VU2 277
876 View LAND_COVER_DR1 277
877 View LAND_COVER_DR2 277
878 View LAND_COVER_DU1 277
879 View LAND_COVER_DU2 278
880 View LAND_COVER_MC1 278
881 View LAND_COVER_MC2 278
882 View LAND_COVER_RPRT_VU 278
883 View LAND_COVER_VU1 278
884 View LAND_COVER_VU2 279
885 View LCBIO_HRDX 279
886 View LCBIO_OVERALL_HR1 279
887 View LCBIO_OVERALL_HR2 279
888 View LCBIO_OVERALL_HR3 279
889 View LCL_10_HR3 279
890 View LCL_5_HR3 280
891 View LCL_HR1 280
892 View LCL_HR1_LocP 280
893 View LCL_HR2 280
894 View LETTERMERGE_DIRECTOR_VU 282
895 View LETTERMERGE_RUNGONLY_VU 282
896 View LETTERMERGE_TEACHER_VU 283
897 View LILAC_METADATA_VU1 284
898 View LILAC_METADATA_VU2 284
899 View LILAC_PHENOPHASE_VU 284
900 View LILAC_PHENOPHASE_VU1 284
901 View LILAC_PHENOPHASE_VU2 285
902 View LILAC_SUMMARY 285
903 View LOGGER_HR0 285
904 View Lilac_HR1 285
905 View MCODE_COUNTS_VU1 286
906 View MCODE_COUNTS_VU2 286
907 View Nitrate_HR1 286
908 View Nitrate_HR2 287
909 View OZONE_VU1 287
910 View OZONE_VU2 288
911 View Ozone_HR1 288
912 View Ozone_HR2 288
913 View Ozone_HR3 288
914 View Ozone_HR4 289
915 View PAN_GLOBE_HR1 289
916 View PAN_GLOBE_HR2 289
917 View PARTICLE96_WS_DR1 289
918 View PARTICLE96_WS_DR2 289
919 View PARTICLE96_WS_DU1 289
920 View PARTICLE96_WS_DU2 289
921 View PARTICLE96_WS_MC1 290
922 View PARTICLE96_WS_MC2 290
923 View PARTICLE96_WS_RPRT_VU 290
924 View PARTICLE96_WS_VU1 290
925 View PARTICLE96_WS_VU2 290
926 View PARTICLE_WS_DR1 290
927 View PARTICLE_WS_DR2 290
928 View PARTICLE_WS_DU1 291

929 View PARTICLE_WS_DU2 291
930 View PARTICLE_WS_MC1 291
931 View PARTICLE_WS_MC2 291
932 View PARTICLE_WS_RPRT_VU 291
933 View PARTICLE_WS_VU1 291
934 View PARTICLE_WS_VU2 291
935 View PG_ATMOS_HR1 292
936 View PG_ATMOS_HR2 292
937 View PG_DissolvedOxy_HR1 292
938 View PG_DissolvedOxy_HR2 293
939 View PG_Grass_HR1 293
940 View PG_Grass_HR2 293
941 View PG_GravMoist12_HR0 295
942 View PG_GravMoist12_HR1 295
943 View PG_GravMoist12_HR2 295
944 View PG_HYDRO_HR1 297
945 View PG_HYDRO_HR2 297
946 View PG_Infiltration_HR1 297
947 View PG_Infiltration_HR2 297
948 View PG_LAND_HR1 299
949 View PG_LAND_HR2 299
950 View PG_NearSurf_HR1 299
951 View PG_NearSurf_HR2 300
952 View PG_Nitrate_HR1 301
953 View PG_Precip_HR1 301
954 View PG_Precip_HR2 302
955 View PG_SOIL_HR1 302
956 View PG_SOIL_HR2 303
957 View PG_acidity_HR1 303
958 View PG_acidity_HR2 303
959 View PG_alkalinity_HR1 303
960 View PG_alkalinity_HR2 304
961 View PG_conductivity_HR1 304
962 View PG_conductivity_HR2 304
963 View PG_nitrate_HR2 305
964 View PG_salinity_HR1 305
965 View PG_salinity_HR2 306
966 View PG_temp_HR1 306
967 View PG_temp_HR2 306
968 View PG_turbidity_HR1 307
969 View PG_turbidity_HR2 307
970 View PHENOLOGY 307
971 View PHENOLOGY_DR1 308
972 View PHENOLOGY_DR2 308
973 View PHENOLOGY_DU1 308
974 View PHENOLOGY_DU2 308
975 View PHENOLOGY_MC1 308
976 View PHENOLOGY_MC2 308
977 View PHENOLOGY_RPRT_VU 308
978 View PHENOLOGY_SITE_DR1 309
979 View PHENOLOGY_SITE_DR2 309
980 View PHENOLOGY_SITE_DU1 309
981 View PHENOLOGY_SITE_DU2 309
982 View PHENOLOGY_SITE_MC1 309
983 View PHENOLOGY_SITE_MC2 309
984 View PHENOLOGY_SITE_RPRT_VU 309
985 View PHENOLOGY_SITE_VU1 309
986 View PHENOLOGY_SITE_VU2 310
987 View PHENOLOGY_VU1 310
988 View PHENOLOGY_VU2 310
989 View PHENOLOGY_WS_DR1 310
990 View PHENOLOGY_WS_DR2 310

991 View PHENOLOGY_WS_DU1 310
992 View PHENOLOGY_WS_DU2 310
993 View PHENOLOGY_WS_MC1 310
994 View PHENOLOGY_WS_MC2 311
995 View PHENOLOGY_WS_RPRT_VU 311
996 View PHENOLOGY_WS_VU1 311
997 View PHENOLOGY_WS_VU2 311
998 View PHN_JoinFour_VU 311
999 View PHN_JoinThree_VU 312
1000 View PH_SOIL_HR1 312
1001 View PH_SOIL_HR2 312
1002 View PH_SOIL_HR3 314
1003 View PH_SURFWATER_HR1 314
1004 View PH_SURFWATER_HR2 314
1005 View PH_SURFWATER_HR3 314
1006 View POSTCHECKLIST_VIEW 315
1007 View PRECHECKLIST_VIEW 315
1008 View PRECIP_RAIN_DR1 315
1009 View PRECIP_RAIN_DR2 315
1010 View PRECIP_RAIN_DU1 315
1011 View PRECIP_RAIN_DU2 315
1012 View PRECIP_RAIN_MC1 316
1013 View PRECIP_RAIN_MC2 316
1014 View PRECIP_RAIN_RPRT_VU 316
1015 View PRECIP_RAIN_VU1 316
1016 View PRECIP_RAIN_VU2 316
1017 View PRECIP_SOLID_DR1 317
1018 View PRECIP_SOLID_DR2 317
1019 View PRECIP_SOLID_DU1 317
1020 View PRECIP_SOLID_DU2 317
1021 View PRECIP_SOLID_MC1 317
1022 View PRECIP_SOLID_MC2 317
1023 View PRECIP_SOLID_RPRT_VU 317
1024 View PRECIP_SOLID_VU1 318
1025 View PRECIP_SOLID_VU2 318
1026 View PROTOCOL_HIERARCHY 318
1027 View Phenology_HR1 318
1028 View Phenology_HR2 319
1029 View Phenology_HR3 319
1030 View Phenology_HR4 319
1031 View Phenology_HR5 319
1032 View Phenology_HR6 320
1033 View Phenology_HR7 320
1034 View Phenology_HR8 320
1035 View REACH250_EarlyDate 322
1036 View REACH250_LateDate 322
1037 View SCHOOL_OBS_RATE 322
1038 View SCHOOL_REPORT_RATE 323
1039 View SITE_LOCATION_DR1 323
1040 View SITE_LOCATION_DR2 323
1041 View SITE_LOCATION_DU2 323
1042 View SITE_LOCATION_MC1 323
1043 View SITE_LOCATION_MC2 323
1044 View SITE_LOCATION_RPRT_VU 323
1045 View SITE_LOCATION_VU1 324
1046 View SITE_LOCATION_VU2 324
1047 View SITE_PHOTO_DR1 324
1048 View SITE_PHOTO_DR2 324
1049 View SITE_PHOTO_DU1 324
1050 View SITE_PHOTO_DU2 324
1051 View SITE_PHOTO_MC1 324
1052 View SITE_PHOTO_MC2 325

1053 View SITE_PHOTO_VU1 325
1054 View SITE_PHOTO_VU2 325
1055 View SITE_PROXIMITY_DR1 325
1056 View SITE_PROXIMITY_DR2 325
1057 View SITE_PROXIMITY_DU2 326
1058 View SITE_PROXIMITY_MC1 326
1059 View SITE_PROXIMITY_MC2 326
1060 View SITE_PROXIMITY_RPRT_VU 326
1061 View SITE_PROXIMITY_VU1 326
1062 View SITE_PROXIMITY_VU2 326
1063 View SKY_OBSCURED_VU1 326
1064 View SKY_OBSCURED_VU2 327
1065 View SMGRAV_WS_DR1 327
1066 View SMGRAV_WS_DR2 327
1067 View SMGRAV_WS_DU1 327
1068 View SMGRAV_WS_DU2 327
1069 View SMGRAV_WS_MC1 327
1070 View SMGRAV_WS_MC2 328
1071 View SMGRAV_WS_RPRT_VU 328
1072 View SMGRAV_WS_VU1 328
1073 View SMGRAV_WS_VU2 328
1074 View SMGYPS_WS_DR1 328
1075 View SMGYPS_WS_DR2 328
1076 View SMGYPS_WS_DU1 328
1077 View SMGYPS_WS_DU2 328
1078 View SMGYPS_WS_MC1 328
1079 View SMGYPS_WS_MC2 329
1080 View SMGYPS_WS_RPRT_VU 329
1081 View SMGYPS_WS_VU1 329
1082 View SMGYPS_WS_VU2 329
1083 View SMT_Block_HR1 329
1084 View SMT_Block_HR2 329
1085 View SMT_Diurnal_HR1 330
1086 View SMT_Diurnal_HR2 330
1087 View SMT_Diurnal_HR3 330
1088 View SMT_GravHole_HR1 331
1089 View SMT_GravHole_HR2 331
1090 View SMT_GravStarInt_HR1 332
1091 View SMT_GravStarInt_HR2 332
1092 View SMT_Logger_HR1 333
1093 View SMT_Logger_HR2 333
1094 View SMT_Moisture_HR1 333
1095 View SMT_Moisture_HR2 334
1096 View SMT_Temp_HR1 334
1097 View SMT_Temp_HR2 334
1098 View SMT_XAllThree_HR1 335
1099 View SMT_XAllThree_HR2 335
1100 View SMT_XAllThree_HR3 335
1101 View SOILCHAR_HR1 336
1102 View SOILCHAR_HR2 336
1103 View SOILCHAR_HR2_Loc 337
1104 View SOILCHAR_HR3 337
1105 View SOILCHAR_HR4 338
1106 View SOILCHAR_HR5 339
1107 View SOILCHAR_SITE_DR1 339
1108 View SOILCHAR_SITE_DR2 339
1109 View SOILCHAR_SITE_DU1 339
1110 View SOILCHAR_SITE_DU2 339
1111 View SOILCHAR_SITE_MC1 339
1112 View SOILCHAR_SITE_MC2 339
1113 View SOILCHAR_SITE_RPRT_VU 340
1114 View SOILCHAR_SITE_VU1 340

1115 View SOILCHAR_SITE_VU2 340
1116 View SOILINFIL_HR1 340
1117 View SOILINFIL_HR2 340
1118 View SOILINFIL_HR3 342
1119 View SOILMOISTURE_SITE_DR1 342
1120 View SOILMOISTURE_SITE_DR2 342
1121 View SOILMOISTURE_SITE_DU1 342
1122 View SOILMOISTURE_SITE_DU2 342
1123 View SOILMOISTURE_SITE_MC1 343
1124 View SOILMOISTURE_SITE_MC2 343
1125 View SOILMOISTURE_SITE_RPRT_VU 343
1126 View SOILMOISTURE_SITE_VU1 343
1127 View SOILMOISTURE_SITE_VU2 343
1128 View SOILTEMP_DAILYAVG 343
1129 View SOILTEMP_DIURNAL 344
1130 View SOIL_CHAR_DR1 344
1131 View SOIL_CHAR_DR2 344
1132 View SOIL_CHAR_DU1 344
1133 View SOIL_CHAR_DU2 344
1134 View SOIL_CHAR_MC1 344
1135 View SOIL_CHAR_MC2 344
1136 View SOIL_CHAR_RPRT_VU 344
1137 View SOIL_CHAR_VU1 345
1138 View SOIL_CHAR_VU2 345
1139 View SOIL_INFILTRATION_DR1 345
1140 View SOIL_INFILTRATION_DR2 345
1141 View SOIL_INFILTRATION_DU1 345
1142 View SOIL_INFILTRATION_DU2 345
1143 View SOIL_INFILTRATION_MC1 345
1144 View SOIL_INFILTRATION_MC2 346
1145 View SOIL_INFILTRATION_RPRT_VU 346
1146 View SOIL_INFILTRATION_VU1 346
1147 View SOIL_INFILTRATION_VU2 346
1148 View SOIL_MOISTURE_RPRT_VU 346
1149 View SOIL_MOISTURE_VU1 346
1150 View SOIL_TEMP_DR1 347
1151 View SOIL_TEMP_DR2 347
1152 View SOIL_TEMP_DU1 347
1153 View SOIL_TEMP_DU2 347
1154 View SOIL_TEMP_HR0 347
1155 View SOIL_TEMP_HR1 347
1156 View SOIL_TEMP_HR2 348
1157 View SOIL_TEMP_MC1 348
1158 View SOIL_TEMP_MC2 348
1159 View SOIL_TEMP_RPRT_VU 348
1160 View SOIL_TEMP_VU1 349
1161 View SOIL_TEMP_VU2 349
1162 View SURFACEWATER_SITE_DR1 349
1163 View SURFACEWATER_SITE_DR2 349
1164 View SURFACEWATER_SITE_DU1 349
1165 View SURFACEWATER_SITE_DU2 349
1166 View SURFACEWATER_SITE_MC1 349
1167 View SURFACEWATER_SITE_MC2 350
1168 View SURFACEWATER_SITE_RPRT_VU 350
1169 View SURFACEWATER_SITE_VU1 350
1170 View SURFACEWATER_SITE_VU2 350
1171 View SURFACE_WATER_DR1 351
1172 View SURFACE_WATER_DR2 351
1173 View SURFACE_WATER_DU1 351
1174 View SURFACE_WATER_DU2 351
1175 View SURFACE_WATER_HR1 351
1176 View SURFACE_WATER_HR2 352

1177 View SURFACE_WATER_HR3 352
1178 View SURFACE_WATER_HR4 352
1179 View SURFACE_WATER_HR5 352
1180 View SURFACE_WATER_HR6 352
1181 View SURFACE_WATER_HRBS 353
1182 View SURFACE_WATER_HRDX 353
1183 View SURFACE_WATER_HRGS 353
1184 View SURFACE_WATER_MC1 353
1185 View SURFACE_WATER_MC2 353
1186 View SURFACE_WATER_RPRT_VU 354
1187 View SURFACE_WATER_VU1 354
1188 View SURFACE_WATER_VU2 354
1189 View SoilNPK_HR1 355
1190 View SoilNPK_HR2 355
1191 View TEMPERATURE_LOGGER_VU1 356
1192 View TEMPERATURE_LOGGER_VU2 357
1193 View TMI_ActiveSchool250DixonCnt 357
1194 View TMI_CurrentGoodImage 358
1195 View TMI_ImageEligibleSchools 358
1196 View TMI_PotentiallyEligibleSchools 358
1197 View TMI_SchoolLocationViaGPS 359
1198 View TREEBIOM_HR1 359
1199 View TREEBIOM_HR1_LocP 359
1200 View TREEBIOM_HR2 360
1201 View TREEBIOM_HR3 361
1202 View TREEBIOM_WS_DR1 361
1203 View TREEBIOM_WS_DR2 361
1204 View TREEBIOM_WS_DU1 362
1205 View TREEBIOM_WS_DU2 362
1206 View TREEBIOM_WS_MC1 362
1207 View TREEBIOM_WS_MC2 362
1208 View TREEBIOM_WS_RPRT_VU 362
1209 View TREEBIOM_WS_VU1 362
1210 View TREEBIOM_WS_VU2 362
1211 View TREEGRASSBIOM_HR4 363
1212 View TREE_BIOMETRY_DR1 363
1213 View TREE_BIOMETRY_DR2 363
1214 View TREE_BIOMETRY_DU1 363
1215 View TREE_BIOMETRY_DU2 363
1216 View TREE_BIOMETRY_MC1 363
1217 View TREE_BIOMETRY_MC2 363
1218 View TREE_BIOMETRY_RPRT_VU 363
1219 View TREE_BIOMETRY_VU1 364
1220 View TREE_BIOMETRY_VU2 364
1221 View TURBIDITY_WS_DR1 364
1222 View TURBIDITY_WS_DR2 364
1223 View TURBIDITY_WS_DU1 364
1224 View TURBIDITY_WS_DU2 364
1225 View TURBIDITY_WS_MC1 364
1226 View TURBIDITY_WS_MC2 364
1227 View TURBIDITY_WS_RPRT_VU 365
1228 View TURBIDITY_WS_VU1 365
1229 View TURBIDITY_WS_VU2 365
1230 View Teacher_FirstActive 365
1231 View Teacher_FirstSchool 365
1232 View Teacher_PreService 366
1233 View Trees_HR0 366
1234 View Turbidity_HR1 366
1235 View Turbidity_HR2 367
1236 View View_642 367
1237 View WATER_TEMP_HR1 367
1238 View WATER_TEMP_HR2 367

1239 View WORKSHOP_VIEW 368
1240 View WWW_USERS_VIEW 368
1241 View ZIPLOC_MISMATCH 368
1242 View ZIPMISSING_MISMATCH 369
1243 View ZIPSTATECITY_MISMATCH 369
1244 View ZIP_TO_COUNTY 369
1245 Procedure CalcAOT 370
1246 Procedure CalcAZTransPct 372
1247 Procedure DiffsolarNoonSQL 372
1248 Procedure EqtimeSQL 372
1249 Procedure ListMonthsCert 373
1250 Procedure ListMonthsNew 374
1251 Procedure ListMonthsPrev 374
1252 Procedure ListMonthsYear 375
1253 Procedure ListTeachers 376
1254 Procedure LocalnoonSQL 377
1255 Procedure TruesolarnoonSQL 377
1256 Procedure UTtimeSQL 378

1 Domain Datatype

Name	Code	Data Type
AOTdomain	AOT_DOM	NUMBER(4,3)
AOTmethodDomain	AOTMETHOD_DOM	NUMBER(1)
AcidityDomain	ACIDITY_DOM	NUMBER(3,1)
AcidityMethodDomain	ACIDITYMETHOD_DOM	NUMBER(1)
ActionTextDomain	ACTIONTEXT_DOM	VARCHAR(150)
ActivityRoleDomain	ACTIVITYROLE_DOM	NUMBER(1)
AddressLineDomain	ADDRESSLINE_DOM	VARCHAR2(50)
AirTempDomain	AIRTEMP_DOM	NUMBER(3,1)
AlkalConstantDomain	ALKALCONSTANT_DOM	NUMBER(3,1)
AlkalDropsDomain	ALKALDROPS_DOM	NUMBER(2)
AlkalRangeDomain	ALKALRANGE_DOM	NUMBER(1)
AlkalinityDomain	ALKALINITY_DOM	NUMBER(3)
AndersonDomain	ANDERSON_DOM	CHAR(2)
AnnualGroupDomain	ANNUALGROUP_DOM	VARCHAR2(9)
AppNumDomain	APPNUM_DOM	CHAR(8)
AugerDiamDomain	AUGERDIAM_DOM	NUMBER(3,1)
AvailabilityDomain	AVAILABILITY_DOM	VARCHAR2(5)
BankMaterialDomain	BANKMATERIAL_DOM	NUMBER(1)
BarometricPressureDomain	BAROMETRICPRESSURE_DOM	NUMBER(4)
BedrockDomain	BEDROCK_DOM	NUMBER(1)
BioSquaresDomain	BIOSQUARES_DOM	NUMBER(3)
BiomassDomain	BIOMASS_DOM	NUMBER(5,1)
BulkDensityDomain	BULKDENSITY_DOM	NUMBER(3,2)
CCcodeDomain	CCCODE_DOM	CHAR(2)
CableNbrDomain	CABLENBR_DOM	VARCHAR(10)
CanopyCoverDomain	CANOPYCOVER_DOM	NUMBER(1)
CarbonateDomain	CARBONATE_DOM	NUMBER(1)
ChannelIdDomain	CHANNELID_DOM	CHAR(1)
CityDomain	CITY_DOM	VARCHAR2(30)
ClientTypeDomain	CLIENTTYPE_DOM	CHAR(2)
CloudCoverDomain	CLOUDCOVER_DOM	NUMBER(2)
CloudPresenceDomain	CLOUDPRESENCE_DOM	NUMBER(1)
Comment500Domain	COMMENT500_DOM	VARCHAR2(500)
CommentDomain	COMMENT_DOM	VARCHAR2(200)
CompasAngleDomain	COMPASANGLE_DOM	NUMBER(3)
ConductivityDomain	CONDUCTIVITY_DOM	NUMBER(4)
CongressBranchDomain	CONGRESS_BR_DOM	CHAR(1)
CongressDistrictDomain	CONGRESSDISTRICT_DOM	CHAR(2)
CongressNbrDomain	CONGRESS_NBR_DOM	NUMBER(2)
ContactCodeDomain		

Name	Code	Data Type
ContactMethodDomain	CONTACTMETHOD_DOM	CHAR(2)
ContainerIdDomain	CONTAINERID_DOM	CHAR(2)
CountryDomain	COUNTRY_DOM	CHAR(2)
CoverClassDomain	COVERCLASS_DOM	Number(2)
CoverTypeDomain	COVERTYPE_DOM	CHAR(2)
CreditTypeDomain	CREDITTYPE_DOM	VARCHAR2(6)
CreditsCodeDomain	CREDITSCODE_DOM	CHAR(1)
DBmasterSiteDomain	DBMASTERSITE_DOM	CHAR(1)
DatabaseActionDomain	DATABASEACTION_DOM	CHAR(1)
DateDomain	DATE_DOM	DATE
DateHourDomain	DATEHOUR_DOM	DATE
DatetimeDomain	DATETIME_DOM	DATE
DensityMethodDomain	DENSITYMETHOD_DOM	NUMBER(1)
Description	DESCRIPTION	Varchar2(800)
DewPointDomain	DEWPOINT_DOM	NUMBER(3,1)
DirectSourceDomain	DIRECTSOURCE_DOM	NUMBER(1)
DirectionAngleDomain	DIRECTIONANGLE_DOM	NUMBER(4,1)
DissolvedOxyDomain	DISSOLVEDOXY_DOM	NUMBER(3,1)
Dist500mLDomain	DIST500MLDOMAIN	Number(3,1)
DocumentDomain	DOCUMENT_DOM	Varchar2(2000)
DuffLitterDomain	DUFFLITTER_DOM	NUMBER(2)
DyingStateDomain	DYINGSTATE_DOM	CHAR(1)
ETconstantDomain	ETCONSTANT_DOM	NUMBER4,3)
ElevationDomain	ELEVATION_DOM	NUMBER(5,1)
EmailDomain	EMAIL_DOM	VARCHAR2(50)
EvalAgreeDomain	EVALAGREE_DOM	NUMBER(1)
EvalCategoryDomain	EVALCATEGORY_DOM	VARCHAR2(10)
EvalEffectiveDomain	EVALEFFECTIVE_DOM	NUMBER(1)
EvalNumberDomain	EVALNUMBER_DOM	NUMBER(4)
EvalPreparedDomain	EVALPREPARED_DOM	NUMBER(1)
EvalRatingDomain	EVALRATING_DOM	NUMBER(3,1)
EvalScoreAvgDomain	EVALSCOREAVG_DOM	NUMBER(2,1)
FIPScountyCode	FIPSCOUNTYCODE	CHAR(5)
FertilityEstDomain	FERTILITYEST_DOM	NUMBER(1)
FlagDomain	FLAG_DOM	CHAR(1)
FranContactTypeDomain	FRANCONTACTTYPE_DOM	CHAR(2)
FranchiseDomain	FRANCHISE_DOM	CHAR(8)
FuelCountDomain	FUELCOUNT_DOM	NUMBER(4)
FullName2Domain	FULLNAME2_DOM	VARCHAR2(160)
FullNameDomain	FULLNAME_DOM	VARCHAR2(80)
GDSdomain		

Name	Code	Data Type
GenusDomain	GENUS_DOM	CHAR(4)
GlobePhaseDomain	GLOBEPHASE_DOM	CHAR(4)
GoalTextDomain	GOALTEXT_DOM	VARCHAR2(2000)
GradeLevelDomain	GRADELEVEL_DOM	CHAR(1)
GradeListDomain	GRADELIST_DOM	VARCHAR(4)
GrantNumberDomain	GRANTNUMBER_DOM	VARCHAR2(5)
GrantScoreDomain	GRANTSCORE_DOM	NUMBER(3,1)
GrantStatusDomain	GRANTSTATUS_DOM	VARCHAR2(2)
GrantTrainingDomain	GRANTTRAINING_DOM	CHAR(1)
GravOffsetDomain	GRAVOFFSET_DOM	NUMBER(4,1)
GreeningCycleDomain	GREENINGCYCLE_DOM	NUMBER(1)
GreeningIdDomain	GREENINGID_DOM	NUMBER(2)
GreeningStateDomain	GREENINGSTATE_DOM	CHAR(1)
GroupCodeDomain	GROUPCODE_DOM	CHAR(3)
GroupIdDomain	GROUPID_DOM	NUMBER(2)
HRnbrDomain	HRNBR_DOM	INTEGER
HazeEstDomain	HAZEEST_DOM	NUMBER(1)
HazeSensitivityDomain	HAZESENSITIVITY_DOM	NUMBER(5,4)
HazeTransmissionDomain	HAZETRANSMISSION_DOM	NUMBER(4,1)
HeightAngleDomain	HEIGHTANGLE_DOM	NUMBER(2)
HeightCmDomain	HEIGHTCM_DOM	NUMBER(3)
HorizonDomain	HORIZON_DOM	CHAR(1)
HourDomain	HOUR_DOM	NUMBER(2)
HumidityDomain	HUMIDITY_DOM	NUMBER(4,1)
HumidityMethodDomain	HUMIDITYMETHOD_DOM	NUMBER(1)
HydroWeightDomain	HYDROWEIGHT_DOM	NUMBER(3,1)
IPpayerDomain	IPPAYER_DOM	CHAR(1)
ImgDataSourceDomain	IMGDATASOURCE_DOM	CHAR(5)
ImgDatumDomain	IMGDATUM_DOM	CHAR(5)
ImgLinesDomain	IMGLINES_DOM	INTEGER
ImgLocatorDomain	IMGLOCATOR_DOM	VARCHAR2(100)
ImgProjDistanceDomain	IMGPROJDISTANCE_DOM	INTEGER
ImgProjUnitsDomain	IMGPROJUNITS_DOM	VARCHAR2(6)
ImgProjValueDomain	IMGPROJVALUE_DOM	FLOAT
ImgProjectionDomain	IMGPROJECTION_DOM	CHAR(3)
ImgSamplesDomain	IMGSAMPLES_DOM	INTEGER
ImgSensorDomain	IMGSENSOR_DOM	CHAR(4)
ImgStatusDomain	IMGSTATUS_DOM	VARCHAR2(6)
InfiltrateDepthDomain	INFILTRATEDEPTH_DOM	NUMBER(3)
InfiltrationDomain	INFILTRATION_DOM	NUMBER(3)
InfiltrationMinDomain		

Name	Code	Data Type
InfiltrationSecDomain	INFILTRSEC_DOM	NUMBER(2)
InvestigationDomain	INVESTIGATION_DOM	CHAR(4)
InvoiceNbrDomain	INVOICENBR_DOM	VARCHAR(30)
IssuesTextDomain	ISSUESTEXT_DOM	VARCHAR2(3000)
KitMfgNameDomain	KITMFGNAME_DOM	VARCHAR2(10)
LanguageCodeDomain	LANGUAGECODE_DOM	CHAR(2)
LanguageDomain	LANGUAGE_DOM	VARCHAR2(12)
LatitudeDomain	LATITUDE_DOM	NUMBER(6,4)
LeafColorDomain	LEAFCOLOR_DOM	VARCHAR2(10)
LeafIdDomain	LEAFID_DOM	NUMBER(1)
LeafLengthDomain	LEAFLENGTH_DOM	NUMBER(3)
LevelTypeDomain	LEVELTYPE_DOM	CHAR(3)
LifeCycleDomain	LIFECYCLE_DOM	CHAR(1)
LilacHeightDomain	LILACHEIGHT_DOM	NUMBER(3)
LilacIdDomain	LILACID_DOM	NUMBER(2)
LilacTypeCodeDomain	LILACTYPECODE_DOM	NUMBER(1)
LocationMethodDomain	LOCATIONMETHOD_DOM	NUMBER(1)
LongitudeDomain	LONGITUDE_DOM	NUMBER(7,4)
MUCcodeDomain	MUC_DOM	VARCHAR(4)
MUCdescriptionDomain	MUCDESCRIPTION_DOM	VARCHAR2(800)
MUCnameDomain	MUCNAME_DOM	VARCHAR2(100)
MUCsourceDomain	MUCSOURCE_DOM	CHAR(1)
MailFormatDomain	MAILFORMAT_DOM	CHAR(3)
MailStatusDomain	MAILSTATUS_DOM	VARCHAR2(5)
MailTopicDomain	MAILTOPIC_DOM	VARCHAR2(10)
MailingNameDomain	MAILINGNAME_DOM	VARCHAR2(15)
MeasurementDomain	MEASUREMENT_DOM	CHAR(2)
MinuteDomain	MINUTE_DOM	NUMBER(2)
ModelNameDomain	MODELNAME_DOM	VARCHAR2(16)
MoistureEstDomain	MOISTUREEST_DOM	NUMBER(1)
MoistureProtocolDomain	MOISTPROTOCOL_DOM	NUMBER(1)
MoneyDomain	MONEY_DOM	NUMBER(10,2)
MotivationTextDomain	MOTIVATIONTEXT_DOM	VARCHAR2(4000)
MovingWaterDomain	MOVINGWATER_DOM	NUMBER(1)
NameDomain	NAME_DOM	VARCHAR2(32)
NameTypeDomain	NAMETYPE_DOM	CHAR(1)
NbrItemsDomain	NBRITEMS_DOM	NUMBER(2)
NbrSchoolsDomain	NBRSCHOOLS_DOM	NUMBER(5)
NitrateDomain	NITRATE_DOM	NUMBER(3,1)
ObscuredDomain	OBSCURED_DOM	NUMBER(2)
ObstacleDistDomain		

Name	Code	Data Type
OrderNumberDomain	ORDERNBR_DOM	CHAR(8)
OrderTypeDomain	ORDERTYPE_DOM	CHAR(5)
OzoneMethodDomain	OZONEMETHOD_DOM	NUMBER(1)
OzonePPBDomain	OZONEPPB_DOM	NUMBER(3)
PETdomain	PET_DOM	NUMBER(7,2)
PaperDocLocDomain	PAPERDOCLOC_DOM	VARCHAR(30)
PartnerDomain	PARTNER_DOM	CHAR(8)
PercentDomain	PERCENT_DOM	NUMBER(4,1)
PersonIdDomain	PERSON_DOM	CHAR(8)
PhotoSubmittedDomain	PHOTOSUBMITTED_DOM	NUMBER(1)
PhotomVoltageDomain	PHOTOMVOLTAGE_DOM	NUMBER(5,4)
PhotometerDomain	PHOTOMETER_DOM	VARCHAR2(8)
PlantedFlagDomain	PLANTEDFLAG_DOM	NUMBER(1)
PostActionItemDomain	POSTACTIONITEM_DOM	VARCHAR(15)
PostalDomain	POSTAL_DOM	VARCHAR2(10)
PreActionItemDomain	PREACTIONITEM_DOM	VARCHAR(15)
PrecipDaysDomain	PRECIPDAYS_DOM	NUMBER(2)
PrefixDomain	PREFIX_DOM	VARCHAR2(10)
PressureMethodDomain	PRESSUREMETHOD_DOM	NUMBER(1)
PrevalenceDomain	PREVALENCE_DOM	CHAR(2)
ProductCodeDomain	PRODUCTCODE_DOM	VARCHAR2(12)
Protocol ListDomain	PROTOCOL_LIST_DOM	VARCHAR2(30)
QualityOfLangDomain	QUALITYOFLANGDOMAIN	CHAR(1)
QuantityDomain	QUANTITY_DOM	NUMBER(6)
QueryStatusDomain	QUERYSTATUS_DOM	CHAR(1)
QuestAnswerDomain	QUESTANSWER_DOM	VARCHAR2(10)
QuestOrderDomain	QUESTORDER_DOM	NUMBER(3)
QuestTargetDomain	QUESTTARGET_DOM	CHAR(2)
QuestVersionDomain	QUESTVERSION_DOM	NUMBER(1)
QuestionTextDomain	QUESTIONTEXT_DOM	Varchar2(255)
RainfallDomain	RAINFALL_DOM	NUMBER(5,1)
RecruitCodeDomain	RECRUITCODE_DOM	CHAR(1)
RecruitTextDomain	RECRUITTEXT_DOM	VARCHAR(20)
RegTypeCodeDomain	REGTYPECODE_DOM	CHAR(2)
RegionCategoryDomain	REGIONCATEGORY_DOM	VARCHAR2(15)
RegionTypeDomain	REGIONTYPE_DOM	VARCHAR2(20)
RegisterCodeDomain	REGISTERCODE_DOM	CHAR(2)
ResponsibilityDomain	RESPONSIBILITY_DOM	VARCHAR(15)
RingDiamDomain	RINGDIAM_DOM	NUMBER(3,1)
RockEstDomain	ROCKEST_DOM	NUMBER(1)
RockVolumeDomain		

Name	Code	Data Type
RoleDomain	ROLE_DOM	Varchar2(15)
RootEstDomain	ROOTEST_DOM	NUMBER(1)
RotClassDomain	ROTCLASS_DOM	NUMBER(1)
SalinityDomain	SALINITY_DOM	NUMBER(3,1)
SalinityMethodDomain	SALINITYMETHOD_DOM	NUMBER(1)
SaltOrFreshDomain	SALTORFRESH_DOM	NUMBER(1)
SalutationDomain	SALUTATION_DOM	VARCHAR2(35)
SchoolFundingDomain	SCHOOLFUNDING_DOM	NUMBER(1)
SchoolIdDomain	SCHOOL_DOM	CHAR(8)
SchoolNameDomain	SCHOOLNAME_DOM	VARCHAR2(100)
SchoolTypeDomain	SCHOOLTYPE_DOM	NUMBER(2)
SeasonDomain	SEASON_DOM	CHAR(4)
ServiceTypeDomain	SERVICETYPE_DOM	CHAR(2)
SessionDomain	SESSION_DOM	CHAR(5)
SiteCodeDomain	SITECODE_DOM	CHAR(3)
SiteDistanceDomain	SITEDISTANCE_DOM	NUMBER(6,1)
SiteProximityDomain	SITEPROXIMITY_DOM	NUMBER(1)
SkyColorEstDomain	SKYCOLOREST_DOM	NUMBER(1)
SlopeDomain	SLOPE_DOM	NUMBER(2)
SnowDailyDomain	SNOWDAILY_DOM	NUMBER(4)
SnowTotalDomain	SNOWTOTAL_DOM	NUMBER(4)
SoilColorDomain	SOILCOLOR_DOM	VARCHAR2(11)
SoilConsistenceEstDomain	SOILCONSISTEST_DOM	NUMBER(1)
SoilDepositDomain	SOILDEPOSIT_DOM	NUMBER(2)
SoilDepthDomain	SOILDEPTH_DOM	NUMBER(2)
SoilDryMthdDomain	SOILDRYMTHD_DOM	NUMBER(1)
SoilDryTimeDomain	SOILDRYTIME_DOM	NUMBER(5,2)
SoilHorDepthDomain	SOILHORDEPTH_DOM	NUMBER(4,1)
SoilMeterDomain	SOILMETER_DOM	NUMBER(3,1)
SoilSourceDomain	SOILSOURCE_DOM	NUMBER(1)
SoilStructEstDomain	SOILSTRUCTEST_DOM	NUMBER(1)
SoilTempDomain	SOILTEMP_DOM	NUMBER(3,1)
SoilTempUsageDomain	SOILTEMPUSAGE_DOM	NUMBER(1)
SoilTextureDomain	SOILTEXTURE_DOM	NUMBER(2)
SoilThermomDomain	SOILTHERMOM_DOM	NUMBER(1)
SoilVolumeDomain	SOILVOLUME_DOM	NUMBER(5,1)
SoilWeightDomain	SOILWEIGHT_DOM	NUMBER(4,1)
SolarDatetimeDomain	SOLARDATETIME_DOM	DATE
SolarDayDomain	SOLARDAY_DOM	DATE
SolarHourDomain	SOLARHOUR_DOM	NUMBER(2)
SolarMinDomain		

Name	Code	Data Type
SolarNoonDiffDomain	SOLARNOONDIFF_DOM	NUMBER(5)
SourceStateDomain	SOURCESTATE_DOM	NUMBER(1)
SpeciesDomain	SPECIES_DOM	CHAR(4)
SpecificGravityDomain	SPECGRAV_DOM	NUMBER(5,4)
SplitSessionDomain	SPLITSESSION_DOM	VARCHAR2(50)
StandingWaterDomain	STANDINGWATER_DOM	NUMBER(1)
State(US)Domain	STATE_US_DOM	CHAR(2)
StateProvDomain	STATEPROV_DOM	VARCHAR2(5)
StockNbrDomain	STOCKNBR_DOM	VARCHAR2(30)
StrategyText	STRATEGYTEXT	VARCHAR2(4000)
StudySiteDomain	STUDYSITE_DOM	CHAR(6)
SuffixDomain	SUFFIX_DOM	VARCHAR2(10)
SurfaceCoverDomain	SURFACECOVER_DOM	NUMBER(1)
SurfaceTypeDomain	SURFACETYPE_DOM	NUMBER(1)
TableNameDomain	TABLENAME_DOM	VARCHAR2(18)
TableTypeDomain	TABLETYPE_DOM	CHAR(2)
TaskIdDomain	TASKID_DOM	Varchar2(10)
TeacherRoleDomain	TEACHERROLE_DOM	VARCHAR2(20)
TeacherStatusDomain	TEACHERSTATUS_DOM	VARCHAR2(5)
TelPrefixUSDomain	TELPREFIXUS_DOM	CHAR(3)
TelephoneDomain	TELEPHONE_DOM	VARCHAR2(25)
ThermomTypeDomain	THERMOMTYPE_DOM	NUMBER(1)
TideTypeDomain	TIDETYPE_DOM	NUMBER(1)
TimeZoneUSDomain	TIMEZONEUS_DOM	VARCHAR2(5)
TimezoneDomain	TIMEZONE_DOM	VARCHAR2(10)
TraceMissDomain	TRACEMISS_DOM	CHAR(1)
TrackingIdDomain	TRACKINGIDDOMAIN	CHAR(7)
TrainerLevelDomain	TRAINERLEVEL_DOM	VARCHAR2(9)
TrainerRoleDomain	TRAINERROLE_DOM	CHAR(7)
TrainerTypeDomain	TRAINERTYPE_DOM	CHAR(4)
TrainingOptionDomain	TRAININGOPTION_DOM	CHAR(2)
TransectIdDomain	TRANSECTID_DOM	NUMBER(2)
TreeCircumDomain	TREECIRCUM_DOM	NUMBER(5,1)
TreeDiamDomain	TREEDIAM_DOM	NUMBER(5,1)
TreeGrassShrubDomain	TREEGRASSSHRUBDOMAIN	CHAR(1)
TreeHeightDomain	TREEHEIGHT_DOM	NUMBER(3,1)
TreeIdDomain	TREEID_DOM	NUMBER(2)
TurbidityEstDomain	TURBIDITYEST_DOM	NUMBER(1)
TurbiditySecchiDomain	TURBSECCHI_DOM	NUMBER(4,2)
TurbidityTubeDomain	TURBTUBE_DOM	NUMBER(3)
URLdomain		

Name	Code	Data Type
UToffsetDomain	UTOFFSET_DOM	CHAR(6)
UrbanCodeDomain	URBANCODE_DOM	CHAR(2)
VendorDomain	VENDOR_DOM	VARCHAR2(12)
VisualDayDomain	VISUALDAY_DOM	DATE
WaterAreaDomain	WATERAREA_DOM	NUMBER(8,3)
WaterContentDomain	WATERCONTENT_DOM	NUMBER(5,2)
WaterDepthDomain	WATERDEPTH_DOM	NUMBER(4,1)
WaterDiffDomain	WATERDIFF_DOM	NUMBER(7,2)
WaterLocationDomain	WATERLOC_DOM	NUMBER(1)
WaterSizeDomain	WATERSIZE_DOM	NUMBER(1)
WaterTempDomain	WATERTEMP_DOM	NUMBER(3,1)
WaterWidthDomain	WATERWIDTH_DOM	NUMBER(5,1)
WavelengthDomain	WAVELENGTH_DOM	NUMBER(4)
WindSpeedDomain	WINDSPEED_DOM	NUMBER(2)
WkshpCategoryDomain	WKSHPCATEGORY_DOM	VARCHAR2(7)
WorkshopDomain	WORKSHOP_DOM	VARCHAR2(8)
YearDomain	YEAR_DOM	CHAR(4)
ZipCodeUS5Domain	ZIPCODEUS5_DOM	CHAR(5)
ZoomLevelDomain	ZOOMLEVEL_DOM	VARCHAR2(50)

2 AOTdomain

2.1 Description of AOTdomain

The AOT Domain consists of values used to represent Aerosol Optical Transmission (AOT), which is derived from geographic location, air temperature, barometric pressure, calibration constants, etc. As of Fall 2000, there is only one method, presented in the Annotation section for this domain.. It is possible that other methods will be defined in the future. AOT usually varies between 0 and 5, but due to roundoff or calibration errors, it is possible that AOT may be slightly negative.

If AOT < 0.1 then HazeEst <= 2; If AOT >0.5 then HazeEst = 5.

2.2 Annotation for AOTdomain

Expression for AOT from D. Brooks home page 7/11/00

```
function get  AOT(form) {
/* Get Julian date */
  m=parseInt(form.month.value);
  d=parseInt(form.day.value);
  y=parseInt(form.year.value);
  hour=parseFloat(form.hr.value);
  minute=parseFloat(form.min.value);
  second=parseFloat(form.sec.value);
  Lat=parseFloat(form.lat.value);
  Lon=parseFloat(form.lon.value);
  temp=Math.ceil((m-14)/12); /* This number is always <= 0. */
  form.JD.value= d - 32075 + Math.floor(1461*(y+4800+temp)/4)
    +Math.floor(367*(m-2-temp*12)/12) /* m-2-temp*12 is always >0. */
    -Math.floor(3*(Math.floor((y+4900+temp)/100))/4);
  form.JD.value=form.JD.value-0.5+hour/24.+minute/1440.+second/86400.;
/* Solar position, ecliptic coordinates */
  dr=Math.PI/180.;
  T=(form.JD.value-2451545.0)/36525.0;
  L0=280.46645+36000.76983*T+0.0003032*T*T;
  M=357.52910+35999.05030*T-0.0001559*T*T-0.00000048*T*T*T;
  M_rad=M*dr;
  e=0.016708617-0.000042037*T-0.0000001236*T*T;
  C=(1.914600-0.004817*T-0.000014*T*T)*Math.sin(M_rad)
    +(0.019993-0.000101*T)*Math.sin(2.*M_rad)+0.000290*Math.sin(3.*M_rad);
/* Replacement code for L_true=fmod(L0+c,360.) */
  L_save=(L0+C)/360.;
  if (L_save < 0.) L_true=(L0+C)-Math.ceil(L_save)*360.;
  else L_true=(L0+C)-Math.floor(L_save)*360.;

  if (L_true < 0.) L_true+=360.;
  f=M_rad+C*dr;
  form.R.value=1.000001018*(1.-e*e)/(1.+e*Math.cos(f));
/* Sidereal time */
  Sidereal_time=280.46061837+360.98564736629*(form.JD.value-2451545.)+0.000387933*T*T
        -T*T*T/38710000.;
/* Replacement code for Sidereal=fmod(Sidereal,360.) */
  S_save=Sidereal_time/360.;
  if (S_save < 0.) Sidereal_time=Sidereal_time-Math.ceil(S_save)*360.;
  else Sidereal_time=Sidereal_time-Math.floor(S_save)*360.;

  if (Sidereal_time < 0.) Sidereal_time+=360.;
/* Obliquity */
  Obliquity=23.+26./60.+21.448/3600.-46.8150/3600.*T-0.00059/3600.*T*T
       +0.001813/3600.*T*T*T;
/* Ecliptic to equatorial */
  Right_Ascension=Math.atan2(Math.sin(L_true*dr)*Math.cos(Obliquity*dr),
                      Math.cos(L_true*dr));
  Declination=Math.asin(Math.sin(Obliquity*dr)*Math.sin(L_true*dr));
  Hour_Angle=Sidereal_time+Lon-Right_Ascension/dr;
  form.Elevation.value=Math.asin(Math.sin(Lat*dr)*Math.sin(Declination)+
          Math.cos(Lat*dr)*Math.cos(Declination)*
          Math.cos(Hour_Angle*dr));
  form.Elevation.value/=dr;
/* Relative air mass */
  cosz=Math.cos((90.-form.Elevation.value)*dr);
```

```
  form.air_m.value=(1.002432*cosz*cosz+0.148386*cosz+0.0096467)/
    (cosz*cosz*cosz+0.149864*cosz*cosz+0.0102963*cosz+0.000303978);
/* AOT calculation */
  V=parseFloat(form.V.value);
  V_dark=parseFloat(form.V_dark.value);
  V0=parseFloat(form.V0.value);
  p=parseFloat(form.p.value);
  form.AOT.value=(Math.log(V0/form.R.value/form.R.value)-Math.log(V-V_dark)-
              0.138*p/1013.25*form.air_m.value)/form.air_m.value;
  form.transmission.value=100*Math.exp(-form.AOT.value)
}
```

2.3 Standard checks for AOTdomain

Low value	-1
High value	5
Unit	AOT
List of values	

3 AOTmethodDomain

3.1 Description of AOTmethodDomain

The AOTmethod domain is a set of digits used to identify the method used to derive the AOT from geographic, location, air temperature, barometric pressure, calibration constants, etc. As of Fall 2000, there is only one method, presented in the AOT Domain Annotation section.

3.2 Standard checks for AOTmethodDomain

Low value	1
High value	1
Unit	code
List of values	1 GLOBE 2000 Method

4 AcidityDomain

4.1 Description of AcidityDomain

The Acidity domain represents the pH measurement for either water acidity or soil acidity. The measurement for water should vary between 3.0 and 12.0 pH units and the measurement for soil should vary between 1.0 and 14.0 pH units.

The pH of distillled water should be about xx.x.

4.2 Standard checks for AcidityDomain

Low value	1.0
High value	14.0
Unit	pH
List of values	

5 AcidityMethodDomain

5.1 Description of AcidityMethodDomain

The AcidityMethod domain represents the method by which a water or soil acidity measurement is taken. It can be one of: pH Paper, pH Pen, or pH Meter.

Low value	0
High value	9
Unit	pH
List of values	0□Unchecked 1□Litmus Paper 2□pH Pen 3□pH Meter

6 ActionTextDomain

6.1 Standard checks for ActionTextDomain

Low value	
High value	
Unit	
List of values	

7 ActivityRoleDomain

7.1 Description of ActivityRoleDomain

The ActivityRole domain is a set of codes to identify the role that a participant in a GLOBE SchoolToSchool Activity can have. Initially, the roles are: Moderator (0), Participant (1), or Scientist(2).

7.2 Standard checks for ActivityRoleDomain

Low value	0
High value	9
Unit	
List of values	0□Moderator 1□Participant 2□Scientist

8 AddressLineDomain

8.1 Description of AddressLineDomain

The AddressLine domain represents local postal information for mail delivery. The line might include items like street name, street number, building number, post box number, neighborhood name, etc. The domain is limited in length so that it can be applied to mailing labels with no further processing. Any required abbreviations to get complete street address information into the defined length is the responsibility of the data entry person; it should be possible to let schools, teachers, and other entities with an address update their own address information.

8.2 Standard checks for AddressLineDomain

Low value	
High value	
Unit	
List of values	

9 AirTempDomain

9.1 Description of AirTempDomain

The AirTemp domain represents air temperature, either at a specific time or a maximum or minimum value over some time interval. It is measured in degrees celsius (degC) and usually varies between -65.0 and +55.0 degC.

9.2 Standard checks for AirTempDomain

Low value	-65.0
High value	+55.0
Unit	deg C
List of values	

10 AlkalConstantDomain

10.1 Description of AlkalConstantDomain

The AlkalConstant domain identifies the alkalinity conversion constant of a given alkalinity kit. The conversion constant for the Hach kit is 6.8. It may vary for other alkalinity kit manufacturers.

10.2 Standard checks for AlkalConstantDomain

Low value	0.0
High value	99.9
Unit	???
List of values	

11 AlkalDropsDomain

11.1 Description of AlkalDropsDomain

The AlkalDrops domain represents the number of drops of an acid titrant used in an experiment to determine the alkalinity of a sample of water.

11.2 Standard checks for AlkalDropsDomain

Low value	0
High value	99
Unit	drops
List of values	

12 AlkalRangeDomain

12.1 Description of AlkalRangeDomain

The AlkalRange domain represents the alkalinity range of the acid titrant added to a water sample. For GLOBE experimenmts it is "High Range" or "Low Range".

12.2 Standard checks for AlkalRangeDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Low Range 2□High Range

13 AlkalinityDomain

13.1 Description of AlkalinityDomain

The Alkalinity domain represents the alkalinity concentration of a water sample. It is measured in mg/Liter as CaCO3 and is usually between 0 and 500 mg/L. Distilled water is usually about 14 mg/L.

13.2 Standard checks for AlkalinityDomain

Low value	0
High value	500
Unit	mg/L
List of values	

14 AndersonDomain

14.1 Description of AndersonDomain

The Anderson domain is a two digit GLOBE Phase 1 code that identifies different types of land cover. It has been replaced in GLOBE Phase 2 protocols by MUC code. There is not a direct translation between the two codes.

14.2 Standard checks for AndersonDomain

Low value	
High value	
Unit	
List of values	

15 AnnualGroupDomain

15.1 Description of AnnualGroupDomain

The AnnualGroup domain is a set of labels used to identify an annual grouping of goals or strategies declared by a U.S. GLOBE Partner. Typically, the labels will reference a school year, e.g. 2000-2001.

15.2 Standard checks for AnnualGroupDomain

Low value	
High value	
Unit	
List of values	

16 AppNumDomain

16.1 Description of AppNumDomain

The AppNum domain represents the original application number that was assigned to a GLOBE school in the Paradox GLOBE Master Database. This value is of historical and archival importance for schools that were assigned an AppNum because it is still used in some other Paradox database files to reference a GLOBE school. Hard copies of original appliccations are stored in filing cabinets sequentially by this number. The number is a sequence of 8 digits.

16.2 Standard checks for AppNumDomain

Low value	
High value	
Unit	

List of values	

17 AugerDiamDomain

17.1 Description of AugerDiamDomain

The AugerDiam domain represents the auger diameter of the soil auger used in a Bulk Density measurement from a soil hole. The diameter is measured in centimeters (cm) with a possible decimal fraction. The diameter usually varies between 3.0 cm and 10.0 cm.

17.2 Standard checks for AugerDiamDomain

Low value	3.0
High value	10.0
Unit	cm
List of values	

18 AvailabilityDomain

18.1 Description of AvailabilityDomain

The Availability domain identifies a set of codes that represent the availability of a workshop training session for new registrations. The values are:

OPEN Open to all US teachers
FULL Full, open only in the event of a cancellation
RESTR Potential Restricted Attendance
LOCAL Restricted to (within state) Teachers
CLOSE Closed (will not be displayed on Web)
PAST Workshop session already completed.
CANCL Workshop canceled

Note: It is expected that the Availability attribute will be updated automatically to change the value to PAST once the date of the workshop has passed.

Some of the above options were added during Summer 1999 and may not appear in older data records.

18.2 Annotation for AvailabilityDomain

FSL has observed that it may eventually be desirable to add a new column to the WORKSHOP SESSION table to hold a more complete explanation of a given Availability code, e.g. the admittance requirements for a potentially restricted session. If such a column is added it might be named AvailabilityText and point to the Comments domain for its data type.

18.3 Standard checks for AvailabilityDomain

Low value	
High value	
Unit	
List of values	OPEN□Open Enrollment FULL□Enrollment Full RESTR□Restricted Access LOCAL□Restricted to Within-state Teachers CLOSE□Closed - Not Published on Web PAST□Workshop Completed CANCL□Workshop Canceled

19 BankMaterialDomain

19.1 Description of BankMaterialDomain

The BankMaterial domain is a set of codes that represent the various types of bank material surrounding a surface water study site.

Current values are: Soil, Rock, Concrete, Vegetated.

19.2 Standard checks for BankMaterialDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Soil 2□Rock 3□Concrete 4□Vegetated Bank

20 BarometricPressureDomain

20.1 Description of BarometricPressureDomain

The BarometricPressure domain represents atmospheric pressure measured in millibars (Mb), also known as hectopascals. Pressures will range from approximately 700 hectopascals (aka millibars) to about 1070, and may be read on some of the aneroid instruments to the nearest 0.5 mb. The valid extreme values for station pressure will be 500 to 1100 millibars.

20.2 Standard checks for BarometricPressureDomain

Low value	500
High value	1100
Unit	Mb
List of values	

21 BedrockDomain

21.1 Description of BedrockDomain

The Bedrock domain is a set of codes that represent various types of bedrock at a surface water study site. The choices are: Granite, Lime Stone, Volcanics, Mixed Sediments, and Unknown.

21.2 Standard checks for BedrockDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Granite 2□Lime Stone 3□Volcanics 4□Mixed Sediments 5□Unknown

22 BioSquaresDomain

22.1 Description of BioSquaresDomain

The BioSquares domain represents the number of squares of green or brown grasses in a grass biometry experiment. It is used as the numerator or denominator of a percentage calculation.

22.2 Standard checks for BioSquaresDomain

Low value	0

High value	999
Unit	cardinal
List of values	

23 BiomassDomain

23.1 Description of BiomassDomain

The Biomass domain represents the biomass (total mass per unit area in g/m2) of live (green) and senescent (brown) grasses. It usually varies between 0.0 and 5000.0 g/m2.

23.2 Standard checks for BiomassDomain

Low value	0.0
High value	5000.0
Unit	g/m2
List of values	

24 BulkDensityDomain

24.1 Description of BulkDensityDomain

The BulkDensity domain represents the bulk density of a soil sample. It is derived by a formula from other soil measurements and is measured (equivalently) in grams/mL or grams/cm3. It usually varies between 0.5 and 3.0 g/mL.

In november 1999 Elissa noted that some Norwegian peat soils had bulk density less than 0.5, so she suggested that we lower the lower limit for range checks to 0.1 g/mL.

24.2 Standard checks for BulkDensityDomain

Low value	0.1
High value	3.0
Unit	g/mL
List of values	

25 CCcodeDomain

25.1 Standard checks for CCcodeDomain

Low value	
High value	
Unit	
List of values	PC□Point of Contact CC□Country Coordinator SP□Special Help Desk Contact DO□GLOBE Desk Officer

26 CableNbrDomain

26.1 Description of CableNbrDomain

The CableNbr domain is a set of values that represent cable numbers of cables sent from GLOBE. Initially, these values identify cables sent in the process of arranging international workshops, e.g. cables to site coordinators, invited countries, etc.

26.2 Standard checks for CableNbrDomain

Low value	
High value	
Unit	
List of values	

27 CanopyCoverDomain

27.1 Description of CanopyCoverDomain

The CanopyCover domain is a set of codes that represent various canopy cover conditions for a study site. A defined condition is considered to be permanent for a study site and is recorded as such in a site characteristic table.

27.2 Standard checks for CanopyCoverDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Open 2□Some Trees Within 30m 3□Canopy Overhead

28 CarbonateDomain

28.1 Description of CarbonateDomain

The Carbonate domain represents an estimate of the free carbonates present in a soil sample. Values are None, Slight, and Strong.

28.2 Standard checks for CarbonateDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□None 2□Slight 3□Strong

29 ChannelIdDomain

29.1 Description of ChannelIdDomain

The ChannelId Domain is a set of codes used to identify the channel on a multi-channel photometer. On a 2-channel photometer, the channels are labeled Red (R) and Green (G). Each channel is calibrated separately, so each channel will have its own calibration data.

29.2 Standard checks for ChannelIdDomain

Low value	
High value	
Unit	
List of values	R□Red G□Green

30 CityDomain

30.1 Description of CityDomain

The City domain consists of character strings that represent the name of a city. It is restricted in length to ensure that City, State/Provence, and Postal Code information can all fit on a single line of a mailing label.

30.2 Standard checks for CityDomain

Low value	
High value	
Unit	
List of values	

31 ClientTypeDomain

31.1 Description of ClientTypeDomain

The ClientType domain is a set of codes to represent the various ways in which schools may send data to GLOBE. The current choices are: WB = Web Input, CD = CD Input (No longer Used), and EM = Email Input.

31.2 Standard checks for ClientTypeDomain

Low value	
High value	
Unit	code
List of values	WB□Web Input CD□CD Input (Old) EM□Email Input

32 CloudCoverDomain

32.1 Description of CloudCoverDomain

The CloudCover domain is a set of codes that represent the possible cloud cover conditions at a given time. The values currently supported are: Clear, Scattered, Broken, and Overcast. As of, Fall 1998, it is mandatory to enter one of these values.

In the past, schools were allowed to enter other cloud observation data without indicating the cloud cover, but this is no longer possible. A code value of 9 is used to represent these legacy missing values.

In Spring 2000, the CloudCover choices were modified as follows:

None (0%)
Clear (>0% to 10%)
Isolated (>10% to 25%)
Scattered (>25% to 50%)
Broken (>50% to 90%)
Overcast (>90% to 100%)
Obscured (Can't Determine)

New codes were chosen for None, Clear, Isolated, and Obscured so that there is no conflict between the old codes (now deprecated) and the new ones defined above.

See the Annotation paragraph for insights on how to handle visualization of time periods that overlap the two CloudCover classification methods.

32.2 Annotation for CloudCoverDomain

From Dixon Butler 1/13/2000

Let me make sure I understand how this would work in implementation, particularly during the considerable overlap period we will experience (and have to support).

If a school inputs data using the 1997 GLOBE data entry page (existing) the data base will assign 0 for clear, 1 for scattered, 2 for broken, and 3 for overcast or 9 is no data is reported. If a school inputs data using the GLOBE 2000 data entry page (new, paralleling the Data Work Sheet I have distributed) then the data base will assign 4 for no clouds, 5 for clear (new definition), 6 for isolated, 7 for scattered (new), and 8 for obscurred.

For visualizations we could offer combined maps that allow for both data types to be reported. Specifically, there could be three color bar with the new one (in the middle from top to bottom covering the new protocol categories using an even deeper blue for no clouds (or black), the same color to be used for clear in both systems, the lighter and darker colors used for 10% - 25% and 25% to 50% on the contours to be used for 6 and 7 respectively. For building contours (just to belabor the obvious), no clouds would be interpreted as 0%, new clear as 5%, isolated as 17.5%, and new scattered as 37.5%.

Graphing presents a less accomodating situation. For graphs, data reported under the two different protocols cannot be mixed as one graph has 4 possilities while the other has 7. So to graph cloud cover one effectively must treat data from a single site collected under the two different systems as being for two different measurements. A combined graph could be offered (given the descrete nature of cloud cover data) with nine levels (horizontal lines on which data may fall) with the lines starting at the bottom corresponding to 4, 0, 5, 6, 1, 7, 2, 3, 8. Data from the two different protocols could be in different colors or not (not would permit multiple schools to be displayed).

32.3 Standard checks for CloudCoverDomain

Low value	0
High value	16
Unit	code
List of values	0□Clear (0% to <10%) Deprecated since Fall 2000 1□Scattered (10% to 50%) Deprecated since Fall 2000 2□Broken (>50% to 90%) Deprecated since Fall 2000 3□Overcast (>90% to 100%) Deprecated since Fall 2000 9□Not Entered - Legacy Only Deprecated since Fall 2000 10□None (0%) 11□Clear (>0% to 10%) 12□Isolated (>10% to 25%) 13□Scattered (>25% to 50%) 14□Broken (>50% to 90%) 15□Overcast (>90% to 100%) 16□Obscured (Can't Determine)

33 CloudPresenceDomain

33.1 Description of CloudPresenceDomain

The CloudPresence domain is a flag that represents the presence or absence of a specific cloud type. At the present time the values are Cloud Type Present (1) and Cloud Type Not Present (0). It is possible that in the future this domain will evolve to record some measure of the amount of that cloud type that is present, e.g. as a percentage of the sky.

33.2 Standard checks for CloudPresenceDomain

Low value	0
High value	9
Unit	code
List of values	0□Clouds Not Present 1□Clouds Present

34 Comment500Domain

34.1 Standard checks for Comment500Domain

Low value	
High value	
Unit	
List of values	

35 CommentDomain

35.1 Description of CommentDomain

The Comment domain consists of character strings that represent user supplied comments. It is restricted in length for convenience and storage efficiency reasons.

35.2 Standard checks for CommentDomain

Low value	
High value	
Unit	
List of values	

36 CompasAngleDomain

36.1 Description of CompasAngleDomain

The CompassAngle domain is a set of values used to report the direction indicated by a compass. It is used in the SITE_OBSTRUCTION table to report the direction in whole degrees East of North of the left and right edges of an obstacle. It is the set of integers from 0 to 359.

For a more accurate measurement of angles, with recording of a measurement method, see the DirectionAngle domain.

36.2 Standard checks for CompasAngleDomain

Low value	0
High value	359
Unit	degrees
List of values	

37 ConductivityDomain

37.1 Description of ConductivityDomain

The Conductivity domain represents the electrical conductivity of a water sample. It is measured in microSiemens/cm and usually varies between 0 and 2000 microSiemens/cm.

37.2 Standard checks for ConductivityDomain

Low value	0
High value	2000
Unit	mS/cm
List of values	

38 CongressBranchDomain

38.1 Description of CongressBranchDomain

The CongressBranch domain identifies a branch of the U.S. Congress to be either House (H) or Senate (S). It is possible that this domain could be expanded in the future to identify other political subdivisions, e.g. County Commissioners, City Aldermen, etc., so that all of the U.S. politicians potentially associated with a U.S. GLOBE school could be readily identified.

38.2 Standard checks for CongressBranchDomain

Low value	
High value	
Unit	code
List of values	H House S Senate

39 CongressDistrictDomain

39.1 Description of CongressDistrictDomain

The CongressDistrict domain identifies the district of the House of Representatives of the 106th Congress. For those states with only one representative, and for Puerto Rico, US Virgin Islands, Washington DC, Guam, and Palau the Congressional District field is set to "00". If the state code identifies a territory which has no representation in the US Congress (certain Pacific territories) this field is blank.

If the CongressDistrict domain is used to represent a Senate seat for a state, then S1 represents the first senate seat and S2 represents the second senate seat. This enhancement is used only in the Congress People table.

39.2 Standard checks for CongressDistrictDomain

Low value	
High value	
Unit	
List of values	

40 CongressNbrDomain

40.1 Description of CongressNbrDomain

The CongressNbr domain is always used in association with a State(US) domain and a CongressBranch domain; it represents the Senator rank or Congressional district within a U.S. state. If the CongressBranch value identifies Senate, then the CongressNbr value is 1 to indicate the senior senator or 2 to indicate the junior senator. If the CongressBranch value identifies House, then the CongressNbr value identifies the congressional district within the given state.

40.2 Standard checks for CongressNbrDomain

Low value	0
High value	99
Unit	cardinal
List of values	

41 ContactCodeDomain

41.1 Description of ContactCodeDomain

The ContactCode domain consists of a set of codes that represent various workshop session contacts for lodging, computers, public affairs, shipping, etc. The possible codes, a label for each code, a person to contact, and contact information for that person are included in the WorkshopContacts table.

Initially this domain is limited to 12 contact codes: Lodging, Admin, Contract, W-Coord, C-Coord, Computer, R-sens, Press, Ship-S, Ship-L, Hotel, US-Emb.

Low value	
High value	
Unit	code
List of values	LODGING□Lodging ADMIN□Administrative CONTRACT□Contract Signer W-COORD□Workshop Coordinator C-COORD□Country Coordinator COMPUTER□Computers R-SENS□Remote Sensing PRESS□Public Affairs SHIP-S□Ship Small Items Contact SHIP-L□Ship Large Items Contact HOTEL□Hotel Contact US-EMB□U.S. Embassy Contact

42 ContactMethodDomain

42.1 Standard checks for ContactMethodDomain

Low value	
High value	
Unit	
List of values	EM□Email FX□Fax PH□Phone CB□Cable PR□Personal LT□Letter UK□Unknown

43 ContainerIdDomain

43.1 Description of ContainerIdDomain

The ContainerId domain represents the identifiers used by schools to uniquely identify a container used for a soil sample in a soil moisture experiment. The style of identification used is up to the school, so any 2 characters or digits are valid.

43.2 Standard checks for ContainerIdDomain

Low value	
High value	
Unit	
List of values	

44 CountryDomain

44.1 Description of CountryDomain

The Country domain identifies a two character code to represent GLOBE countries. The two characters are those defined by the Internet Society for country identification. The full country name is given in the CountryNameCodes table. Information about the number of schools in a country and other country-wide attributes of schools are maintained in the CountryInfo table.

International Contacts and International Coordinators are kept in the GlobeTeacher table with LifeCycle = 'Admin' and linked to a GLOBE School whose SchoolType = 'District Office'. The TeacherRole attribute in the TeacherToSchool table identifies that teacher as an International Contact or an International Coordinator.

44.2 Standard checks for CountryDomain

Low value	
High value	
Unit	
List of values	

45 CoverClassDomain

45.1 Description of CoverClassDomain

The CoverClass domain identifies the canopy cover percentage class for various canopy measurements in the Fire Fuel Ecology protocol.

45.2 Standard checks for CoverClassDomain

Low value	0
High value	99
Unit	%-class
List of values	1□Less than 1% 2□1% to 5% 10□5% to 15% 20□15% to 25% 30□25% to 35% 40□35% to 45% 50□45% to 55% 60□55% to 65% 70□65% to 75% 80□75% to 85% 90□85% to 95% 99□95% to 100%

46 CoverTypeDomain

46.1 Description of CoverTypeDomain

The CoverType domain identifies the different types of "canopy" types that are potentially recorded at a FIRE FUel Ecology Sstudy site. The four base types are: Live Shrub (LS), Dead Shrub (DS), Live Herbaceous (LH), and Dead Herbaceous (DH). In addition, a Live Dominant (LD) and Live Co-dominant (LC) types may be used when it is not possible to do a complete Land Cover Quantitative analysis of the site.

46.2 Standard checks for CoverTypeDomain

Low value	
High value	
Unit	
List of values	LS□Live Shrub DS□Dead Shrub LH□Live Herbaceous DH□Dead Herbaceous LD□Live Dominant LC□Live Co-dominant

47 CreditTypeDomain

47.1 Description of CreditTypeDomain

The CreditType domain is a set of codes that identify the type of credits being offered, e.g. Continuing Education, Graduate Science, Graduate Mathematics, Undergraduate Science, etc.

47.2 Standard checks for CreditTypeDomain

Low value	
High value	
Unit	
List of values	

48 CreditsCodeDomain

48.1 Description of CreditsCodeDomain

The CreditsCode domain is a set of codes that represent the status of academic credits available to a teacher resulting from the teacher's attendance and participation in a GLOBE workshop session.

48.2 Standard checks for CreditsCodeDomain

Low value	
High value	
Unit	code
List of values	N□None P□Pending R□Received

49 DBmasterSiteDomain

49.1 Description of DBmasterSiteDomain

The DBmasterSite domain is a name that identifies the GLOBE Master Site at which data was first entered in the Distributed Globe Database. When this data is then replicated at other sites, the Master Site identifier is also replicated, thereby identifying the MasterSite at which the original entries, or later updates or deletions, were made.

49.2 Standard checks for DBmasterSiteDomain

Low value	
High value	
Unit	
List of values	

50 DatabaseActionDomain

50.1 Description of DatabaseActionDomain

The DatabaseAction domain identifies a database action to be one of the following: Update, Insert, or Delete.

50.2 Standard checks for DatabaseActionDomain

Low value	
High value	
Unit	code
List of values	U□Update I□Insert D□Delete

51 DateDomain

51.1 Description of DateDomain

The Date domain identifies a calendar day. It is often used in primary keys to identify the date of a GLOBE measurement. Sometimes it is a local date and sometimes it is a UT date; check the column name to determine which it is. The date VALUE satisfies the integrity constraint: TRUNC(VALUE, DD) = VALUE.

51.2 Standard checks for DateDomain

Low value	
High value	
Unit	day
List of values	

52 DateHourDomain

52.1 Description of DateHourDomain

The DateHour domain represents Datetime values that consist of Date and Hour. Sometimes it is a local date/hour and sometimes it is a UT date/hour; check the column name to determine which it is. The date VALUE satisfies the integrity constraint: TRUNC(VALUE, HH) = VALUE.

52.2 Standard checks for DateHourDomain

Low value	
High value	
Unit	day_hour
List of values	

53 DatetimeDomain

53.1 Description of DatetimeDomain

The DateTime domain represents Datetime values that consist of Date, Hour, Minute, and Second. Sometimes it is a local datetime and sometimes it is a UT datetime; check the column name to determine which it is.

53.2 Standard checks for DatetimeDomain

Low value	
High value	
Unit	
List of values	

54 DensityMethodDomain

54.1 Description of DensityMethodDomain

The DensityMethod domain is a set of codes to identify whether the soil sample was collected by the Pit Method or by the Auger Method. It is a derived value entered by the Web Page software depending on which Data Entry Page was used.

54.2 Standard checks for DensityMethodDomain

Low value	
High value	

Unit	code
List of values	1□Pit Method 2□Auger Method

55 Description

55.1 Standard checks for Description

Low value	
High value	
Unit	
List of values	

56 DewPointDomain

56.1 Description of DewPointDomain

The DewPoint domain is equivalent to the AirTemp domain, because dew point is defined as the temperature at which the air can no longer hold moisture.

56.2 Standard checks for DewPointDomain

Low value	????
High value	????
Unit	Celsius
List of values	

57 DirectSourceDomain

57.1 Description of DirectSourceDomain

The DirectSource domain is a set of codes used to represent the various source methods for determining the direction from one location to another. For definition of direction see the DirectionAngle domain.

57.2 Standard checks for DirectSourceDomain

Low value	0
High value	9
Unit	code
List of values	0□Unknown 1□From 8 choices (N,NE,E,SE,S,SW,W,NW) 2□From 16 choices (N,NNE,NE,...,NNW) 3□Estimate (within 10 deg) 4□Measurement (within 1 deg) 5□Precision tools (within .1 deg)

58 DirectionAngleDomain

58.1 Description of DirectionAngleDomain

The DirectionAngle domain represents the compass direction from one location to another. It is measured in degrees between 0 and 360. Standing at the source location and considering one line drawn toward the North and a second line drawn toward the target location, the direction angle is the angle measured in a clockwise direction from the line toward the North to the line toward the target location. By convention North is recorded as 0 deg rather than 360 deg.

A DirectionAngle measurement is usually associated with a DirectSource value, which identifies the method used to determine the

direction. See the DirectSource domain for its defintion.

See also the CompassAngle domain and the HeightAngle domain, each of which allow measurement of an angle as an integer and do not require recording of the source of the measurement.

58.2 Annotation for DirectionAngleDomain

For elementary school classes, the Direction may just be estimated as one of the following: N, NE, E, SE, S, SW, W, NW. For example, if SiteB is "South East" of Site A, and if SiteA is the known site, then the Direction from SiteA to SiteB would be recorded in the database as 135 degrees, together with a flag to indicate the estimation method. For older students more accurate measurements can be made using an accurate map, a compass, or a surveyor's transit.

58.3 Standard checks for DirectionAngleDomain

Low value	0.0
High value	359.9
Unit	degrees
List of values	

59 DissolvedOxyDomain

59.1 Description of DissolvedOxyDomain

The DissolvedOxy domain represents the concentration of dissolved oxygen in water. It is measured in mg/L and usually varies between 0.0 and 15.0 mg/L. The dissolved oxygen concentration in a distilled water sample should be greater then 5.0 mg/L. The units mg/L and ppm are considered to be equivalent.

59.2 Standard checks for DissolvedOxyDomain

Low value	0.0
High value	15.0
Unit	mg/L=ppm
List of values	

60 Dist500mLDomain

60.1 Description of Dist500mLDomain

The Dist500mL domain is used to represent the distance on a 500mL container from the 500mL mark to the base of the container. It is measured in centimeters and usually varies between 25.0 and 30.0 cm.

60.2 Standard checks for Dist500mLDomain

Low value	10.0
High value	40.0
Unit	cm
List of values	

61 DocumentDomain

61.1 Description of DocumentDomain

The document domain represents GLOBE documents that are stored in the database. GLOBE documents include GLOBE Bulletins and workshop reports.

61.2 Standard checks for DocumentDomain

Low value	
High value	
Unit	
List of values	

62 DuffLitterDomain

62.1 Description of DuffLitterDomain

The DuffLitter domain is used to measure the depth (cm) of the "Duff" and "Litter" at a Fire Fuel Ecology study site. Usually, two separate measurements will be recorded in the database, i.e. DuffDepth and TotalDepth, from which other interesting measurements, e.g. LitterDepth or LitterPct, can be derived.

62.2 Standard checks for DuffLitterDomain

Low value	0
High value	99
Unit	cm
List of values	

63 DyingStateDomain

63.1 Description of DyingStateDomain

The DyingState domain identifies the various states that a leaf or blade of grass might go through during its GreenDown cycle. The semantics of each state are as follows.

Unchanged (U) --- This is the state that precedes the GreenDown cycle. Schools may enter mutiple instances of this state, but each should be accompanied by a comment that describes something of interest or gives a prediction of when they anticipate coloration to start. The last day of the Unchanged state is defined to be the start day of the GreenDown cycle. If there are no instances of the Unchanged state, then the first day of all GreenDown instances is the start day for that cycle.

Colorchange (C) --- From Unchanged to Fallen leaf or End of change. This is the state where schools are making daily observations of leaf coloration by comparison with the GLOBE Plant Color Guide. The first day of the Colorchange state is defined to be the FirstChg Day. The last day of the Colorchange state is defined to be the LastChg Day.

Fallen leaf (F) --- Leaf destroyed or lost. Schools may record multiple instances of this event if something interesting happens to the place where the leaf used to be, e.g. a new second leaf begins to grow, the bark changes color, etc. If the leaf falls before reaching the End state, then the leaf is permanently labeled as lost and will not be used in any summary calculations involving averages. The first day of the Fallen state is defined to be the Fall Day.

Snow covered (S) --- This is the state where the coloration of a leaf or blade of grass cannot be observed because of snow. This state may overlap the other states! The first day of the Snow covered state is defined to be the FirstSnow Day. The last day of the Snow covered state is defined to be the LastSnow Day.

End of change (E) --- Leaf has stopped changing color. Schools may record the day on which they have determined that coloration has stopped. This is sometimes difficult to ascertain so a school is not prohibited from entering multiple days when they think this has happened. Schools may also use multiple End of change state instances to describe interesting events that happen before the leaf falls! The first day of the End of change state is defined to be the End Day.

63.2 Standard checks for DyingStateDomain

Low value	
High value	
Unit	identifier

List of values	U□Unchanged C□Color change L□Leaf has fallen S□Snow covered E□End of color change

64 ETconstantDomain

64.1 Description of ETconstantDomain

The ETconstant domain represents the extraterrestrial constant (ET), as determined by a Langley plot calibration or by transfer from a primary reference instrument. A two-channel instrument will have two such constants.

64.2 Standard checks for ETconstantDomain

Low value	
High value	
Unit	
List of values	

65 ElevationDomain

65.1 Description of ElevationDomain

The Elevation domain represents the relative elevation of a study site (measured in meters) either above (+ values) or below (- values) mean sea level. The decimal (or binary) precision of the Elevation attribute may vary, depending upon the accuracy of the measurement device, but will usually have decimal precision of at least 1 decimal place. Elevation usually varies between -300.0 and 5000.0 meters.

65.2 Standard checks for ElevationDomain

Low value	-300.0
High value	5000.0
Unit	m
List of values	

66 EmailDomain

66.1 Description of EmailDomain

The Email domain represents an electronic mail address (usually an Internet email address).

66.2 Standard checks for EmailDomain

Low value	
High value	
Unit	
List of values	

67 EvalAgreeDomain

67.1 Description of EvalAgreeDomain

The EvalAgree domain identifies the score given by a Teacher, or by a Trainer, when asked to Agree or Disagree with a given statement on an evaluation form. The scoring choices are:

0) Not Applicable

1) Strongly Disagree
2) Disagree
3) Agree
4) Strongly Agree

When stored in a relational database, a Null value, if allowed, is used for when a question Is Not Answered.

When calculating averages, only values strictly greater than 0 should be considered.

Whenever the average score is stored in a summary table, the EvalScoreAvg domain should be used to store the result with one decimal point of precision.

67.2 Standard checks for EvalAgreeDomain

Low value	0
High value	9
Unit	code
List of values	0□Not Applicable 1□Strongly Disagree 2□Disagree 3□Agree 4□Strongly Agree

68 EvalCategoryDomain

68.1 Description of EvalCategoryDomain

The EvalCategory domain is a set of codes to represent the various categories of the questions on a teacher or trainer workshop evaluation form. Many of the codes duplicate the codes defined for a GLOBE Invetigation (see InvestigationDomain) whereas others cover other topics related to a workshop session, e.g. Facilities, Learning skills, Preparedness, etc.

68.2 Standard checks for EvalCategoryDomain

Low value	
High value	
Unit	code
List of values	ATM□Atmosphere HYD□Hydrology BIO□Biology & Land Cover GPS□Global Positioning System SOI□Soils LEARN□Learning TECH□Techniques PREP□Preparedness INTRO□Introduction FACIL□Facilities QUEST□Essay Question MISC□Miscellaneous SUPPLSYS□Supplemental Systems

69 EvalEffectiveDomain

69.1 Description of EvalEffectiveDomain

The EvalEffective domain identifies the score given by each Participant in a GLOBE Train-the-Trainer workshop when asked if a particular segment of the Workshop Sesssion was effective in addressing the activities and protocols of that segment. The scoring choices are:

0) Not Trained, Was Not Covered
1) Definitely NOT Effective
2) Somewhat Effective
3) Effective

4) Very Effective

When stored in a relational database, a Null value, if allowed, is used for when a question Is Not Answered.

When calculating averages, only values strictly greater than 0 should be considered.

Whenever the average score is stored in a summary table, the EvalScoreAvg domain should be used to store the result with one decimal point of precision.

69.2 Standard checks for EvalEffectiveDomain

Low value	0
High value	4
Unit	
List of values	0 Not Trained, Was not covered 1 Definitely NOT Effective 2 Somewhat Effective 3 Effective 4 Very Effective

70 EvalNumberDomain

70.1 Description of EvalNumberDomain

The EvalNumber domain is a set of integer identifiers from 1 to 9999. It is used to disambiguate the Trainer and/or Teacher evaluations filled out by trainers and/or teachers who attended a specific workshop session. This column is introduced to accommodate the possibility that evaluations can be filled out anonymously so that TeacherId or TrainerId will be null.

These numbers will be generated automatically by the system when a teacher or trainer (or a GLOBE staff member) completes an evaluation form via a Web page.

Evaluations from multiple workshops may be done in parallel - this domain assumes that no more than 9,999 evaluations will be created from the beginning to the end of any evaluation period! This means RCG has to be careful to avoid duplication when they enter a large number of evaluations from legacy workshops, or enter a new evaluation many months after the bulk of the evaluations were entered.

70.2 Standard checks for EvalNumberDomain

Low value	
High value	
Unit	
List of values	

71 EvalPreparedDomain

71.1 Description of EvalPreparedDomain

The EvalPrepared domain identifies the score given by each Teacher when asked if a particular segment of the Workshop Sesssion adequately prepared them to implement in the classrooom the activities and protocols of that segment. The scoring choices are:

0) Not Trained, Was Not Covered
1) Definitely NOT Prepared to Implement
2) Partially Prepared to Implement
3) Adequately Prepared to Implement
4) Fully Prepared to Implement with Comfort

When stored in a relational database, a Null value, if allowed, is used for when a question Is Not Answered.

When calculating averages, only values strictly greater than 0 should be considered.

Whenever the average score is stored in a summary table, the EvalScoreAvg domain should be used to store the result with one decimal point of precision.

71.2 Standard checks for EvalPreparedDomain

Low value	0
High value	9
Unit	code
List of values	0□Not Trained 1□Not Prepared 2□Partially Prepared 3□Adequately Prepared 4□Fully Prepared

72 EvalRatingDomain

72.1 Description of EvalRatingDomain

The EvalRating domain identifies the overall score given by each Trainer when asked to rate the workshop on a scale of 1-10 (10 being highest).

When stored in a relational database, a Null value, if allowed, is considered to be equivalent to not answering the question at all. When stored in an ascii file, the Null value is represented by 0. Thus, when calculating averages, only values strictly greater than 0 should be considered.

Since the EvalRating domain already includes one decimal digit of precision, any average of these values can be stored in the same datatype, also having one digit of precision.

72.2 Standard checks for EvalRatingDomain

Low value	0
High value	10
Unit	cardinal
List of values	

73 EvalScoreAvgDomain

73.1 Description of EvalScoreAvgDomain

The EvalScoreAvg domain identifies the average score of values taken either from the EvalAgree domain or the EvalPrepared domain. Both of these domains accept integer values from 0 to 4, inclusive, but 0 does not participate in averaging. Thus average values will satify the constraint: 1.0 <= Avg <= 4.0

73.2 Standard checks for EvalScoreAvgDomain

Low value	1.0
High value	4.0
Unit	cardinal
List of values	

74 FIPScountyCode

74.1 Standard checks for FIPScountyCode

Low value	
High value	

Unit	
List of values	

75 FertilityEstDomain

75.1 Description of FertilityEstDomain

The FertilityEst domain represents a gross estimate of the amount of nitrogen, phosporous, or potassium in a soil sample. Possible values are none, low, medium, and high. These values are calculated over multiple samples and then averaged, so numeric values for none (1), low (2), medium(3), and high(4) are used. Because the numeric value is just an estimate to begin with, the averaged values are rounded only to the closest integer value.

If the question is left unanswered, then a null value in the database is defined to be equivalent to a 0 value of this domain. When calculating the average of a set of fertility measurements, the value 0 should never be included in the average.

75.2 Standard checks for FertilityEstDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□None 2□Low 3□Medium 4□High

76 FlagDomain

76.1 Description of FlagDomain

The Flag domain is a set of codes that indicate whether or not a flag is set, or whether the status of the flag is unknown. The allowable values are: Yes (Y), No (N), and Unknown (U). The default value is Unknown.

Some legacy columns are still using a coded representation for this field. Unless otherwise indicated, the coded values are:

0 = Unchecked
1 = Yes
2 = No

76.2 Standard checks for FlagDomain

Low value	
High value	
Unit	boolean
List of values	Y□Yes N□No U□Unknown

77 FranContactTypeDomain

77.1 Description of FranContactTypeDomain

The FranContactType domain is a collection of codes to identify a Franchise contact person. Initially, there are codes to designate only two contacts: Franchise Coordinator (FC) and Training Contact (TR). Other contact types may be added later.

77.2 Standard checks for FranContactTypeDomain

Low value	
High value	
Unit	
List of values	FC□Franchise Coordinator TR□Training Contact

78 FranchiseDomain

78.1 Description of FranchiseDomain

The Franchise domain is a set of identifiers for a GLOBE Franchise. It is used as the primary key of the Franchise table and as a Foreign Key of any table that may reference a franchise.

78.2 Standard checks for FranchiseDomain

Low value	
High value	
Unit	
List of values	

79 FuelCountDomain

79.1 Description of FuelCountDomain

The FuelCount domain is used to count the number of branches that cross the transect in a certain length. The 1 hour and 10 hour branches (0-1 cm and 1-3 cm) are counted only between the 5 and 7 meter marks on the transect; the 100 hour branches (3-8cm) are counted between the 5 and 10 meter marks; and the 1000 hour branches (8+ cm) are counted between the 5 and 25 meter marks.

79.2 Standard checks for FuelCountDomain

Low value	
High value	
Unit	
List of values	

80 FullName2Domain

80.1 Description of FullName2Domain

The FullName2 domain consists of a 2-byte (Unicode) character string used to represent the name, in a local language, of a person, place, or thing.

80.2 Standard checks for FullName2Domain

Low value	
High value	
Unit	
List of values	

81 FullNameDomain

81.1 Description of FullNameDomain

The FullName domain represents the full name of a person, place, or thing. It is limited in length in order to accommodate the

restricted length of a line on a mailing label.

81.2 Standard checks for FullNameDomain

Low value	
High value	
Unit	
List of values	

82 GDSdomain

82.1 Description of GDSdomain

The GDS domain represents the Growing Degree Summation (GDS) measurement of environmental warming. GDS depends upon the daily average temperature (avg of Max and Min temps) for each day of the warming season. The warming season begins Jan 1 for the Northern hemisphere and July 1 for the Southern hemisphere. The GDS for a given day after the start of the warming season is the sum of all positive average uemperatures (deg C) from the beginning of the growing season up to that day. Negative temperatures (deg C) are ignored. Some protocols are only interested in the GDS for the past 30-days.

82.2 Standard checks for GDSdomain

Low value	0
High value	3000.00
Unit	deg C
List of values	

83 GenusDomain

83.1 Description of GenusDomain

The Genus domain is a set of codes that represent the genus of trees at a biology or land cover study site. The Species is recorded in a separate attribute. The 4-character code used at present is just the first 4 characters of the full latin genus name; it uniquely identifies the genus of interest.

83.2 Standard checks for GenusDomain

Low value	
High value	
Unit	
List of values	

84 GlobePhaseDomain

84.1 Description of GlobePhaseDomain

The GlobePhase domain represents a published collection of protocols under which GLOBE measurements are taken. A GLOBE phase identifies a version of the GLOBE protocols published in the GLOBE Program Teacher's Guide. The first phase, called GLOBE I, is identified by 1995 and covers protocols initiated in 1995 and 1996. The second phase, called GLOBE II, is identified by 1996, and covers protocols initiated in 1996.

The phase after GLOBE II is informally called GLOBE 2.5 and is represented by 1997 and covers protocols initiated in 1997. The Teacher's Guide for GLOBE 2.5 is simply identified as GLOBE 1997. Plans for the next version of the GLOBE Teacher's Guide are informally referenced as GLOBE III, but it is not yet certain how those new protocols will be published or how values for GlobePhase will be determined. It is likely that GlobePhase will continue to be identified by a character string of the year in which it was initiated.

GlobePhase is identified in the Measure Codes table to identify the phases in which the MeasureCode is valid.

Low value	
High value	
Unit	Ver No
List of values	

85 GoalTextDomain

85.1 Description of GoalTextDomain

The GoalText domain is the set set of character strings that can be used to describe the major annual goals listed by U.S. Partners in their roles as Recruiters, Trainers, and Mentors of GLOBE Teachers. Currently this text has no internal structure, but conceivably in the future it may satisfy some specific XML Document Type Definition.

85.2 Standard checks for GoalTextDomain

Low value	
High value	
Unit	
List of values	

86 GradeLevelDomain

86.1 Description of GradeLevelDomain

The GradeLevel domain is a set of codes that represent grade levels in a school. The codes are explained in the SchoolGradeCodes table.

86.2 Standard checks for GradeLevelDomain

Low value	
High value	
Unit	code
List of values	P□Primary M□Middle U□Upper H□High School C□College/University

87 GradeListDomain

87.1 Description of GradeListDomain

The GradeList domain is a set of values that identify a list of GradeLevel codes, where each code is represented by a character from the GradeLevel domain. The codes are explained in the SchoolGradeCodes table.

87.2 Standard checks for GradeListDomain

Low value	
High value	
Unit	list

List of values	P□Primary M□Middle U□Upper H□High PM□Primary/Middle MU□Middle/Upper UH□Upper/High PMU□Primary/Middle/Upper MUH□Middle/Upper/High PMUH□Primary/Middle/Upper/High C□College

88 GrantNumberDomain

88.1 Description of GrantNumberDomain

The GrantNumber domain identifies the grant number of a grant received by a U.S. school to help with equipment, communications, and teacher training.

88.2 Standard checks for GrantNumberDomain

Low value	
High value	
Unit	
List of values	

89 GrantScoreDomain

89.1 Description of GrantScoreDomain

The GrantScore domain identifies the score recieved by a school on the grant identified by GrantNumber.

89.2 Standard checks for GrantScoreDomain

Low value	0.0
High value	100.0
Unit	
List of values	

90 GrantStatusDomain

90.1 Description of GrantStatusDomain

The GrantStatus domain is a set of codes that represent the status of a grant given to a school by GLOBE in the early years of the program.

A grant may include funding for one or more of the following items: Computer, Instruments, Modem, or Teacher Training.

GrantStatus has only the following two codes:

01 Completed
02 Funds still available

The Grant Number of the grant is identified by an attribute in the GlobeSchool table.

90.2 Standard checks for GrantStatusDomain

Low value	
High value	
Unit	
List of values	

91 GrantTrainingDomain

91.1 Description of GrantTrainingDomain

The GrantTraining domain is a set of codes that indicate whether or not a teacher has had GLOBE training paid for by a grant.

91.2 Standard checks for GrantTrainingDomain

Low value	
High value	
Unit	code
List of values	G□Grant Pay S□School Pay O□Other

92 GravOffsetDomain

92.1 Description of GravOffsetDomain

The GravOffset domain identifies the offset (in meters) for various measurements taken for the Transect Gravimetric soil moisture protocol. When associated with other gravimetric protocols (e.g. Star, Hole), its value should be null.

92.2 Standard checks for GravOffsetDomain

Low value	0.0
High value	100.0
Unit	m
List of values	

93 GreeningCycleDomain

93.1 Description of GreeningCycleDomain

The GreeningCycle domain is a set of numbers that identifies the grening cycle within a calendar year. Most species have only one greening cycle per year, but some grasses may have up to three cycles per year. Also, because of drought or disease, a tree or shrub may have more than one greening cycle per year.

93.2 Standard checks for GreeningCycleDomain

Low value	1
High value	3
Unit	
List of values	

94 GreeningIdDomain

94.1 Description of GreeningIdDomain

The GreeningId domain is the set of identifiers used to identify trees, shrubs, or square-meter of turf patches at a Phenology site used

for GreenUp or GreenDown protocols. These identifiers are generated by GLOBE to be unique within each SiteId.

94.2 Standard checks for GreeningIdDomain

Low value	
High value	
Unit	
List of values	

95 GreeningStateDomain

95.1 Description of GreeningStateDomain

The GreeningState domain identifies the various states that a leaf or blade of grass might go through during its growing cycle. The semantics of each state are as follows.

Dormant (D) --- No swelling yet. This is the state that precedes the growing season. Schools may enter mutiple instances of this state, but each should be accompanied by a comment that describes something of interest or gives a prediction of when they anticipate swelling to start. The LastDay of the Dormant state is defined to be the start day of the GreenUp cycle. If there are no instances of the Dormant state, then the first day of the greening cycle is the start day for that cycle.

Swelling (S) --- From Dormant to Budburst. This state may only last for a few days. Schools may enter multiple instances of this state with a comment describing swelling progress. The first day of the Swelling state is defined to be the Swell Day. There is no notion of a Swelling state for grasses; instead, grasses change immediately from Dormant to Growing.

Budburst (B) --- State after swelling when first shoots appear. This state usually lasts only one day, but schools are not prohibited from entering multiple days for the Budburst state, each with a comment giving a reason why growth is not yet measurable, e.g. a sudden cold snap. The first day of the Bbudburst state is defined to be the Budburst Day. There is no notion of a Budburst state for grasses; instead, grasses change immediately from Dormant to Growing.

Growing (G) --- Leaf length is being measured. This is the state where schools are making daily measurements of leaf or grass growth. The first day of the Growing state is defined to be the Grow Day. It's often difficult to determine when a leaf has stopped growing, so in addition to a school's declaration that a leaf is Mature, the GLOBE system will calculate the day on which a leaf first reaches 98% of its maximum length. That day is defined to be the Pct98Day.

Mature (M) --- Leaf has stopped growing. Schools may record the day on which they have determined that the leaf is mature. This is sometimes difficult to ascertain so a school is not prohibited from entering multiple days when they think this has happened. Schools can also use multiple Mature state instances to describe interesting events that happed during the summer! The first day in the Mature state is defined to be the Mature Day.

Lost (L) --- Leaf destroyed or lost. Schools may record multiple instances of this event if something intersting happens to the place where the leave used to be, e.g. a new second leaf begins to grow. If the leaf is Lost before reaching the Mature state, then the leaf is permanently labeled as lost and will not be used in any summary calculations involving averages.

95.2 Standard checks for GreeningStateDomain

Low value	
High value	
Unit	
List of values	D□Dormant S□Swelling B□Budburst G□Growing M□Mature L□Lost

96 GroupCodeDomain

96.1 Standard checks for GroupCodeDomain

Low value	
High value	
Unit	
List of values	

97 GroupIdDomain

97.1 Description of GroupIdDomain

The GroupIddomain identifies a GLOBE Mail Pal group within a specific school. The ID is just a number that is arbitrarily assigned. Once a group is created, it exists forever; when it is deactivated, it is no longer listed.

97.2 Standard checks for GroupIdDomain

Low value	
High value	
Unit	
List of values	

98 HRnbrDomain

98.1 Description of HRnbrDomain

The HRnbr Domain identifies the n-th Honor Roll. Honor Rolls began for time periods ending in May 1999 and are calculated every two months.

Honor Roll time periods vary from 4-month periods for Atmosphere, Clouds, Hydrology, etc. to 12-month periods for Land Cover/Biology, etc. The periods may overlap and sometimes one set of measurements qualifies for up to 6 different honor rolls.

The Honor Roll is calculated about the 9-th or 10-th day of June, August, October, December, February, and April, for time periods ending on the last day of the previous month.

98.2 Standard checks for HRnbrDomain

Low value	
High value	
Unit	
List of values	

99 HazeEstDomain

99.1 Description of HazeEstDomain

The HazeEstimate Domain is a set of digits used for a human estimate of the haze condition of the atmosphere. It ranges from 1 to 5 to capture conditions from Clear to Very Hazy.

99.2 Standard checks for HazeEstDomain

Low value	1
High value	5
Unit	code

List of values	1□Unusually Clear 2□Clear 3□Somewhat Clear 4□Hazy 5□Very Hazy

100 HazeSensitivityDomain

100.1 Description of HazeSensitivityDomain

The HazeSensitivity domain is the dark votlage or temperature sensitivity of the photometer. It is measured in V/OC, where V is Volts and OC is ???

100.2 Standard checks for HazeSensitivityDomain

Low value	
High value	
Unit	
List of values	

101 HazeTransmissionDomain

101.1 Description of HazeTransmissionDomain

The HazeTransmission domain consists of derived numeric values used to estimate the percentage of sunlight transmitted through the atmosphere. It is defined to be 100*EXP(-AOT) whre AOT is the Aerosol Optical Transmission. Since it is possible for AOT to be zero, transmission can be 100%. Due to roundoff or calibration errors, it is possible that AOT may be slightly negative. When that happens, transmission is defined to be 100%.

101.2 Standard checks for HazeTransmissionDomain

Low value	0
High value	100
Unit	Celsius
List of values	

102 HeightAngleDomain

102.1 Description of HeightAngleDomain

The HeightAngle domain is a set of values used to report the angle measured between the horizon and the height of an obstruction. It is measured with a clinometer and is used in the SITE OBSTRUCTION table. It is the set of integers from 0 to 89.

For a more accurate measurement of angles, with recording of a measurement method, see the DirectionAngle domain.

102.2 Standard checks for HeightAngleDomain

Low value	0
High value	89
Unit	degrees
List of values	

103 HeightCmDomain

103.1 Description of HeightCmDomain

The HeightCm domain is used primarily in the Atmosphere site definition table to record the the height (in cm) above the ground for the rain gauge, max/min thermometer, and ozone strip. It could be used anywhere for cm measurements that vary between 0 and 999.

103.2 Standard checks for HeightCmDomain

Low value	0
High value	999
Unit	cm
List of values	

104 HorizonDomain

104.1 Description of HorizonDomain

The Horizon domain consists of a set of codes to represent the various Horizons of soil taken from a soil hole. At present, the codes can be one of: O, A, E, B, C, or R.

The Globe II Teachers Guide also discusses subdivisions of the B Horizon and identifies them as B1, B2, or B3. It is possible that subdivisions of other horizons may also be investigated.

For reporting purposes, a subdivision of a horizon is treated as if it were a separate horizon. For this reason a separate HorizonNbr is assigned for each Horizon or SubHorizon identified at a study site, along with the Horizon classification from this domain and recording of its top depth and bottom depth.

104.2 Standard checks for HorizonDomain

Low value	
High value	
Unit	code
List of values	O□O Horizon A□A Horizon E□E Horizon B□B Horizon C□C Horizon R□R Horizon

105 HourDomain

105.1 Description of HourDomain

The hour domain represents an hour (in UT time) of the 24-hour day. It varies between 0 and 23.

105.2 Standard checks for HourDomain

Low value	0
High value	23
Unit	hour
List of values	

106 HumidityDomain

106.1 Description of HumidityDomain

The Humidity Domain consists of values used to represent relative humidity. For the sling psychrometer measurement method, it is derived from WetBulb and DryBulb temperatures and is a ratio that is often viewed as a percentage. For the hygrometer method, it is read digitally from the hygrometer, but an air temperature is also required and is labeled as the Dry Bulb temperature.

106.2 Standard checks for HumidityDomain

Low value	????
High value	????
Unit	%
List of values	

107 HumidityMethodDomain

107.1 Description of HumidityMethodDomain

From: "Travis Andersen" <tanderse@fsl.noaa.gov>
Date: Mon, 24 Apr 2000 14:25:19 -0600
For humidity I was going to have the Sling Psychrometer be a 1 and the Digital Hygrometer be a 2 in the Method column for the Humidity table. If this is not what you want then let me know and I'll code for the numbers you come up with.

107.2 Standard checks for HumidityMethodDomain

Low value	0
High value	9
Unit	digit
List of values	0□Not Available 1□Sling Psychrometer 2□Digital Hygrometer

108 HydroWeightDomain

108.1 Description of HydroWeightDomain

The HydroWeight domain represents the weight of a sand, silt, or clay measurement in the particle-size distribution protocol using the hydrometer method. It is measured in grams and usually varies between 0 and 25 gm.

See the SoilWeight domain for the weights of soil samples taken in soil characteristics, soil moisture, and soil infiltration protocols.

108.2 Standard checks for HydroWeightDomain

Low value	0
High value	25
Unit	gm
List of values	

109 IPpayerDomain

109.1 Description of IPpayerDomain

The IPpayer domain is a set of codes that represent who pays for the IP connection at a GLOBE school. The IP connection for some schools is paid for by GLOBE (G), or by the school itself School (S), in other cases the payer may simply be Unknown (X).

Globe paid IP service for some schools was discontinued in Spring 1998; those schools will have their IPpayer code changed from G to D, for Discontinued, and will have a comment added to the Comments field of the Globe School table.

109.2 Standard checks for IPpayerDomain

Low value	
High value	
Unit	code
List of values	G□GLOBE pays S□School pays D□Discontinued X□Unknown

110 ImgDataSourceDomain

110.1 Description of ImgDataSourceDomain

The ImgDataSourceDomain is a set of codes that represent the possible data sources for creation of a landsat image. Initially, codes exist only for Landsat 4,5, and 7. Other codes will be added as appropriate.

110.2 Standard checks for ImgDataSourceDomain

Low value	
High value	
Unit	
List of values	LS-04□Landsat-4 LS-05□Landsat-5 LS-07□Landsat-7

111 ImgDatumDomain

111.1 Description of ImgDatumDomain

The ImgDatumDomain is a set of codes to represent the datum projection parameter in a landsat image projection. Landsat-5 images have this value set to GRS80; Landsat-7 uses WGS84. The set of codes will be extended as appropriate.

111.2 Standard checks for ImgDatumDomain

Low value	
High value	
Unit	
List of values	GRS80□Default for Landsat-5 WGS84□Default for Landsat-7

112 ImgLinesDomain

112.1 Description of ImgLinesDomain

The ImgLines Domain identifies the number of lines used to create the image. A line is parallel to a longitude and thus the number of lines represents the Width of an image.

112.2 Standard checks for ImgLinesDomain

Low value	
High value	
Unit	
List of values	

113 ImgLocatorDomain

113.1 Description of ImgLocatorDomain

The ImgLocatorDomain identifies the location of the physical file containing the image being described. The image locator may be a file name in a known file system or it may be a Universal Resource locator (URL) accessible by a Web Browser.

113.2 Standard checks for ImgLocatorDomain

Low value	
High value	
Unit	
List of values	

114 ImgProjDistanceDomain

114.1 Description of ImgProjDistanceDomain

The ImgProjDistance Domain identifies the linear projection distance associated with a pixel in a projection image. For square pixels this is the approximate height and width of the pixel. This number is often refered to as the resolution of an image. The measurement units are given by an associated ProjectionUnits attribute in the same row. For Landsat images this distance is usually around 30 meters.

114.2 Standard checks for ImgProjDistanceDomain

Low value	
High value	
Unit	
List of values	

115 ImgProjUnitsDomain

115.1 Description of ImgProjUnitsDomain

The ImgProjUnits Domain identifies the linear measurement units used for image projection. The default is meters, but other units are possible.

115.2 Standard checks for ImgProjUnitsDomain

Low value	
High value	
Unit	
List of values	

116 ImgProjValueDomain

116.1 Description of ImgProjValueDomain

The ImgProjValue Domain represents a linear measurement on the image. The units of the measurement is given by the value of a corresponding ProjectionUnits attribute in the same row. The default is always meters.

116.2 Standard checks for ImgProjValueDomain

Low value	
High value	
Unit	

List of values	

117 ImgProjectionDomain

117.1 Description of ImgProjectionDomain

The ImgProjection Domain is a set of codes that represent the map projection used in generating an image. UTM (universal transverse mercator) is currently used for Landsat-5 images; other possible values are SOM (space oblique mercator), LCC (Lambert conformal conic), etc. The set of codes will be expanded as appropriate.

117.2 Standard checks for ImgProjectionDomain

Low value	
High value	
Unit	
List of values	UTM□Universal Transverse Mercator SOM□Space Oblique Mercator LCC□Lambert Conformal Conic

118 ImgSamplesDomain

118.1 Description of ImgSamplesDomain

The ImgSamples Domain identifies the number of samples used to create the image. A sample varies with latitude and thus the number of samples represents the Height of an image.

118.2 Standard checks for ImgSamplesDomain

Low value	
High value	
Unit	
List of values	

119 ImgSensorDomain

119.1 Description of ImgSensorDomain

The ImgSensorDomain is a set of codes to represent the sensor type that is used for the image data source. For Landsat-5, the sensor type is TM10, for Landsat-7, the sensor type is ETM+. The codes will be extended as appropriate if other image types are used.

119.2 Standard checks for ImgSensorDomain

Low value	
High value	
Unit	
List of values	TM10□??? ETM+□Enhanced Thematic Mapper

120 ImgStatusDomain

120.1 Description of ImgStatusDomain

The ImgStatusDomain is an unstructured domain used to indicate the status of a request from GLOBE to EROS that an image be sent to a specific school. Some possible values are:

Delay -- Meaning that for some reason (explained in the Comments) the image could not be sent immediately, but that the request is

still active and an image will be sent as soon as practical.

Cancel -- Meaning that a request to send an image is no longer valid and GLOBE wishes to Cancel the request previously made to EROS. The reason for the cancelation may be given in the Comments.

Failed -- Meaning that for some reason (explained in the Comments) it is not possible to send an image to the identified school now or in the near future. EROS will take no further action on this specific request.

BadLoc -- Meaning that an image was send to a school, but for some reason the school location on the map was not correct, probably because of an error in the school's location given in the SITE LOCATION table. After making location corrections, GLOBE may initiate a new request that a new image be sent to that school.

Lost -- Meaning that an image was sent to a school, but never received by the school, possibly because of incorrect or incomplete mailing information in the GLOBE SCHOOL table. After making corrections, GLOBE may initiate a new request that a new image be sent to that school.

120.2 Standard checks for ImgStatusDomain

Low value	
High value	
Unit	
List of values	

121 InfiltrateDepthDomain

121.1 Description of InfiltrateDepthDomain

The InfiltrateDepth domain represents the depth of soil through which water flows in a soil infiltration experiment. The depth is measured in millimeters (mm) and the range of values is usually between 20 and 40 mm. Some experiments may fix the depth at 2 cm (20 mm) or 3 cm (30 mm).

This value is used to calculate the flow rate (mm/min), which is an estimate of the rate at which water is absorbed into soil. Explanations of three different kinds of flow rates are given in the SoilInfiltration table.

121.2 Standard checks for InfiltrateDepthDomain

Low value	20
High value	300
Unit	mm
List of values	

122 InfiltrationDomain

122.1 Description of InfiltrationDomain

The Infiltration domain represents the rate of absorption of water into the soil. It is measured in millimeters per minute (mm/min) and usually varies between 0 and 400 mm/min.

122.2 Standard checks for InfiltrationDomain

Low value	0
High value	400
Unit	mm/min
List of values	

123 InfiltrationMinDomain

123.1 Description of InfiltrationMinDomain

The InfiltrationMin domain represents the number of whole minutes elapsed as part of a soil infiltration experiment. The whole number of seconds in the same measurement is stored in a separate attribute.

123.2 Standard checks for InfiltrationMinDomain

Low value	@
High value	
Unit	
List of values	

124 InfiltrationSecDomain

124.1 Description of InfiltrationSecDomain

The InfiltrationSec domain represents the whole number of seconds elapsed as part of a soil infiltration experiment. The whole number of minutes in the same measurement is stored in a separate attribute.

124.2 Standard checks for InfiltrationSecDomain

Low value	
High value	
Unit	
List of values	

125 InvestigationDomain

125.1 Description of InvestigationDomain

The Investigation domain is a set of codes that represent a GLOBE Investigation. Invetigations are defined in the lastest version of the GLOBE Teacher's Guide and Investigation codes are defined in the Investigate Codes table. The relationship among Globe Investigations, Globe Sites, and Globe Measurements is given in the Investigation Tree table.

125.2 Standard checks for InvestigationDomain

Low value	
High value	
Unit	code
List of values	Atmo□Atmosphere Hydr□Hydrology LcBi□Land Cover/Biology Soil□Soil Meta□Metadata/Site Definition (GPS+) Phen□Phenology/Seasons Tool□Toolkit

126 InvoiceNbrDomain

126.1 Description of InvoiceNbrDomain

The InvoiceNbr domain is a set of values that represent a vendor's invoice number for a GLOBE order. It is an unstructured string in order to accommodate the styles and formats of different vendors.

126.2 Standard checks for InvoiceNbrDomain

Low value	
High value	
Unit	
List of values	

127 IssuesTextDomain

127.1 Description of IssuesTextDomain

The IssuesText domain is the set set of character strings that can be used to describe the follow-up and training issues faced by U.S. Partners in their roles as Recruiters, Trainers, and Mentors of GLOBE Teachers. Currently this text has no internal structure, but conceivably in the future it may satisfy some specific XML Document Type Definition.

127.2 Standard checks for IssuesTextDomain

Low value	
High value	
Unit	
List of values	

128 KitMfgNameDomain

128.1 Description of KitMfgNameDomain

The KitMfgName domain identifies the manufacturer of a kit used in GLOBE experiments. Usually a manufacturer is LaMotte or Hach, but other names are permitted.

128.2 Standard checks for KitMfgNameDomain

Low value	
High value	
Unit	
List of values	

129 LanguageCodeDomain

129.1 Description of LanguageCodeDomain

A two character code to represent a spoken language. Use codes standardized by WWW.

129.2 Standard checks for LanguageCodeDomain

Low value	
High value	
Unit	
List of values	

130 LanguageDomain

130.1 Description of LanguageDomain

The Language domain is a set of values to represent GLOBE supported languages. Initially, GLOBE supports: English, French, Spanish, Russian, Chinese, Arabic. This domain will likely expand to include other languages as needed to support GLOBE activities.

130.2 Standard checks for LanguageDomain

Low value	
High value	
Unit	
List of values	

131 LatitudeDomain

131.1 Description of LatitudeDomain

The Latitude domain represents the global latitude of a study site. Latitude is recorded as a real number between -90 degrees and +90 degrees, inclusive. Latitudes in the Northern Hemisphere are positive and those in the Southern Hemisphere are negative. The North Pole has latitude +90 degrees and the South Pole has latitude -90 degrees. The decimal (or binary) precision of a Latitude attribute may vary, depending upon the accuracy of the measurement device, but will usually have decimal precision of at least 3 decimal places.

131.2 Standard checks for LatitudeDomain

Low value	-90
High value	+90
Unit	degrees
List of values	

132 LeafColorDomain

132.1 Description of LeafColorDomain

The LeafColor domain is a set of codes from the GLOBE Plant color guide. Each code is a character string of codes and numbers, e.g. 5G7/12 or 2.5Y8/10

The following character codes have special meaning:

0 The leaf fell off or grass got destroyed
-1 The leaf or grass got buried in snow before reaching maturity.

132.2 Standard checks for LeafColorDomain

Low value	
High value	
Unit	
List of values	

133 LeafIdDomain

133.1 Description of LeafIdDomain

The LeafId domain is a set of identifiers assigned by GLOBE to identify a leaf of a shrub, or a blade of grass, in the Grennup/Greendown protocols. Normally the values vary from 1 to 4, but could be larger or smaller if more leaf samples are used. for trees and shrubs, the number 1 identifies the leaf furthest out on the limb, with higher numbers for each successive leaf toward the center. For grass, the numbers have no positional meaning.

If a leaf is destroyed during a growing cycle, then a higher numbered leaf identifies a replacement.

133.2 Standard checks for LeafIdDomain

Low value	1

High value	9
Unit	
List of values	

134 LeafLengthDomain

134.1 Description of LeafLengthDomain

The LeafLength domain is the length of a leaf of a tree or shrub, or the length of a blade of grass, in the GreenUp protocol.

134.2 Standard checks for LeafLengthDomain

Low value	
High value	
Unit	
List of values	

135 LevelTypeDomain

135.1 Description of LevelTypeDomain

The LevelType domain is used to classify GLOBE protocols as one of the following: Basic (BAS), Advanced (ADV), Optional (OPT), or Special (SPE).

The MEASURE CODES table identifies which level has been assigned to each protocol

NOTE: Some measure codes at a higher position in the measure code hierarchy do not map to a single LevelType, so their LevelType is left null (on display they may be identified as "Mixed") , but they will always be the parent of measure codes lower in the hierarchy that do map to a single LevelType.

135.2 Standard checks for LevelTypeDomain

Low value	
High value	
Unit	
List of values	BAS□Basic ADV□Advanced OPT□Optional SPE□Special

136 LifeCycleDomain

136.1 Description of LifeCycleDomain

The LifeCycle domain is a set of codes used to represent the various stages in the life cycle of a GLOBE School, a GLOBE Teacher, a GLOBE partner Country, or a GLOBE Partner. A LifeCycle code is always paired with a CycleChangeDate to identify the DATE at which the code was last changed. The GLOBE database does not track the history of changes to LifeCycle codes, but it does track the training history of all teachers and the timespan that any teacher was ever associated with a school.

The possible LifeCycle codes are as follows:

P	Pre-candidate	(School, Teacher, Partner)
C	Candidate	(School, Teacher, Country, Partner)
A	Active	(School, Teacher, Country, Partner)
M	adMinistrative	(School, Teacher)
T	parTner	(School, Teacher)
I	Inactive	(School, Teacher, Country, Partner)
J	Inactive	(Only used for former administrative schools)
R	Reporting	(Teacher - non-GLOBE-trained at GLOBE school)

O Other Deprecated -- and being phased out

Note that the word "Partner" is overloaded. A Partner school is different from a Partner country or a GLOBE Partner (formerly Franchise). The context determines the usage of the term as an adjective (to qualify School, Teacher, or Country), or as a noun (GLOBE Partner).

Note that Countries only use 3 of the codes, and Partners only use 4, and that the J code is reserved for use only with schools that were formerly administrative and are now inactive. The two different codes for inactive schools are necessary in order to maintain a complete separation among GLOBE schools that can enter GLOBE data and administrative schools that can only enter test data. The Reporting (R) code is reserved for the special case of non-GLOBE-trained teacher who takes over the GLOBE reporting activities at a GLOBE school. These teachers will be the subjects of an intensive recruiting effort.

Explanation

Pre-candiate (P) -- This is the status assigned to a brand new teacher or school that has just been added to the database. A new TeacherId or SchoolId has been assigned to the new entity but integrity checking to ensure that it is a valid entry and does not duplicate or obviate some previous entry may not have been fully completed. After integrity checking is completed (usually by RCG), then the status changes from "Pre-candidate" to "Candidate". The "Pre-candidate" status will likely be assigned to people who register as a potential new GLOBE teacher via self-entry on some public WWW GlobeTeacher Data Entry Page. The school entered by that person will also be assigned a status of "Pre-Candidate". This is a very temporary designation and should usually evolve to some other LifeCycle designation within a few days. In February, 2001, at Greg Cox's request, we extended the Pre-candidate (P) designation to GLOBE Partners. The intent is to be able to delineate those organizations GLOBE is talking with but have not yet begun the paperwork process of a "candidate" partnership.

Candidate (C) -- This is the status of a school or teacher that has expressed an interest in the GLOBE program, has been checked out by RCG, but has not yet completed GLOBE training, or alternatively, has not yet been issued a SchoolId and password (for split sessions). A school will not become Active until at least one teacher becomes Active, but schools will not automatically lose Active status if a teacher leaves. A Country or a Partner may hold this designation in the months before a GLOBE agreement is signed. Right now there is no attempt to delete "Candidate" schools or teachers from the database, so a teacher who registers for a workshop and fails to show up at that workshop is carried in the database as "Candidate" for an indeterminate period of time.

Active (A) -- This is the status of a teacher who has completed GLOBE training (or been issued a password for split sessions), or of a school with at least one teacher having completed GLOBE training. It is not necessary that the school has begun sending data; the sending of data is recorded in a different place in the GLOBE database. A Country or a Partner cannot have an "Active" LifeCycle until after a formal GLOBE agreement has been signed. A policy for setting codes to active is given in the Annotation section below. A teacher or school retains an Active designation for an indefinite period of time, unless explicitly changed because of retirement or an explicit request that mailings, etc. be stopped. If an active school loses its Globe trained teacher for some reason, the school does NOT automatically lose its "Active" status; an explicit action is required in order to change a school from Active (A) to Inactive (I) or back to Candidate (C).

Administrative (M) -- This is the status of GLOBE staff, trainers, scientists, advisors, international contacts and coordinators, and other people who are not classroom teachers of GLOBE students. Normally such a person will be listed in the GLOBE TEACHER table and the institution that they are associated with will be listed in the GLOBE SCHOOL table, both with an "Administrative" LifeCycle designation. The particular role that administrative teachers play with respect to the institutions they are associated with is given by the TeacherRole attribute of the TEACHER TO SCHOOL table. For example, the GLOBE scientists who are Principal Investigators are listed in the GLOBE TEACHER table, their universities are listed in the GLOBE SCHOOL table, and the TeacherRole attribute in the TEACHER TO SCHOOL table is set to "P.I.". Similarly, it is under consideration for international coordinators and contacts to be listed in the GLOBE TEACHER table, their institutional offices listed in the GLOBE SCHOOL table, and their roles in the TEACHER TO SCHOOL table be listed as "I-COORD" or "I-CONTACT", respectively. An administrative Country is one that appears in the COUNTRIES table for some administrative reason, but that is not a candidate to become a GLOBE partner country; the only occurrence at this point is the "ZZ" country designation that is used for training and testing purposes! It does not represent a real country.

Partner (T) -- This is the status of a school that is participating in the GLOBE program as the result of an agreement with some other organization. A partner school has all of the rights and privileges of a GLOBE school, but retains its affiliation with the other organization. See Dixon Butler's memo dated October 12, 1999. Similarly, a Partner (T) teacher is a teacher that is affiliated with a partner school. A partner teacher has a teacherId just like a GLOBE teacher, but there is likely no training record for that teacher in the REGISTRATION table. If a partner teacher decides to go through GLOBE training, then they would enter the P,C,A lifecycle with the intent of becoming a GLOBE teacher. Their LifeCycle would change from Partner (T) to Pre-candiate (P), Candidate (C), or Active (A), as appropriate. Similarly, upon request, any partner school could have its LifeCycle code changed from T to A if one of its teachers became a GLOBE Teacher. This notion of "partner" applies only to Schools and to Teachers; it does NOT apply to Countries or Partners. An active GLOBE Partner Country or GLOBE Partner always has a LifeCycle of Active (A).

Reporting (R) -- This is the status of a non-GLOBE-trained teacher at a GLOBE school who submits their name as a replacement for a GLOBE teacher who has left that school. Such teachers will be intensively recruited to try to get them to register for GLOBE training! These names will never appear on any list of GLOBE teachers.

Inactive (I) -- This is the status of a teacher who has retired or otherwise left the GLOBE program, of a school that has made an explicit

request that they be removed from GLOBE mailing lists, etc., or of a Country or Partner that has revoked its GLOBE agreement. If a school has reported data to the student data archive, then that school is considered to be a GLOBE school in perpetuity, even if that school's LifeCycle is changed to "Inactive" for some administrative reason. Except for schools that have reported valid data, Inactive entities should be assumed to not exist for most reporting or counting purposes.

Inactive (J) -- This is the status of a school that was formerly listed as an Administrative (M) school, but is now Inactive (J). It is necessary to have this second notion of Inactive schools in order to maintain a complete separation among GLOBE schools, that can enter GLOBE data, and administrative schools, that can only enter test data, throughout their entire life cycles!

Other (O) -- This identifies a status not covered by any of the above. This designation is currently deprecated, so every new Teacher, School, County, or Partner should be labeled with exactly one of the above LifeCycle codes, and the Date at which it became effective. There are only a handful of "Other" LifeCycle designations left in the database and they are gradually being recoded to some other appropriate LifeCycle code.

136.2 Annotation for LifeCycleDomain

LifeCycle Transitions

The following are the only valid transitions among the various LifeCycle codes for Schools, Teachers, Countries, or Partners. The keyword "Initial" is used to indicate codes that are valid for newly created entities.

Valid GLOBE School Transitions

```
Initial --> P
P --> C
P --> A
Initial --> C
C --> A
C --> I
I --> C                 -- very rare (possible if school has no data!)
Initial --> A
A --> C                 -- very rare (possible if school has no data!)
A --> I
I --> A
Initial --> T
T --> A
T --> I
I --> T
```

Valid Administrative School Transitions

```
Initial --> M
M --> J
J --> M
```

Valid Teacher Transitions

```
Initial --> P
P --> C
P --> A
Initial --> C
C --> A
C --> I                 -- very rare (normally stays C)
I --> C                 -- very rare (coming back out of retirement)
Initial --> A
A --> C                 -- rare (started but did not complete training)
A --> I
I --> A
A --> M  -- very rare (but not prohibited - see Note)
Initial --> T
T --> A
T --> I
I --> T
T --> M
Initial --> M
M --> A  -- rare (completes GLOBE training - see Note)
M --> I
I --> M
Initial --> R
R --> C
```

Note: It is normally desirable for a GLOBE teacher to retain their Active (A) status, even when they complete Train-the-Trainer workshops with the intent to train other teachers. A new candidate teacher who completes Train-the-Trainer will normally have Active (A) status rather than Administrative (M) status.

Valid Country Transitions

Initial --> C
Initial --> A
C --> A
A --> I
I --> A
Initial --> M -- reserved for very special cases!

Annotation

Here are the processes for changing lifecycle designations for schools and teachers:

US SCHOOLS

1) New teacher signs up for a GLOBE contract training (not a Partner training)
Teacher ID issued in database, LifeCycleCode=Candidate
If new school, School ID issued in database, LifeCycleCode=Candidate
If Inactive School (with no data), LifeCycleCode=Candidate
If Inactive School (with data), LifeCycleCode=Active
Otherwise, School LifeCycleCode remains unchanged

2) New teacher shows up for a training
If Partner training, Teacher ID and School ID issued in database for new teachers and schools.
Teacher and School LifeCycleCode=Pre-Candidate if new
RCG changes Teacher and School LifeCycleCode from Pre-Candidate to Candidate
Workshop marks Candidate School and/or Teacher as "ID given out"
Teacher and School LifeCycleCode=Active
School Password created in database
Teachers can log in and enter data at this point
Password is stored in the school httpinfo table
Teacher given School ID and Password: Registration RegisterCode=ID
GLOBE contract trainings hand them out on last day of training
Partner trainings hand them out on first day of training
Extended trainings hand them out after first session

3) Workshop Ends
If teacher completes the training, Registration RegisterCode=CW
If teacher does *not* complete workshop:
Registration RegisterCode=AW (Attended and Withrew)
School LifeCycleCode remains Active and password remains!! (YES -- this is correct - it is intended for teachers who want to continue entering data while they try again later to complete GLOBE training). However, schools codes can be explicity changed to whatever code is most appropriate.

INTERNATIONAL SCHOOLS

In theory this process should be the same as for US schools, but in practice, this is what happens:

1) GLOBE is contacted with list of new schools and teachers for that country
School ID and Teacher ID created
School and Teacher LifeCycleCode=Active
Registration RegisterCode=CW (Completed Workshop)
School Password created
Teachers can log in and enter data
Teachers given School ID and Password
School and Teacher LifeCycleCode remain unchanged unless requested by Country Coordinator.

Note #1

A school will show up on a Visualization map if:
it is Active OR it has data (with LifeCycle code I)

Note #2

A "Certified" teacher is one with a Registration RegisterCode=CW (Completed Workshop)

136.3 Standard checks for LifeCycleDomain

Low value	
High value	
Unit	code
List of values	P□Pre-candidate C□Candidate A□Active I□Inactive M□adMinistrative T□parTner J□Inactive (formerly M school) R□Reporting teacher O□Other (Deprecated)

137 LilacHeightDomain

137.1 Description of LilacHeightDomain

The LilacHeight domain represents the height of a lilac shrub when it is first planted at a Lilac Phenology study site It is measured in centimeters (cm) (no fractional part) and usually varies between xx and yy centimeterss.

137.2 Standard checks for LilacHeightDomain

Low value	0
High value	
Unit	cm
List of values	

138 LilacIdDomain

138.1 Description of LilacIdDomain

The LilacId domain is a set of identifiers for individual lilac shrubs.

138.2 Standard checks for LilacIdDomain

Low value	
High value	
Unit	id
List of values	

139 LilacTypeCodeDomain

139.1 Description of LilacTypeCodeDomain

The LilacTypeCode domain is a set of codes to represent lilac types. Initially there are only two types (Common and Clonal), but it is possible that additional Clonal types may be added in the future.

139.2 Standard checks for LilacTypeCodeDomain

Low value	0
High value	9

Unit	
List of values	1□Common Lilac 2□Clonal Lilac (Red Rothomagensis)

140 LocationMethodDomain

140.1 Description of LocationMethodDomain

The LocationMethod domain is a set of codes that represent the method by which the location of a study site is determined. The main purpose is to distinguish between GPS determined locations and others (e.g. Map).

Initially, the DataSource for SITE LOCATION was allowed to be 0 if Unchecked, 1 if the site was located by GPS, and 2 if the site was located by some Other (accurate) means.

In March 2000, the Data Entry page for Site Definition was modified to allow 4 choices for how the site location was determined: Default to GLOBE server coordinates (0), GPS (1), Other (2), and Default to School Location (3). One of these 4 must be chosen. The default to GLOBE Server coordinates was the previous default if the item was not checked.

For the purposes of the Honor Roll, sites must be located by GPS or Other, a Default location is not sufficient.

140.2 Standard checks for LocationMethodDomain

Low value	0
High value	9
Unit	code
List of values	0□Default to GLOBE Server coordinates 1□GPS Reading 2□Other (accurate) 3□Default to School Location

141 LongitudeDomain

141.1 Description of LongitudeDomain

The Longitude domain represents the global longitude of a study site. Longitude is recorded as a real number greater than or equal to -180 degrees and less than +180 degrees. The Prime Meridian (through Greenwich, England) has longitude 0 degrees. The 180-th Meridian (in the Pacific Ocean) has longitude -180 degrees. The International Date Line is not identical to the 180-th Meridian, although they overlap in many locations. Longitudes West of the Prime Meridian are negative and those East of the Prime Meridian are positive. By default, we assign a longitude of 0 degrees to both the North and South Poles. The decimal (or binary) precision of a Longitude attribute may vary, depending upon the accuracy of the measurement device, but will usually have decimal precision of at least 3 decimal places.

141.2 Standard checks for LongitudeDomain

Low value	-180
High value	+180
Unit	degrees
List of values	

142 MUCcodeDomain

142.1 Description of MUCcodeDomain

The MUCcode domain represents GLOBE Phase 2 codes that identify different types of land cover. It is defined by the Modified UNESCO Classification System (MUC). It replaces the Anderson code used for land cover identification in GLOBE Phase 1 protocols. There is not a direct translation between the two codes.

Low value	0000
High value	9999
Unit	code
List of values	

143 MUCdescriptionDomain

143.1 Standard checks for MUCdescriptionDomain

Low value	
High value	
Unit	
List of values	

144 MUCnameDomain

144.1 Description of MUCnameDomain

The MUCname domain represents the names (actually short descriptions) assigned to each of the MUC codes defined by the Modified UNESCO Classification Scheme (MUC). Longer descriptions with examples are carried in a different attribute.

144.2 Standard checks for MUCnameDomain

Low value	
High value	
Unit	
List of values	

145 MUCsourceDomain

145.1 Description of MUCsourceDomain

The MUCsource domain is a set of codes to indicate the source of the MUCcode measurement in the LAND COVER table. Valid sources include Ground Observation (G), which implies a physical presence at the site to determine the MUCcode, Map Derivation (M), which is interpreted from looking at a map or satellite image, Anderson Derived (A), meaning the MUC was derived from an AndersonCode, or External Site (X) metadata observations.

The External Site observations may not satisfy all of the protocol procedures for valid land cover observations, especially with regard to site size and site homogeneity, so they will not be counted as valid land cover observations for land cover counting purposes.

The Map Derivations are to be used as part of the Accuracy Assesment protocol described in the Teacher's Guide.

Anderson Derivations are deprecated as of Fall 1998.

145.2 Standard checks for MUCsourceDomain

Low value	
High value	
Unit	
List of values	G□Ground Observation M□Map Derivation X□External Site A□Anderson Derived (legacy only)

146 MailFormatDomain

146.1 Description of MailFormatDomain

The MailFormat domain is a set of codes that represent different possible formats of mail sent from GLOBE to some recipient; initial values include: Hard Copy Letter, Cable, and Electromic Mail.

146.2 Standard checks for MailFormatDomain

Low value	
High value	
Unit	code
List of values	LTR□Letter CBL□Cable EML□E-mail

147 MailStatusDomain

147.1 Description of MailStatusDomain

The MailStatus domain is a set of codes that identify the status of a piece of mail sent from GLOBE (via RCG) to Teachers, Schools, or Partners. The codes are owned by RCG and may change as circumstances dictate.

147.2 Standard checks for MailStatusDomain

Low value	
High value	
Unit	
List of values	

148 MailTopicDomain

148.1 Description of MailTopicDomain

The MailTopic domain is a set of codes that represent possible topics for correspondence sent from GLOBE. Initially, this domain represents correspondence related to international workshops and identifies correspondence from GLOBE to be one of: partner country invitation to international workshop , non-partner country invitation to international workshop, logistics letter to site coordinator in host country, or other.

In early Spring 2001, this domain was expanded to include other topics not yet fully defined in the architecture document. These include: Teacher Update mailings and Equipment mailings, etc. RCG will experiment with appropriate topic labels before they become formalized.

148.2 Standard checks for MailTopicDomain

Low value	
High value	
Unit	code
List of values	P-Invite□Partner Invitation NP-Invite□Non-Partner Invitation Logistics□Site Logistics Other□Other

149 MailingNameDomain

149.1 Description of MailingNameDomain

The MailingName domain is a unique identifier for each GLOBE mailing handled by RCG. It is a short name chosen by RCG.

For example, the name "UPD2000" identifies the calendar year 2000 mailing to all GLOBE teachers asking them for updates to their school and teacher information.

149.2 Standard checks for MailingNameDomain

Low value	
High value	
Unit	
List of values	

150 MeasurementDomain

150.1 Description of MeasurementDomain

The Measurement domain is a 2-character code used to identify a GLOBE measurement group. GLOBE measurement group codes are defined and described in the Measure Codes table. The relationship among Globe Investigations, Globe Sites, and Globe Measurements is given in the Investigation Tree table.

There is a close relationship between measurement groups and tables used to store student data measurements.

150.2 Standard checks for MeasurementDomain

Low value	
High value	
Unit	code
List of values	FD□School Front Door GPS AT□Air Temperature CO□Cloud Observation PR□Precip Rain PS□Precip Solid SW□Surface Water LC□Land Cover SI□Species Identification TB□Tree Biometry GB□Grass Biometry SM□Soil Moisture SC□Soil Characterization ST□Soil Temperature SF□Soil Infiltration BB□Bud Burst PH□Phenology SP□Site Photo

151 MinuteDomain

151.1 Description of MinuteDomain

The minute domain represents a minute (in UT time) of an hour in the 24-hour day. It varies between 0 and 59 (See also the HourDomain).

151.2 Standard checks for MinuteDomain

Low value	0
High value	59
Unit	minute

List of values	

152 ModelNameDomain

152.1 Description of ModelNameDomain

The ModelName domain identifies a specific model of a kit from a manufacturer identified in the KitMfgName domain.

152.2 Standard checks for ModelNameDomain

Low value	
High value	
Unit	
List of values	

153 MoistureEstDomain

153.1 Description of MoistureEstDomain

The MoistureEst domain is a set of codes that represent a gross estimate of the moisture conditions in a soil sample. The current values are: Dry, Moist, Wet. When a more accurate calculation of the water content of a soil sample is required, the WaterContent domain is used.

153.2 Standard checks for MoistureEstDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Dry 2□Moist 3□Wet

154 MoistureProtocolDomain

154.1 Description of MoistureProtocolDomain

The MoistureProtocol domain is a set of codes that represent the different protocols for measuring water content of a soil sample. The choices are: Star Gravimetric, Transect Gravimetric, Hole Gravimetric, and Calibrated Gypsum Block. Additional choices have been added subsequently.

A new protocol for Watermark Block Probes was added in Spring 1999.

Several schools are using other experimental Block protocols, so a new option for Other Block was added in September 1999. The Other Block protocol is to be treated just like Gypsum Block or Watermark Block in any analysis or visualizations.

See Dixon's email from Feb 1, 2002, for possible additional "sensor types" to be used in the future, including: "watermark - Delmhorst" or "watermark - Irrometer" or "watermark - Watchdog"

Added the following from Jim Washburn (via Mike Turpin) Feb 19, 2002:

Please enter the Manufacturer/Brandname of the meter you use to read the Watermark sensor:
- Delmhorst (handheld) Range 0-100 arbitary
- Irrometer/Watermark (handheld) Range 0-200 "centibars"
- Spectrum/Watchdog (logger) Range 0-200 "centibars"
- Other

Low value	0
High value	9
Unit	code
List of values	0□Unchecked (deprecated) 1□Star Gravimetric 2□Transect Gravimetric 3□Hole Depth Gravimetric 4□Gypsum Block (deprecated) 5□Delmhorst/Watermark 6□Other Block 7□Irrometer/Watermark 8□Spectrum/Watchdog (logger)

155 MoneyDomain

155.1 Description of MoneyDomain

The Money domain represents a currency amount is some separately identified currency. This domain does NOT maintain the currency identifier; instead, the currency identifier is carried in a separate attribute.

155.2 Standard checks for MoneyDomain

Low value	
High value	
Unit	
List of values	

156 MotivationTextDomain

156.1 Description of MotivationTextDomain

The MotivationText domain is the set set of character strings used by U.S. Partners to describe their motivation for becoming a GLOBE Partner. Currently this text has no internal structure, but conceivably in the future it may satisfy some specific XML Document Type Definition.

156.2 Standard checks for MotivationTextDomain

Low value	
High value	
Unit	
List of values	

157 MovingWaterDomain

157.1 Description of MovingWaterDomain

The MovingWater domain is a set of codes that identify various types of moving water. The current accepted values to identify moving water type at a surface water study site are: River, Stream, or Other. This domain may evolve to include other types of moving water.

157.2 Standard checks for MovingWaterDomain

Low value	0
High value	9
Unit	code

List of values	0□Unchecked 1□Stream 2□River 3□Other

158 NameDomain

158.1 Description of NameDomain

The Name domain is a character string data type that allows names of normal length, e.g. up to 30 characters. Longer names may use the FullName domain, and ISO10646 or UNICODE names may use the FullName2 domain. The prefix and suffix of a person's name should use the PrefixDomain and SuffixDomain domains , respectively.

158.2 Standard checks for NameDomain

Low value	
High value	
Unit	
List of values	

159 NameTypeDomain

159.1 Description of NameTypeDomain

The NameType domain is a code used to identify whether or not a teacher put their name on a teacher evaluation form. If the NameType is "null" then the teacher put their name on the evaluation form and the TeacherId identifies that teacher. if the NameType is R then the teacher did not put their name ondthe form and theTeacherId is "randomly" assigned to be one of the teachers who attended the workshop.

159.2 Standard checks for NameTypeDomain

Low value	
High value	
Unit	
List of values	R□Randomly Assigned TeacherId

160 NbrItemsDomain

160.1 Description of NbrItemsDomain

The NbrItems domain identifies different samples in a GLOBE student experiment. It is used as the primary key in a number of worksheet (WS) tables that maintain multiple raw data measurements of the same phenomenon. It is also used in other attributes that involve counts of occurrences or other happenings.

160.2 Standard checks for NbrItemsDomain

Low value	0
High value	99
Unit	cardinal
List of values	

161 NbrSchoolsDomain

161.1 Description of NbrSchoolsDomain

The NbrSchools domain identifies the total number of schools satisfying a given classification. In the CountryInfo table it is used to

approximate the total number of K-12 schools in the entire country.

161.2 Standard checks for NbrSchoolsDomain

Low value	0
High value	99999
Unit	cardinal
List of values	

162 NitrateDomain

162.1 Description of NitrateDomain

The Nitrate domain represents nitrite or nitrate concentration in a surface water sample. In the Hydrology SurfaceWater table, it is used to measure Nitrate2ppm, NitriteNitrate, and NitriteOnly. It is measured in mg/L (equiv to ppm) and usually varies between 0 and 10.00 mg/L.

162.2 Annotation for NitrateDomain

On aug 19, FSL announced that Nitrate and Nitrite measurements were being stored with only one decimal point of precision, but fields in the range checks document swere maintained at NUMBER(4,2).

162.3 Standard checks for NitrateDomain

Low value	0.0
High value	10.0
Unit	mg/L=ppm
List of values	

163 ObscuredDomain

163.1 Description of ObscuredDomain

The Obscured domain is a set of codes to identify the reason that the sky might be obscured so that a CloudCover observation cannot be taken. If CloudCover is not "Obscured" then this value could be Unchecked (0), or it could indicate the reason for mild obscured view that didn't interfere with a CloudCover reading; if CloudCover is "Obscured" then this value should identify one of the choices -- Fog, Smoke, Haze, Volcanic Ash, Dust, Sand, Spray, Heavy Rain, Heavy Snow, or Blowing Snow.

163.2 Standard checks for ObscuredDomain

Low value	0
High value	10
Unit	code
List of values	0□Unchecked 1□Fog (FG) including Mist and Drizzle 2□Smoke (FU) 3□Haze (HZ) 4 □Volcanic Ash (VA) 5□Dust 6□Sand 7□Spray (PY) 8□Heavy Rain (RA) 9□Heavy Snow (SN) 10□Blowing Snow (BLSN)

164 ObstacleDistDomain

164.1 Description of ObstacleDistDomain

The ObstacleDist domain represents the approximate distance in whole meters from an Atmospheric Site location to the nearest building or tree.

164.2 Standard checks for ObstacleDistDomain

Low value	0
High value	1000
Unit	m
List of values	

165 OrderNumberDomain

165.1 Description of OrderNumberDomain

The OrderNumber domain is a set of values that represent order numbers for GLOBE orders. It is used as the primary key of the GlobeOrder table. An order number is assigned by GLOBE to track its own orders. Values of this domain are used in several tables to help track the process of ordering materials for GLOBE workshop sessions or the sending of equipment or other supplies to a GLOBE school.

165.2 Standard checks for OrderNumberDomain

Low value	
High value	
Unit	
List of values	

166 OrderTypeDomain

166.1 Description of OrderTypeDomain

The OrderType domain is a set of codes that represent the different types of GLOBE orders. Orders can be of the following different types:

Warehouse to Workshop - An order is sent to a warehouse asking that materials or equipment be sent to a workshop session. Such orders will decrease the QuantityOnHand in the GlobeProduct table. Sometimes such orders will be sent to an intermediate place, e.g. GLOBE, a Franchise, or a Country Coordinator, for subsequent distribution to a workshop session.

Vendor to Warehouse - An order is sent to a source vendor asking that materials be sent to a warehouse. Such orders will increase the QuantityOnHand in the GlobeProduct table

Vendor to Workshop - An order is sent to a source vendor asking that materials be sent directly to a workshop session. Such orders will have no effect on the QuantityOnHand in the GlobeProduct table. This type of order should be quite rare!

166.2 Standard checks for OrderTypeDomain

Low value	
High value	
Unit	code
List of values	WH-WS Warehouse to Workshop VR-WH Vendor to Warehouse VR-WS Vendor to Workshop

167 OzoneMethodDomain

167.1 Description of OzoneMethodDomain

The OzoneMethod domain identifies the protocol used to estimate the Ozone present in the atmosphere. At present there is only one method used; it is described in the GLOBE 2000 Teacher's Guide.

167.2 Standard checks for OzoneMethodDomain

Low value	1
High value	1
Unit	code
List of values	1□GLOBE 2000 Method

168 OzonePPBDomain

168.1 Description of OzonePPBDomain

[NEED TEXT] -- is it represented as an Integer? What is range of values?

168.2 Standard checks for OzonePPBDomain

Low value	0
High value	260
Unit	PPB
List of values	

169 PETdomain

169.1 Description of PETdomain

The PET domain represents the Potential EvapoTranspiration (PET) measurement of moisture availability in an environment. It is measured by comparing the input of water tonuhe surface of the Earth with the output of the amount of water that could leave the surface. Water input is measured by total precipitation (in mm), including both rain and snow. Water output is determined by Evaporation and by Transpmration. Transpiration is the process of water loss from plants while they absorb CO2 for photosynthesis. A simple estimate of PET for GLOBE purposes is given by a pre-determined relationship dependent only upon the average daily tempereture for each day. It is measured in mm/day. The daily values are then summed over a 30-day period to give a total PET (in mm) for any given 30-day period.

PET is used in the calculation of Water Difference (WD) as explained in the WaterDiff domain.

169.2 Standard checks for PETdomain

Low value	-420.00
High value	45000.00
Unit	mm
List of values	

170 PaperDocLocDomain

170.1 Description of PaperDocLocDomain

The PaperDocLoc domain represents possible locations of a hard copy paper document. It is defined as a free flowing text field so that any description of a document's location can be entered.

170.2 Standard checks for PaperDocLocDomain

Low value	
High value	
Unit	

List of values	

171 PartnerDomain

171.1 Description of PartnerDomain

The Partner doamin is the set of identifiers for US GLOBE Partners, a re-naming of the former GLOBE Franchises. In order to avoid changes in the database, all former tables will retain FranchiseId as the identifier, but all new tables involving US Partners will use PartnerId. In all cases where a Franchise and a Partner are the same entity, the PartnerId and FranchiseId will be tyhe same identifier.

171.2 Standard checks for PartnerDomain

Low value	
High value	
Unit	
List of values	

172 PercentDomain

172.1 Description of PercentDomain

The Percent domain represents a percentage or other ratio of two numbers.

172.2 Standard checks for PercentDomain

Low value	0.0
High value	100.0
Unit	percenta
List of values	

173 PersonIdDomain

173.1 Description of PersonIdDomain

The PersonId domain represents character string identifiers for a person. It is used as the primary key of any table that identifies a single person, especially the GlobeTeacher and GlobeTrainer tables. We would like to keep PersonId's unique across all tables in the distributed database and would like to be able to insert a teacher record into the Trainer table without having an Id conflict. However, no constraint at this time will enforce this requirement.

At some point in the future we may want to define a Person table, with Globe Teacher and Globe Trainer, and possibly others, as subtables.

The format for generating this identifier is to use 2 characters and 6 digits. The 2 characters will be used to disambiguate among various different sources that may be assigning Person Id's so that Id's can be assigned in parallel.

173.2 Standard checks for PersonIdDomain

Low value	
High value	
Unit	
List of values	

174 PhotoSubmittedDomain

174.1 Description of PhotoSubmittedDomain

The PhotoSubmitted domain records the answer of a school when asked the question of whether or not photos of a site have been submitted to GLOBE. An answer to this question is optional and the result is stored in the database only when the school says that photos have been Previously Submitted (1), with or without an explanation, or says that they are Not Submitted (0), usually with some text explaining their current status.

174.2 Standard checks for PhotoSubmittedDomain

Low value	
High value	
Unit	
List of values	0 Not Submitted 1 Previously Submitted

175 PhotomVoltageDomain

175.1 Description of PhotomVoltageDomain

The PhotomVoltage Domain is a range of values used to record the dark voltage and the sunlight voltage of each channel of a multi-channel photometer. Normally the Sunlight voltage has at most 3 decimal digits of accuracy and Dark voltage has at most 4. For any given measurement the Sunlight voltage must be greater than the Dark voltage.

No collection of voltage measurements from the same group and the same channel should differ by more than 10%.

175.2 Standard checks for PhotomVoltageDomain

Low value	0
High value	0.05
Unit	volts
List of values	

176 PhotometerDomain

176.1 Description of PhotometerDomain

The Photometer Domain is the set of identifiers used to identify one of the photometers approved for use by GLOBE schools. The format is XXX-dddd, where X is a character and d is a digit. A photomoter must be calibrated before it can be used, and the calibration values are stored in the PHOTOMETER CALIB table.

176.2 Standard checks for PhotometerDomain

Low value	
High value	
Unit	
List of values	

177 PlantedFlagDomain

177.1 Description of PlantedFlagDomain

The PlantedFlag domain is set to 1 if the DatePlanted field in LILAC METADATA is null. This covers the choice on the Lilac Data Entry page that says the tree was planted before 1997.

177.2 Standard checks for PlantedFlagDomain

Low value	
High value	

Unit	
List of values	

178 PostActionItemDomain

178.1 Description of PostActionItemDomain

The PostActionItem domain is a set of values that identify action items that must be completed after each workshop session is completed. The action items may differ for each workshop type (i.e. CONT, FIT, GTTT, INT-R)) and new action items may be created at any time by the administrators of each workshop type.

Initially, the post-workshop action items are: Roster, Report, Evaluation, Equipment

178.2 Standard checks for PostActionItemDomain

Low value	
High value	
Unit	
List of values	

179 PostalDomain

179.1 Description of PostalDomain

The Postal domain represents the postal code of a mailing address in any country. It consists of character strings sufficiently long to maintain the postal codes of all countries of interest for GLOBE mailings.

179.2 Standard checks for PostalDomain

Low value	
High value	
Unit	
List of values	

180 PreActionItemDomain

180.1 Description of PreActionItemDomain

The PreActionItem domain is a set of values that identify action items that must be completed prior to each workshop session. The action items may differ for each workshop type (i.e. CONT, FIT, GTTT, INT-R) and new action items may be created at any time by the administrators of each workshop type.

Initially, the pre-workshop action items are:

Agenda <describe what it is>

Contact <describe what it is>

Letter <describe what it is>

180.2 Standard checks for PreActionItemDomain

Low value	
High value	
Unit	
List of values	

181 PrecipDaysDomain

181.1 Description of PrecipDaysDomain

The PrecipDays domain represents the numbers of days elapsed since the last precipitation reading (rain or solid) was made. It varies from 1 to usually less than 7 days.

181.2 Standard checks for PrecipDaysDomain

Low value	1
High value	99
Unit	days
List of values	

182 PrefixDomain

182.1 Description of PrefixDomain

The Prefix domain is a character string data type of up to 10 characters to represent the prefix of a person's name. Normal prefixes are Mr., Mrs., Ms., Dr., Sr., Lt. Col., Herr, etc. At the present time the domain is not restricted to a fixed set of legal prefixes. The suffix to a person's name should use the SuffixDomain domain.

182.2 Standard checks for PrefixDomain

Low value	
High value	
Unit	
List of values	

183 PressureMethodDomain

183.1 Description of PressureMethodDomain

The PressureMethod domain identifies the method used to determine a barometric pressure. As of March 2001, GLOBE uses only one method, identified by the number 1.

183.2 Standard checks for PressureMethodDomain

Low value	
High value	
Unit	
List of values	

184 PrevalenceDomain

184.1 Description of PrevalenceDomain

The Prevalance domain is a set of codes that represent the prevalence of a given species at a species identification site. At the current time, prevalence can be either Dominant or Co-Dominant; it is possible that this domain will evolve to include other prevalence classes (e.g. Sub-Dominant, Moderate Instances, Rare Instances), or it is possible that a new numeric domain will evolve that tries to quantify the prevalence (e.g. from 1 to 9) of a given species at a given study site.

184.2 Standard checks for PrevalenceDomain

Low value	

High value	
Unit	code
List of values	DM□Dominant CD□Co-Dominant SD□Sub-Dominant SI□Some Instances RI□Rare Instances

185 ProductCodeDomain

185.1 Description of ProductCodeDomain

The ProductCode domain is code that represents all of the various products that are of interest to GLOBE operations, especially products that get sent to teacher training workshops. The product codes are defined and described in the Globe Product table.

185.2 Standard checks for ProductCodeDomain

Low value	
High value	
Unit	
List of values	

186 Protocol ListDomain

186.1 Description of Protocol ListDomain

The ProtocolList domain contains comma-separated lists of protocol measurements taken from the Measurement domain, i.e. " AT,CO,PR", "SW,SM", etc.

In the GLOBE TRAINER table, the ProtocolsBySelf attribute is a list of protocol measurements that the Trainer claims to be able to teach, and the ProtocolsByGlobe attribute is a list of protocol measurements that GLOBE has recognized as those the Trainer is authorized to teach.

186.2 Standard checks for Protocol ListDomain

Low value	
High value	
Unit	
List of values	

187 QualityOfLangDomain

187.1 Description of QualityOfLangDomain

The QualityOfLanguage domain identifies several levels of language capability for a speaker of that language. The initial levels are Capable (C), Fluent (F), and Native (N).

187.2 Standard checks for QualityOfLangDomain

Low value	
High value	
Unit	
List of values	C□Capable F□Fluent N□Native

188 QuantityDomain

188.1 Description of QuantityDomain

the quantity domain represents a quantity of items that might vary from 0 to 999,999. It represents the quantity on hand for items in a watrehouse.

188.2 Standard checks for QuantityDomain

Low value	
High value	
Unit	
List of values	

189 QueryStatusDomain

189.1 Description of QueryStatusDomain

The QueryStatus Domain is a set of codes that identify the type of queries that a country contact should receive. For now, the Help Desk will treat the column with "Y" values as "yes" the identified contact receives all queries from anyone in that country. We'll leave the value "null" to mean "no" the person doesn't receive any queries. The only acceptable values are "null" and "Y". This column will always be "owned" by the Help Desk for its specific use.

Later, if it would be profitable to classify different types of queries, the GLOBE HQ and/or the Help Desk could define codes for the new classifications without impacting any other application. The "null" value will always mean "no queries" and "Y" will always mean "all queries". Any other character will represent a specific subset of query types, e.g. "A" for "Atmosphere", "D" for "aDministrative", "S" for "all science", etc. with any combination receiving a different character code.

In September 2001, the Help Desk decided to define two new values for QueryStatus. The value "B" means that the contact receives a blind copy of all queries, and the value "C" means that the contact receives a regular copy of all queries. This change is made to accommodate a request from Canada that certain contacts receive ONLY a blind copy of queries.

189.2 Standard checks for QueryStatusDomain

Low value	
High value	
Unit	
List of values	Y□Email/Yes B□Email/BlindCopy C□Email/Copy

190 QuestAnswerDomain

190.1 Description of QuestAnswerDomain

The QuestAnswer domain identifies the Domain of the answer to some question in a GLOBE evaluation form. Initially the answer domains include:

Agree	1=SD, 2=D, 3=A, 4=SA ([Strongly] Agree or Disagree)
Prepared	1=Unprepared, 2=Slightly, 3=Somewhat, 4=Full Prepared
Rating	A rating that varies from 1 to 10 where 10 is best.
Text	A textual response to a question (uses the Comments domain).

For Train-the-Trainer forms a new AnswerType of "Effective" was created. Since Trainer-the-Trainers are already experienced, it makes more sense to ask them if the training was effective rather than if it prepared them.

Effective will be handled the same as Prepared for statistical analysis.

Low value	
High value	
Unit	code
List of values	Agree□EvalAgree Domain Prepared□EvalPrepared Domain Rating□EvalRating Domain Text□Comments Domain Effective□Domain

191 QuestOrderDomain

191.1 Description of QuestOrderDomain

The QuestionOrder domain identifies the order in which questions are asked on an evaluation form. TheQuestionOrder uniquely identifies the question order for questions associated with each FormId in the Evaluation Form table.

191.2 Standard checks for QuestOrderDomain

Low value	
High value	
Unit	
List of values	

192 QuestTargetDomain

192.1 Description of QuestTargetDomain

The QuestionTarget domain is a set of codes to identify the type of GLOBE evaluation that a question is intended to address. Initially, there are only four types: TE for questions addressed to Teachers who have attended a workshop, TR for questions addressed to Trainers who have taught at a workshop session, TT for questions intended participants at a Train-the-Trainer workshop, and XX for questions that could be asked of anyone.

192.2 Standard checks for QuestTargetDomain

Low value	
High value	
Unit	code
List of values	TE□Teacher Question TR□Trainer Question TT□Teacher&Trainer Question

193 QuestVersionDomain

193.1 Description of QuestVersionDomain

The QuestionVersion domain identifies the version of a guestion. A new version of a question should cover the same topic area, and have the same type of answer, but the text of the question may vary with each version.

193.2 Standard checks for QuestVersionDomain

Low value	0
High value	9
Unit	version

List of values	

194 QuestionTextDomain

194.1 Description of QuestionTextDomain

The QuestionText domain identifies the text of questions used in the questions for Teacher or Trainer evaluations of workshops.

194.2 Standard checks for QuestionTextDomain

Low value	
High value	
Unit	
List of values	

195 RainfallDomain

195.1 Description of RainfallDomain

The Rainfall domain represents the amount of rain collected in a rain gauge. It is measured in millimeters (mm) and usually varies between 0.0 and 1500.0 mm on a daily basis. This domain may not be appropriate for identifying total rain accumulation over an extended period of time.

195.2 Standard checks for RainfallDomain

Low value	0.0
High value	9999.9
Unit	mm
List of values	

196 RecruitCodeDomain

196.1 Description of RecruitCodeDomain

The RecruitCode domain is a set of codes that represent the various ways in which a person or school is recruited to the GLOBE program. The code identifies only a broad category of recruitment possibilities; a separate RecruitText domain allows a short text string to identify more recruitment detail.

Initially, the recruit classifications are: Site (S), Web (W), Globe (G), or other (O).

196.2 Standard checks for RecruitCodeDomain

Low value	
High value	
Unit	code
List of values	S□Site W□Web G□Globe O□Other

197 RecruitTextDomain

197.1 Description of RecruitTextDomain

The RecruitText domain is a set of short text strings that allow further description of an associated RecruitCode domain. The RecruitCode identifies only a broad category of recruitment possibilities; whereas the separate RecruitText allows a short text string to

identify more recruitment detail, e.g. the name of a person, site, or franchise.

197.2 Standard checks for RecruitTextDomain

Low value	
High value	
Unit	
List of values	

198 RegTypeCodeDomain

198.1 Description of RegTypeCodeDomain

The RegTypeCode domain is a collection of codes to identify the type of teacher registration received for a workshop session. Initially the codes are for Fax, E-Mail, Phone, Web, and Walk In.

Status as of 6/8/98:

Looks like we can finally close out this issue and achieve a non-NULL value for RegTypeCode in the REGISTRATION table.

As I see it, we can do the following with a clean conscience:

1) Assume that all existing non-NULL RegTypeCode values are correct.

2) Karlene choose a Date before which she "knowns" that all RegTypeCode's should be "FX" (for Fax or other hard copy kept at GLOBE HQ) and communicate that Date to Mike.

3) Mike Update the Registration table to set all NULL RegTypeCode's to "FX" where StatusChgDate is less than the Date from Karlene. Unfortunately, there may be some ambiguity here because some StatusChgDate's may be set to a relatively recent Date when we did a bulk load from the old Paradox database, or may still be NULL. But most of those should now match up with the dates of the identified workshop session, at least for those that completed the workshop. Mike and Karlene agree on a best case strategy and execute it.

4) Mike Update the Registration table to set all remaining NULL RegTypeCode's to "UK" (for legacy unknown) and let the rest of us know how many fall into that category.

5) Mike write a constraint to enforce the rule that RegTypeCode be NOT NULL, and also make other modifications as necessary to ensure that the WWW Registration Data Entry Page handles these codes properly and that "UK" cannot be used for new registrations.

6) Len document all of the above in the data dictionary.

198.2 Standard checks for RegTypeCodeDomain

Low value	
High value	
Unit	code
List of values	FX□Fax EM□E-Mail PH□Phone WB□Web WI□Walk In UK□Unknown (legacy only)

199 RegionCategoryDomain

199.1 Description of RegionCategoryDomain

The Category domain is used to categorize various geographic entities identified in the GEOGRAPHIC REGIONS table. For example, one category of Regions might be Continents, anothert might be Mountains, another might be Crops, etc.

The range of values permitted for Category should be very flexible so the GLOBE administrators, Scientists, and PI's, can have freedom

to define new Regions and new categories as needed.

199.2 Standard checks for RegionCategoryDomain

Low value	
High value	
Unit	
List of values	

200 RegionTypeDomain

200.1 Description of RegionTypeDomain

The RegionType domain gives the name of the type of political subdivision for a region of a country, e.g.: State, Province, Territory, District, Protectorate, etc.

200.2 Standard checks for RegionTypeDomain

Low value	
High value	
Unit	
List of values	

201 RegisterCodeDomain

201.1 Description of RegisterCodeDomain

The RegisterCode domain is a set of codes that represent different status conditions in the registration procedure for a teacher to attend a workshop session. Initially the codes identify a teacher's preference for one or more workshop sessions, later the codes track various stages in the registration process, finally the codes identify whether a teacher withdrew or completed the workshop session, and if academic credits were possible, indicates whether the requirements for credit were successfully completed.

When the RegisterCode domain is used for Trainers. i.e. in the Trainer Training table, then the registration preference codes are not used; instead, attendance is by invitation only.

201.2 Standard checks for RegisterCodeDomain

Low value	
High value	
Unit	code
List of values	P1□First Preference P2□Second Preference P3□Third Preference TR□Tentative Registration RF□Received Fee (meals) RC□Registration Canceled NS□No Show ID□School ID Sent AW□Attended Workshop CW□Completed Workshop

202 ResponsibilityDomain

202.1 Standard checks for ResponsibilityDomain

Low value	

High value	
Unit	
List of values	

203 RingDiamDomain

203.1 Description of RingDiamDomain

The RingDiam domain represents the diameter of the inner and outer rings in a soil infiltration experiment. It is measured in cm with up to one decimal of precision.

203.2 Standard checks for RingDiamDomain

Low value	0
High value	
Unit	cm
List of values	

204 RockEstDomain

204.1 Description of RockEstDomain

The RockEst domain is a 1-digit code that represents a gross estimate of the number of rocks present in a soil sample. It has the values: None, Few, Many.

204.2 Standard checks for RockEstDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□None 2□Few 3□Many

205 RockVolumeDomain

205.1 Description of RockVolumeDomain

The RockVolume domain represents the volume of the rocks contained in a soil sample taken from a soil characterization study site. This volume is estimated by measuring the volume of water displaced in a beaker when rocks are added to it. All measurements (water, rocks+water, and rocks) for this experiment are in milliLiers (mL), or equivalently (cm3), and usually vary between 0 and 400 mL. The volume of the soil sample is measured by the separate SoilVolumeDomain.

205.2 Standard checks for RockVolumeDomain

Low value	0
High value	800
Unit	mL = cm3
List of values	

206 RoleDomain

Low value	
High value	
Unit	
List of values	

207 RootEstDomain

207.1 Description of RootEstDomain

The RootEst domain is a 1-digit code that represents a gross estimate of the number of roots present in a soil sample. It has the values: None, Few, Many.

207.2 Standard checks for RootEstDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□None 2□Few 3□Many

208 RotClassDomain

208.1 Standard checks for RotClassDomain

Low value	
High value	
Unit	
List of values	1□Class 1 - newly fallen logs 2□Class 2 - still contain bark 3□Class 3 - no bark, sap, or live tissue 4□Class 4 - rottten but intact 5□Class 5 - fully rotten

209 SalinityDomain

209.1 Description of SalinityDomain

The Salinity domain represents the salinity of a water sample. It is measured in ??? (ppt) and usually varies between 0.0 and 50.0.

209.2 Standard checks for SalinityDomain

Low value	0.0
High value	50.0
Unit	ppt
List of values	

210 SalinityMethodDomain

210.1 Description of SalinityMethodDomain

The SalinityMethod domain represents the method by which a surface water salinity measurement is taken. It can be one of: Hydrometer Method or Titration Method.

210.2 Standard checks for SalinityMethodDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Hydrometer Method 2□Titration Method

211 SaltOrFreshDomain

211.1 Description of SaltOrFreshDomain

The SaltOrFresh domain is a set of codes to identify whether a surface water study site is a fresh water site or a salt water site.

211.2 Standard checks for SaltOrFreshDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Salt Water 2□Fresh Water

212 SalutationDomain

212.1 Description of SalutationDomain

The Salutation domain consists of strings that represent a short form of a person's name. It is often used as the salutation line in a letter or as an informal association of the person with some other GLOBE entity (e.g. Teacher associated with Class).

212.2 Standard checks for SalutationDomain

Low value	
High value	
Unit	
List of values	

213 SchoolFundingDomain

213.1 Description of SchoolFundingDomain

The SchoolFunding domain is a pair of numeric codes that indicate if the School is Publicly Funded or Privately funded. This is an important piece of information because sometimes donations given to GLOBE are intended only for Publicly Funded schools.

213.2 Standard checks for SchoolFundingDomain

Low value	0
High value	9
Unit	code
List of values	0□None indicated 1□Publicly Funded 2□Privately Funded

214 SchoolIdDomain

214.1 Description of SchoolIdDomain

The SchoolId domain consists of codes that represent the identity of a GLOBE school. It is used as the primary key of the GlobeSchool table and as a foreign key in many other tables.

At present the Globe Master Database (in Paradox) and the FSL student data tables (in Oracle) are using different identifiers for SchoolID. The Master DB uses an integer (called AppNum) generated when a new school record is entered into the Paradox database and the FSL database uses an 8-character code with the first two characters the country code of the home country of the school, and in the U.S. with the next two characters identifying the home state of the school, and the last four characters randomly generated. The FSL school identifier is then kept in the Master DB as the SiteId attribute. The GLOBE integrated database will use the FSL format, but will retain the Paradox format in the GlobeSchool table as an attribute of legacy interest.

214.2 Standard checks for SchoolIdDomain

Low value	
High value	
Unit	
List of values	

215 SchoolNameDomain

215.1 Description of SchoolNameDomain

The SchoolName domain consists of strings that represent the complete and official name of a GLOBE school.

215.2 Standard checks for SchoolNameDomain

Low value	
High value	
Unit	
List of values	

216 SchoolTypeDomain

216.1 Description of SchoolTypeDomain

The SchoolType domain is a set of codes that identify the "best" classification of the school type. The list of possible types include: No Special Type (this is the default), Charter School, Magnet School, Home School, District Office, Science Center, Museum, State or National Park, School for Special Needs, GLOBE Campaign. Privately funded schools are not classified by the type of private funding they receive or by the religious group they might be affiliated with.

The school is asked to choose the single type that BEST describes their school.They should not check multiple types, even though that might make sense for some schools.

In September of 2000, Mike, Avnish and Len decided to identify DoD APO schools by using the Globe School.ParentOrganization column. If the school is a DoD school, then ParentOrg='USDOD'. This convention was not well advertised, so other conventions were used after that, including using AddressLine2 to make the designation. In August 2001, we're re-evaluating whether the existing convention is adequate, or if we should define a new SchoolType category for such schools. See email discussions from 20 August 2001 for resolution.

216.2 Standard checks for SchoolTypeDomain

Low value	0
High value	9
Unit	code

List of values	0□No Special Type 1□Charter School 2□Magnet School 3□Home School 4□District Office 5□Science Center 6□Museum 7□State or National Park 8□School for Special Needs 9□GLOBE Campaign 10□Other

217 SeasonDomain

217.1 Description of SeasonDomain

The Season domain is a code that presents various parts of the calendar year. For example, SU97 represents the Summer of 1997, FA97 represents the Fall of 1997, and SP98 represents the Spring of 1988. Months 1-5 of the calendar year are identified by the code SP, months 6-8 by the code SU, and months 9-12 by the code FA.

The special code, TBD, indicates that the session is tentatively planned for sometime in the future.

The special code, LGCY, indicates that the workshop is a legacy workshop; i.e., a pseudo workshop to collect together all teachers certified as trained by the country, but where explicit workshop information is unknown.

There are no Year 2000 issues here because this code treats Season as a category, not as a date. It should never be used for chronolocal ordering of the rows or any other date-related operation
.

217.2 Standard checks for SeasonDomain

Low value	
High value	
Unit	
List of values	

218 ServiceTypeDomain

218.1 Standard checks for ServiceTypeDomain

Low value	
High value	
Unit	code
List of values	PS□Pre-Service IS□In-Servicxe PI□Both Pre-Service and In-Service

219 SessionDomain

219.1 Description of SessionDomain

The Session domain consists of codes that represent a GLOBE teacher's training workshop session held at a specific time and place. It serves as the primary key of the WorkshopSession table.

The format of a Session identifier is 2 characters followed by 3 digits. The characters are used to disambiguate among various generators of Session Id's so that they may be generated in parallel.

Low value	
High value	
Unit	
List of values	

220 SiteCodeDomain

220.1 Description of SiteCodeDomain

The SiteCode domain is a set of codes to represent the different types of study or sample sites defined and described in the GLOBE Teachers Guide. The codes are identified and described in the Site Codes table. The relationship among Globe Investigations, Globe Sites, and Globe Measurements is given in the Investigation Tree table.

A special SiteCode is used to identify the Main Entrace of a GLOBE School as described by the GPS Investigation protocol of the GLOBE Teachers Guide. The special site code for the main entrance is SCH and the measurement code for that measurement is FD for Front Door. All GPS readings captured by the School Location Data Entry Sheet will be stored in the Site Location table with SiteId = 'SCH-01', and the Data Activity table will record that action with siteId = 'SCH-01' and with MeasureCode = 'FD'.

In November, 1998, the Land Cover Site (LCS) code was split into two more explicit codes:

LCL Land Cover Qualitative
LCN Land Cover Quantitative

220.2 Standard checks for SiteCodeDomain

Low value	
High value	
Unit	
List of values	SCH School Location ATM Atmosphere Site SWS Surface Water Site SMS Soil Moisture Site SCS Soil Characterization Site BIO Biology Site LCS Land Cover Site (Deprecated since Nov '98) LCL Land Cover Qualitative LCN Land Cover Quantitative PHN Phenology Site LIL Phenology Site for Lilacs

221 SiteDistanceDomain

221.1 Description of SiteDistanceDomain

The SiteDistance domain represents the distance between two GLOBE school study sites. The distance is measured in meters and is always a non-negative value.

221.2 Standard checks for SiteDistanceDomain

Low value	0.0
High value	99999.9
Unit	m
List of values	

222 SiteProximityDomain

222.1 Description of SiteProximityDomain

The SiteProximity domain is a set of codes used for a soil characterization study site location to show its proximity to other given sites, e.g. near Soil Moisture, near Atmosphere, near Soil Moisture and Atmosphere, In or near Biology, or Other.

If Other is specified, then a short description of the type of method should be specified and maintained in the accompanying SiteProximityOther column.

222.2 Standard checks for SiteProximityDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Near Soil Moisture site 2□Near Atmosphere site 3□Near Soil Moist and Atmosphere sites 4□In or near Biology site 5□Other

223 SkyColorEstDomain

223.1 Description of SkyColorEstDomain

The SkyColorEst Domain is a set of values used by a human to estimate sky color. Color will be on a scale from 1 to 5 representing Deep Blue to Milky White. Sky Color is not a scientific measure or color; instead it represents a human impression of color and as such may vary depending on local expectations. See the discussion of Sky Color in the Teacher's Guide.

223.2 Standard checks for SkyColorEstDomain

Low value	1
High value	5
Unit	code
List of values	1□Deep Blue 2□Blue 3□Light Blue 4□Pale blue 5□Milky

224 SlopeDomain

224.1 Description of SlopeDomain

The Slope domain represents the slope (in degrees) of a soil characterization study site. It varies between 0 and 90 degrees, but is usually less than 45 degrees.

224.2 Standard checks for SlopeDomain

Low value	0
High value	45
Unit	deg
List of values	

225 SnowDailyDomain

225.1 Description of SnowDailyDomain

The SnowDaily domain represents the daily solid accumulation on a snow board at an atmosphere study site. It is measured in millimeters (mm) and usually varies between 0 and 2000 mm per day, but with multiple day accumulations up to 6000 mm possible.

225.2 Standard checks for SnowDailyDomain

Low value	0
High value	2000
Unit	mm
List of values	

226 SnowTotalDomain

226.1 Description of SnowTotalDomain

The SnowTotal domain represents the total solid accumulation on a snow board at an atmosphere study site. It is measured in millimeters (mm) and usually varies between 0 and 6000 mm.

226.2 Standard checks for SnowTotalDomain

Low value	0
High value	6000
Unit	mm
List of values	

227 SoilColorDomain

227.1 Description of SoilColorDomain

The SoilColor domain is a set of codes that represent the color of soil in a soil sample. The color of soil is measured by Munsell colors in the format -- Hue:Value/Chroma. Hue is a number (possibly with one decimal point) followed by one or two characters; Value is a number (possibly with one decimal point); Chroma is a single digit.

227.2 Standard checks for SoilColorDomain

Low value	
High value	
Unit	Musell
List of values	

228 SoilConsistenceEstDomain

228.1 Description of SoilConsistenceEstDomain

The SoilConsistenceEst domain is a set of codes that represent a gross estimate of the consistence of a soil sample. The current values are: Loose, Friable, Firm, and Extremely firm. When a more accurate measurement of soil consistence is required, the BulkDensity, Porosity, and/or SoilTexture domains may be used.

228.2 Standard checks for SoilConsistenceEstDomain

Low value	0
High value	9
Unit	code

List of values	0□Unchecked 1□Loose 2□Friable 3□Firm 4□Extremely Firm

229 SoilDepositDomain

229.1 Description of SoilDepositDomain

The SoilDeposit domain is a set of codes that represent the parent material of soil samples from a soil characteristics site. It is specified in the GLOBE Teacher's Guide to be one of: Bedrock, Glacial Deposit, Volcanic Deposit, Stream Deposit, Wind Blown Deposit, Lake Deposit, Other, or Don't Know.

If Other is specified, then a short description of the type of parent material should be specified and maintained in the accompanying ParentMatOther column.

229.2 Standard checks for SoilDepositDomain

Low value	0
High value	10
Unit	code
List of values	0□Unchecked 1□Bedrock 2□Glacial Deposit 3□Volcanic Deposit 4□Stream Deposit 5□Wind Blown Deposit 6□Lake Deposit 7□Don't Know 8□Marine Deposit 9□Colluvium 10□Other

230 SoilDepthDomain

230.1 Description of SoilDepthDomain

The SoilDepth domain represents the depth of measurements taken in soil moisture and soil temperature experiments. It is measured in centimeters and takes on one of the following values: 5cm, 10cm, 30cm, 60cm, 90cm. The 5cm value is interpreted to mean "just below surface" and could vary from 0cm to 5cm.

230.2 Standard checks for SoilDepthDomain

Low value	
High value	
Unit	cm
List of values	5□ 10□ 30□ 60□ 90

231 SoilDryMthdDomain

231.1 Description of SoilDryMthdDomain

The DryMethod domain identifies the drying method used to dry out a soil sample. The choices are Oven (at 2 different temps),

Microwave, or Other.

231.2 Standard checks for SoilDryMthdDomain

Low value	0
High value	9
Unit	code
List of values	0 Unchecked 1 Oven (95-105 degC) 2 Microwave 3 Oven (75-94 degC) 4 Other

232 SoilDryTimeDomain

232.1 Description of SoilDryTimeDomain

The SoilDryTime domain represents the time (in hours with a two-place decimal fraction) that soil was dried in a soil moisture (Gravimetric method) measurement of a soil sample.

232.2 Standard checks for SoilDryTimeDomain

Low value	0.0
High value	120.0
Unit	hours
List of values	

233 SoilHorDepthDomain

233.1 Description of SoilHorDepthDomain

The SoilHorDepth domain represents the depth of either the top or bottom edge of a soil horizon at a soil characterization study site. It is also used to measure the top and bottom depths of a soil sample taken from the horizon. It is measured in centimeters (cm) with a possible decimal fraction.

233.2 Standard checks for SoilHorDepthDomain

Low value	0.0
High value	200.0
Unit	cm
List of values	

234 SoilMeterDomain

234.1 Description of SoilMeterDomain

The SoilMeter domain represents the reading of a gypsum block soil moisture meter at a soil moisture study site. Meter readings usually vary from 0 to about 200, but may vary from 0.0 to 999.9. The SoilMeter reading must be calibrated with other soil moisture measurements (using gravimetric methods) before a water content of the soil can be derived from the meter reading.

The range and units of known meters are:

Delmhorst (handheld) Range: 0-100 No Units
Irrometer/Watermark Range: 0-200 "centibars"
Spectrum/Watchdog Range: 0-200 "centibars"

Low value	0.0
High value	999.9
Unit	
List of values	

235 SoilSourceDomain

235.1 Description of SoilSourceDomain

The SoilSource domain is a set of codes that represent the type of collection method used to collect soil samples. It is specified in the GLOBE Teacher's Guide to be one of: Soil Pit, Auger Hole, Surface (<10 cm), Excavation, Road Cut, or Other.

If Other is specified, then a short description of the type of method should be specified and maintained in the accompanying SoilSourceOther column.

235.2 Standard checks for SoilSourceDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Soil Pit 2□Auger Hole 3□Surface (<10cm) 4□Excavation 5□Road Cut 6□Other

236 SoilStructEstDomain

236.1 Description of SoilStructEstDomain

The StructureEst domain is a set of codes that represent a gross estimate of the structure of a soil sample. The current values are: Granular, Blocky, Platy, Prismatic, Columnar, Single-grained, and Massive. When a more accurate measurement of soil structure is required, the BulkDensity, Porosity, and/or SoilTexture domains may be used.

236.2 Standard checks for SoilStructEstDomain

Low value	0
High value	9
Unit	code
List of values	1□Granular 2□Blocky 3□Platy 4□Prismatic 5□Columnar 6□Single-Grained 7□Massive 0□Unchecked

237 SoilTempDomain

237.1 Description of SoilTempDomain

The SoilTemp domain represents the soil temperature at a specific time. It is measured in degrees celsius (degC) and usually varies between -10.0 and +60.0 degC.

237.2 Standard checks for SoilTempDomain

Low value	-10.0
High value	60.0
Unit	deg C
List of values	

238 SoilTempUsageDomain

238.1 Description of SoilTempUsageDomain

The SoilTempUsage domain is a set of codes to identify the intended usage of soil temperatures measured according to the Soil Temperature Protocol in the Soil Investigation. The usages can be DailyAvg, Diurnal, both, or neither.

238.2 Standard checks for SoilTempUsageDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked/Neither 1□DailyAvg 2□Diurnal 3□Daily&Diurnal

239 SoilTextureDomain

239.1 Description of SoilTextureDomain

The SoilTextureEst domain is set of codes that represent one of 12 choices for estimating the texture of a soil sample. A 13th choice for "organic" soil was added in July 1999 to be used primarily by schools in Iceland.

An unknown value is represented by zero -- thus in general, any column pointing to this domain will be NOT NULL.

239.2 Standard checks for SoilTextureDomain

Low value	0
High value	12
Unit	code
List of values	0□Unchecked 1□sandy clay 2□sandy clay loam 3□sandy loam 4□silty clay 5□silty clay loam 6□silt loam 7□loamy sand 8□sand 9□silt 10□clay 11□clay loam 12□loam 13□organic

240 SoilThermomDomain

240.1 Description of SoilThermomDomain

The SoilThermom domain is a set of codes that represent the type of thermometer used to calculate the temperature of soil. Temperature readings are usually taken at depths near the surface (e.g. 5 cm and 10 cm). The accuracy of the measurement is dependent upon the type of thermometer used.

As of Spring 2002, thee is also a ThermomDomain which combines this SoilThermom domain with a new domain intended for air temperature thermometers or for dual-purpose thermometers. The codes for those thermometers include the codes for soil-only thermometers so that the same domain can be used for dual-purpose thermometers.

240.2 Standard checks for SoilThermomDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Digital 2□Dial 3□Other

241 SoilVolumeDomain

241.1 Description of SoilVolumeDomain

The SoilVolume domain represents the volume of a soil sample taken from a soil characterization study site. In the Pit Method it should vary between 100 and 500 mL (measured in whole mL), but in the Auger Method it may vary from 39.0 to 1963.0 mL (measured within 1 decimal place accuracy). It is also used to measure the volume of soil that settled out of a soil and water mixture at specific time intervals. It is measured in milli-liters (mL), or equivalently cubic centimeters (cm3), and usually varies between 0 and 400 mL. The total sample size should be greater than 20mL. The volume of rocks in the sample is measured by the separate RockVolumeDomain.

241.2 Standard checks for SoilVolumeDomain

Low value	20.0
High value	1963.0
Unit	mL=cm3
List of values	

242 SoilWeightDomain

242.1 Description of SoilWeightDomain

The SoilWeight domain represents the weight of a soil sample in a soil moisture sample, a soil infiltration sample, or a soil sample taken from a soil characterization study site. It is measured in grams and usually varies between 25 and 500 grams.

See the HydroWeight domain for sand, silt, and clay weights used in the particle-size distribution protocol by the hydrometer method.

242.2 Standard checks for SoilWeightDomain

Low value	10.0
High value	900.0
Unit	gm
List of values	

243 SolarDatetimeDomain

243.1 Description of SolarDatetimeDomain

The SolarDatetimeDomain identifies Local Solar Time at the location of a Globe measurement. It is derived from UT Time by a caluculation based on geographic location. It 's precision is up to 1 second. It may vary by up to an hour, or slightly more, from Local Clock Time.

SolarTime = UTtime + (Longitude/15) (Check this??)

243.2 Standard checks for SolarDatetimeDomain

Low value	
High value	
Unit	time
List of values	

244 SolarDayDomain

244.1 Description of SolarDayDomain

The SolarDay domain identifies the Solar Day that a measurement was taken. Visual Day and Solar Day are two different alternatives for estimating local solar time based on UTtime and Longitude.

SolarDay = TRUNC(UTtime + UTtoLocalConversion, DD)
VisualDay = TRUNC((SolarDay + 12hours) - UTtoLocalConversion), DD)
UTtoLocalConversion = UTtime + (Longitude/15) (Check this??)

Solar Day is used by NASA in various calculations. It is also used in the Primary Key of many tables to identify the local solar time that a measurement was taken.

244.2 Standard checks for SolarDayDomain

Low value	
High value	
Unit	time
List of values	

245 SolarHourDomain

245.1 Description of SolarHourDomain

The SolarHourDomain identifies the hour of Local Solar Time at the location of a Globe measurement. It is derived from UT Time in two steps: 1) Local Solar Time is derived from UT time by a caluculation based on geographic location, 2) Local Solar Time is truncated to Hour, and then the Day information is removed. The result is a number between 0 and 23 that identifies the Local Solar Hour. It may vary by up to an hour, or slightly more, from Local Clock Time.

245.2 Standard checks for SolarHourDomain

Low value	0
High value	23
Unit	hour
List of values	

246 SolarMinDomain

246.1 Description of SolarMinDomain

The SolarMinDomain identifies the minute of Local Solar Time at the location of a Globe measurement. It is derived from UT Time in two steps: 1) Local Solar Time is derived from UT time by a caluculation based on geographic location, 2) Local Solar Time is truncated to Minute, and then the Day and Hour information is removed. The result is a number between 0 and 59 that identifies the Local Solar Minute. It may vary by up to an hour, or slightly more, from Local Clock Time.

246.2 Standard checks for SolarMinDomain

Low value	0
High value	59
Unit	min
List of values	

247 SolarNoonDiffDomain

247.1 Description of SolarNoonDiffDomain

The SolarNoonDiff domain measures the difference, in seconds, between Solar Noon calculated by Solar Time and True Solar Noon. The calculation is:

```
SolarNoonDiff =   UTtime - SolarNoon          if   UTtime > SolarNoon
                  0                           if   UTtime < SolarNoon < UTtime + 59:59
                  SolarNoon - UTtime - 59:59  if  UTtime + 59:59 < SolarNoon
```

The difference will often be be within 15 minutes, but in extreme cases near the polar regions could be as much as 4-8 hours.

247.2 Standard checks for SolarNoonDiffDomain

Low value	0
High value	9999
Unit	seconds
List of values	

248 SourceStateDomain

248.1 Description of SourceStateDomain

The SourceState domain is a set of codes that represent the source state of a surface water site; it explains the reason that surface water measurements cannot be taken that day.

248.2 Standard checks for SourceStateDomain

Low value	0
High value	9
Unit	code
List of values	0 Unchecked 1 Frozen 2 Dried 3 Flooded 4 Not Reachable

249 SpeciesDomain

249.1 Description of SpeciesDomain

The Species domain is a set of codes that represent the species of trees at a biology or land cover study site. The Genus is recorded in a separate attribute. The 4-character code used at present is just the first 4 characters of the full latin species name; it uniquely identifies the species of interest.

249.2 Standard checks for SpeciesDomain

Low value	
High value	

Unit	
List of values	

250 SpecificGravityDomain

250.1 Description of SpecificGravityDomain

The SpecificGravity domain represents the specific gravity of surface water from a surface water study site (most meaningful for salt water sites) or a soil and water mixture prepared from a soil sample taken from a soil characterization study site. Specific Gravity has no units. The range of surface water specific gravity is usually from 1.0000 to 1.0700. The range of the soil and water mixture is usually from 0.9993 to 1.0300.

250.2 Standard checks for SpecificGravityDomain

Low value	0.999
High value	1.07
Unit	None
List of values	

251 SplitSessionDomain

251.1 Description of SplitSessionDomain

The SplitSession domain is a text field used to explain the days that a workshop session will meet if the days are not contiguous. The SplitSession value is always used in conjunction with BeginDate and EndDate values, which identify the first and last meeting days of a split session. For example, if a split session meets on every Saturday between the Begin and End dates, the value might be simply 'Saturdays'; if a split session meets non-regularly, then the actual meeting days may be listed.

If a SplitSession value is NULL, then it can be assumed that the workshop session is not a split session and that it meets every day from the BeginDate to the EndDate.

251.2 Standard checks for SplitSessionDomain

Low value	
High value	
Unit	
List of values	

252 StandingWaterDomain

252.1 Description of StandingWaterDomain

The StandingWater domain is a set of codes that represent various types of standing water. The current values for identifying standing water at a surface water study site are: Pond, Lake, Reservoir, or Other. It is possible that this domain will evolve over time to include other standing water types.

In March, 1999, we added new codes for Bay, Ditch, Ocean. Since the definition of "Other" now is less expansive than previously, the version date of each record in the SurfaceWater Site table was changed, and a comment was inserted to draw attention to the change.

252.2 Standard checks for StandingWaterDomain

Low value	0
High value	9
Unit	code

List of values	0□Unchecked 1□Pond 2□Lake 3□Reservoir 4□Other (<4/99) 5□Bay 6□Ditch 7□Ocean 8□Other (>4/99)

253 State(US)Domain

253.1 Description of State(US)Domain

The State(US) domain represents the codes used to identify a State in the United States of America. It is a 2 character code representing the 50 U.S. states and territories. This domain is used only in cases where only a US state is identified by an attribute; in cases where an international state, province, or other political region is to be identified, the StateProvDomain is used instead.

253.2 Standard checks for State(US)Domain

Low value	
High value	
Unit	
List of values	

254 StateProvDomain

254.1 Description of StateProvDomain

The StateProv domain is a set of codes that represent a political structure within a country. Such codes are identified and described in the StaeProvCodes table.

254.2 Standard checks for StateProvDomain

Low value	
High value	
Unit	
List of values	

255 StockNbrDomain

255.1 Description of StockNbrDomain

The StockNbr domain is a set of values that represent vendor's stock numbers for products of interest to GLOBE. It is an unstructured string in order to accommodate the styles and formats of different vendors.

255.2 Standard checks for StockNbrDomain

Low value	
High value	
Unit	
List of values	

256 StrategyText

256.1 Description of StrategyText

The StrategyText domain is the set set of character strings that can be used to describe the annual strategies listed by U.S. Partners in their roles as Recruiters, Trainers, and Mentors of GLOBE Teachers. Currently this text has no internal structure, but conceivably in the future it may satisfy some specific XML Document Type Definition.

256.2 Standard checks for StrategyText

Low value	
High value	
Unit	
List of values	

257 StudySiteDomain

257.1 Description of StudySiteDomain

TheStudySite domain consists of a set of codes that identify a study site or a sample site within a GLOBE school. A study site code is unique only within that school.

The format of the study-site code will be XXX-dd, where XXX is from the SiteCode domain and dd is a pair of digits to identify a local instance. If a school has more than 99 sites of a given type (e.g. LCL), then the format for newly created sites changes to XXXddd. The StudySite identifier should be treated as an object-identifier, because at some point in the future the GLOBE database may be modeled with object facilities. Thus the SiteCode should never be stipped off the StudySite identifier to identify the study site code; instead, the site type code should always be retrieved from the SiteCode column of the Study Site table.

A special SiteCode is used to identify the Main Entrace of a GLOBE School as described by the GPS Investigation protocol of the GLOBE Teachers Guide. The special site code for the main entrance is SCH and the measurement code for that measurement is FD for Front Door. All GPS readings captured by the School Location Data Entry Sheet will be stored in the Site Location table with SiteId = 'SCH-01', and the Data Activity table will record that action with siteId = 'SCH-01' and with MeasureCode = 'FD'.

257.2 Standard checks for StudySiteDomain

Low value	
High value	
Unit	
List of values	

258 SuffixDomain

258.1 Description of SuffixDomain

The Suffix domain is a character string data type of up to 10 characters to represent the suffix to a person's name. Normal suffixes are Esq., III, Sr., Jr., M.S., B.A., D.Ed., PhD, etc. At the present time the domain is not restricted to a fixed set of legal suffixes. The prefix to a person's name should use the PrefixDomain.

258.2 Standard checks for SuffixDomain

Low value	
High value	
Unit	
List of values	

259 SurfaceCoverDomain

259.1 Description of SurfaceCoverDomain

The SurfaceCover domain is a set of codes that represent the various surface cover conditions that may exist at a study site. Some of the surface cover conditions are permanent and are recorded as site characteristics in an AtmosphereSite or SoilMoistureSite table. Others vary from day to day and are recorded as daily metadata in a SoilMoisture or SoilMoistureWS table.

This domain was modified in October 2000 to add Sand, Roof, and Other. If "Roof" or "Other" is chosen, then a SurfCovDescribe comment field is provided to further describe the composition of the roof (e.g. asphalt, tar, cement, shake, etc.).

259.2 Standard checks for SurfaceCoverDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Paved 2□Bare Ground 3□Short Grass (<10cm) 4□Long Grass (>10cm) 5□Sand 6□Roof 7□Other

260 SurfaceTypeDomain

260.1 Description of SurfaceTypeDomain

The SurfaceType domain is a set of codes that represent the various types of surface disturbances at a soil moisture study site. The choices for surface type are: Natural, Plowed, Graded, Backfill, Compacted, Other.

260.2 Standard checks for SurfaceTypeDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Natural 2□Plowed 3□Graded 4□Backfill 5□Compacted 6□Other

261 TableNameDomain

261.1 Description of TableNameDomain

The TableName domain represents the name of a table in the GLOBE Database.

261.2 Standard checks for TableNameDomain

Low value	
High value	
Unit	
List of values	

262 TableTypeDomain

262.1 Description of TableTypeDomain

The TableType domain is a set of codes that represent a type of study site table in the GLOBE database. These codes are used to distinguish among at least the following types of tables related to study sites: study site characteristics, student data tables, student data worksheets.

262.2 Standard checks for TableTypeDomain

Low value	
High value	
Unit	code
List of values	SC Site Characteristics SD Summary Data WS WorkSheet

263 TaskIdDomain

263.1 Description of TaskIdDomain

The TaskId domain is a set of identifiers for GLOBE learning tasks.

263.2 Standard checks for TaskIdDomain

Low value	
High value	
Unit	
List of values	

264 TeacherRoleDomain

264.1 Description of TeacherRoleDomain

The TeacherRole domain is a set of values that represent the various roles that a teacher may play in association with a GLOBE school. It most cases the teacher will be a classroom teacher, but in other cases the teacher may be a resource teacher, an administrative aide, or be associated with the school in some other capacity. This domain is not defined to have a pre-determined set of values; instead, an arbitrary text description can be inserted.

Two special codes will be used to identify whether TeacherRole is as an International Contact (I-CONTACT) or as an International Country Coordinator (I-COORD)

264.2 Standard checks for TeacherRoleDomain

Low value	
High value	
Unit	
List of values	

265 TeacherStatusDomain

265.1 Standard checks for TeacherStatusDomain

Low value	
High value	
Unit	
List of values	

266 TelPrefixUSDomain

266.1 Standard checks for TelPrefixUSDomain

Low value	
High value	
Unit	
List of values	

267 TelephoneDomain

267.1 Description of TelephoneDomain

The Telephone domain is a set of character strings that represent full international telephone numbers. An example is +61 6 216 5933, where 61 is the country code for Australia, 6 is the city code for Canberra, 216 is a local exchange, etc. The "+" indicates that international access codes, independent of the number being called, may need to be dialed first. If an extension is needed, it can follow the number, e.g. x325. Following this format, North American numbers with extensions would be in the form: +1-202-395-7600 x325.

267.2 Standard checks for TelephoneDomain

Low value	
High value	
Unit	
List of values	

268 ThermomTypeDomain

268.1 Description of ThermomTypeDomain

The ThermomType domain is a numeric code to identify the type of thermometer used in the GLOBE Atmosphere investigation for air temperature or in the GLOBE Soil investigation for soil temperature.

The codes of this domain include the previously defined codes of the SoilThermomType domain so that they can be referenced in both situations in metadata tables.

The codes as of Spring 2002 are as follows:

0 Unknown
1 Digital - Soil Only
2 Dial - Soil Only
3 Other - Soil or Air
4 MaxMin U-tube Mercury - Air Only
5 MaxMin Alcohol based - Air Only
6 Max/Min Digital (Air&Soil probes: 0.1 precision)

268.2 Standard checks for ThermomTypeDomain

Low value	0
High value	9
Unit	encoding
List of values	0□Unknown 1□Digital - Soil Only 2□Dial - Soil Only 3□Other - Soil or Air 4□MaxMin U-tube Mercury - Air Only 5□MaxMin alcohol based - Air Only 6□Max/Min Digital (Air&Soil probes: 0.1 precision)

269 TideTypeDomain

269.1 Description of TideTypeDomain

The TideType domain is a set of codes to represent either a High Tide or a Low Tide.

269.2 Standard checks for TideTypeDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□High Tide 2□Low Tide

270 TimeZoneUSDomain

270.1 Description of TimeZoneUSDomain

The TimezoneUS domain is a collection of codes to represent time zones used by US states and territories. It includes EST, CST, MST, PST, as well as some numeric offsets to represent islands off the East or West coasts, e.g. PST-1, EST+1.

270.2 Standard checks for TimeZoneUSDomain

Low value	
High value	
Unit	
List of values	

271 TimezoneDomain

271.1 Description of TimezoneDomain

The TimeZone domain identifies the local time zone of a GLOBE school. International schools are asked in which time zone they are located, whereas U.S. schools have not been asked this question in the past. Sometimes the schools answer the question by filling in a local name for the time zone (up to 10 characters) and sometimes they give an offset of + or - hours from Greenwich time. At the present time the values for this attribute are not standardized; whatever is given by the school is recorded into the database.

The SQL-92 standard defines a time zone interval as follows:

```
<time zone interval> ::=
        <sign> <hours value> <colon> <minutes value>
```

so the format is "+hh:mm" or "-hh:mm", a 6 character field. This accommodates the several time zones in Asia that vary by a specific number of minutes from the rest of the world.

271.2 Standard checks for TimezoneDomain

Low value	
High value	
Unit	
List of values	

272 TraceMissDomain

272.1 Description of TraceMissDomain

The TraceMiss domain is a set of codes that represent a precipitation measurement that is reported to be a Trace (T) amount, Missing/Melted (M), or Accumulated (A) over several days. In the database, a Missing/Melted (M) measurement is recorded as a NULL value and a Trace (T) amount is recorded as the smallest positive number that can be represented in the identified field.

A TMAflag value of Derived (D) indicates that the value was derived from another reported value. For example, in the PRECIP_SOLID table a zero value for NewAmount or CurrentDepth may result in a derived value of zero for RainEquivalent or WaterEquiv, respectively. The derived value is calculated as such only if an explicit value was not reported by the school.

A TMAflag value of Estimated (E) is used in some view tables to indicate that a water equivalent value was estimated from a snowdepth value, usually by dividing by 7. Values so marked have no scientific validity! They occur only in the views that lead to the snapshot tables PHENOLOGY_WORKUP and GDS_MOISTCHG_30DAYS.

The Trace or Missing Flag that is associated with a numeric measurement allows disambiguatuion among values equivalent to the numeric equivalent of T or M that were determined by other means.

Use of this domain for Air and Soil Temperature

The Missing (M) value of this domain may also be used for air or soil temperature measurements. Beginning in Fall 2001, the meaning of "X" or "M" values in an air temperature or soil temperature data entry field will result in the following (temp, flag) entries in the database:

1) A numeric value in an email data entry for a temperature field will result in that value being stored in the temperature field in the database, and a null value being stored in its associated flag. The implied semantics is that the school thinks the value supplied is a good value for that temperature.

2) An "X" in an email data entry for a temperature field will result in both the temperature value and its associated flag being null in the database. The implied semantics is that no attempt was made to measure that temperature.

3) An "M" in an email data entry for a temperature field will result in the temperature value being null and its associated flag having the value "M" in the database. The implied semantics is that an attempt to record the temperature was made -- but for some reason (maybe specified in the Comments) the attempt was not successful and the desired temperature value is "Missing".

We will not allow a non-null value for temperature to be paired with a non-null value for its associated flag. The database design allows for this constraint to be waived in the future if GLOBE decides that would be useful! Such a change would require re-designing the email data entry.

272.2 Standard checks for TraceMissDomain

Low value	
High value	
Unit	code
List of values	T□Trace M□Missing/Melted A□Accumulated D□Derived E□Estimated

273 TrackingIdDomain

273.1 Description of TrackingIdDomain

The TrackingId domain identifies each piece of mail sent from GLOBE (via RCG) to teachers, schools, or possibly in the future partners.

Sometimes a piece of mail is sent multiple times to the same recipient because the original piece was returned with an inadequate address, etc. Thus in the MAILING_STATUS table, the TrackingId is paired with a SendDate to identify each effort to send the same piece of mail to a recipient.

At other times, when a piece of mail is lost, a new replacement mailing will be created with a new TrackingId. RCG will develop a policy over time for when it is best to retain an original TrackingId and when it is better to create a new TrackingId.

Currently, a TrackingId is up to a 5 digit identifier. It is definied to be a CHAR(7) datatype to allow for a 2 character lead-in and a 5 digit number code. The 2-char lead may identify whose sequencer was used to generate the 5 digit identifier.

273.2 Standard checks for TrackingIdDomain

Low value	
High value	
Unit	
List of values	

274 TrainerLevelDomain

274.1 Description of TrainerLevelDomain

The TrainerLevel domain keeps track of the "level" attained by a trainer. Initially GLOBE supports the following levels:

Trainer -- the person has attended a train-the-trainer (TTT) workshop and completed the course of instruction in that protocol area, or GLOBE HQ makes a decision that a person is qualified to be a trainer. Requires no special blessing from GLOBE HQ. Teachers who successfully complete a TTT workshop in a given protocol group will automatically be added to GLOBE TRAINER and linked at the Trainer level to that protocol group. GLOBE Trainer's may be the lead trainer at training workshops for Teachers.

AssistTTT -- Cleared to Assist, i.e. the person satsifies the requirements of the "Trainer" level and in addition is cleared by GLOBE HQ as qualified to assist at train-the-trainer (TTT) workshops.

LeadTTT -- Cleared to Lead, i.e. the person satisfies the requirements of the "AssistTTT" level and in addition has acted as an Assistant Trainer at a GLOBE HQ led TTT workshop at least once, and has been deemed qualified by GLOBE HQ to to be a Lead Trainer at a GLOBE HQ led TTT workshop for a given protocol group.

Master -- Master Trainer. This means that the person satisfies the requirements of the "LeadTTT" level and in addition has acted as a Lead Trainer at a GLOBE HQ led TTT workshop at least once, and has been deemed qualified by GLOBE HQ to be a Master Trainer for a given protocol group. A Master Trainer is qualified to train Trainers at any TTT workshop (GLOBE led or not). A Master Trainer who has not trained in 3 years reverts to "LeadTTT" status.

A second attribute of TrainerLevel is BestowedBy. This is a text string that gives the name of the person from GLOBE HQ who changes a TrainerLevel from Trained to some higher level. We could populate this field automatically with the database UserName of the person who makes such an update!

274.2 Standard checks for TrainerLevelDomain

Low value	
High value	
Unit	
List of values	Trainer□GLOBE Trainer AssistTTT□Cleared to Assist at TTT LeadTTT□Cleared to lead at TTT Master□Master Trainer

275 TrainerRoleDomain

275.1 Description of TrainerRoleDomain

The TrainerRoleCode domain is a set of codes that represent the various roles to be filled by GLOBE trainers at a GLOBE workshop session. The possible trainer roles are as follows:

(1) GLOBE Training Team Facilitator (TTFACIL)
Instructional Role: Team leader and workshop facilitator.
Administrative Role: Communication liaison with the Contract Site Coordinator.

(2) GLOBE Scientist (SCIENCE)
Instructional Role: Provide instruction in the science, the protocols and, possibly, some of the learning activities.
Administrative Role: Assist in pre- and post-workshop set-up, clean-up.

(3) GLOBE Technology Educator (TECH-ED)
Instructional Role: Provide instruction in the use of computers, GLOBE Web space and the World Wide Web.
Administrative Role: Verifies that computers and software are ready for use.

(4) GLOBE Teacher Educator
Instructional Role: Provide instruction in the learning activities and other educational issues and, possibly, some of the protocols.
Administrative Role: Assist in pre- and post-workshop set-up, clean-up.

Subroles under GLOBE Scientist include:
(Deprecated when STAFF PROTOCOLS gets implemented)

Scientist - Atmosphere (SCI-ATM)
Scientist - Soils (SCI-SOI)
Scientist - Land Cover and Biology (SCI-LCB)
Scientist - Hydrology (SCI-HYD)

Note: If SCIENCE is specified, then the table STAFF PROTOCOLS table will list the investigation area or areas that the trainer trained. (Implementation planned Spring 2002).

Subroles under Teacher Educator include:

Teacher Educator - Learning Activities (TEA-LRN)
Teacher Educator - GLOBE Implementation (TEA-IMP)

275.2 Standard checks for TrainerRoleDomain

Low value	
High value	
Unit	code
List of values	TTFACIL□Training Team Facilitator SCI-ATM□Scientist - Atmosphere SCI-HYD□Scientist - Hydrology SCI-LCB□Scientist - Land Cover & Biology SCI-SOI□Scientist - Soils TEA-LRN□Teacher Educator - Learning Activities TEA-IMP□Teacher Educator - GLOBE Implementation TECH-ED□Technology Educator SCIENCE□Science Training of specific protocols (cf STAFF_PROTOCOLS) ASSISNT□Science Training Assistant of specific protocols (cf STAFF_PROTOCOLS)

276 TrainerTypeDomain

276.1 Description of TrainerTypeDomain

The TrainerType domain is a set of codes to identify the primary type of training that the trainer does. Initially the type is one of the following:

Contract Trainer (CNTR)
GRIT Corps Trainer (GRIT)
Franchise Trainer (FRAN)
International Trainer (INTL)

276.2 Standard checks for TrainerTypeDomain

Low value	
High value	
Unit	code
List of values	CONT□Contract Trainer GRIT□GRIT Trainer FRAN□Franchise Trainer INTL□International Trainer

277 TrainingOptionDomain

277.1 Description of TrainingOptionDomain

The TrainingOption domain is a set of codes that identify the training option chosen or assigned for a workshop training session. The codes are as follows:

5A Registration for a 5-day workshop with the intent to take the Advanced option on the 4-th and 5-th days. If instead, they actually take the Basic option, this code can be changed to 5B at workshop completion.

5B Registration for a 5-day workshop with the intent to take the Basic option on the 4-th and 5-th days. If instead, they actually take the Advanced option, this code can be changed to 5A at workshop completion.

5? Registration for a 5-day workshop with no indication as to the intent for the 4-th and 5-th days. Must be changed to 5A or 5B, or possibly 4A or 4B, or maybe even 3B at workshop completion, depending on what portion of the workshop was attended.

2A Registration for a 2-day workshop (the 4-thand 5-th days of a 5-day workshop) with the intent to take the Advanced protocol option.

2B Registration for a 2-day workshop (the 4-thand 5-th days of a 5-day workshop) with the intent to take the Basic option as a review or refresher of Basic protocols.

2? Registration for a 2-day workshop with no indication as to the option intended. Must be changed to 2A or 2B (or 1A or 1B if they only complete 1 day) at workshop completion.

3B A special non-advertised code used to identify those people who complete the first 3 days of the workshop, and become registered Globe teachers, but fail for whatever reason to attend the 4-th day of a 4-day workshop or the last 2 days of a 5-day workshop.

In October 2000, the following codes were added:

7C Seven day Basic and Advanced (used for TTT workshops)

7A Seven day Advanced (usage?)

7B Seven Day Basic (usage?)

The following are legacy codes:

4A Registration for a 4-day workshop with the intent to take the Advanced option on the 4-th day. If instead, they actually take the Basic option, this code can be changed to 4B at workshop completion.

4B Registration for a 4-day workshop with the intent to take the Basic option on the 4-th day. If instead, they actually take the Advanced option, this code can be changed to 4A at workshop completion.

4? Registration for a 4-day workshop with no indication as to the intent for the 4-th day. Must be changed to 4A, 4B, or 3B at workshop completion.

1A Registration for a 1-day workshop (the 4-th day only) with the intent to take the Advanced protocol option.

1B Registration for a 1-day workshop (the 4-th day only) with the intent to take the Basic option as a review or refresher of Basic protocols.

1? Registration for a 1-day workshop with no indication as to the option intended. Must be changed to 1A or 1B at workshop completion.

4L A special code used to identify the "legacy" training option for all GLOBE teacher training workshops held prior to Summer 1998. The 4L option is not valid for workshops beginning in Summer 1998 and beyond.

LG A special code used to indicate that the workshop is a legacy workshop; i.e., a pseudo workshop to collect together all teachers certified as trained by a country, but where explicit workshop information is unknown.

No changes will be necessary to the existing RegisterStatusCode column in the Registration table. The existing codes (P1, P2, P3, TR, RF, RC, NS, AW, CW) can be paired with the new TrainingOption codes described above to indicate whether the training option is "intended" or "completed".

277.2 Standard checks for TrainingOptionDomain

Low value	
High value	
Unit	
List of values	5A□5-day w Advanced option 5B□5-day w Basic option 5?□5-day w Unknown option 2A□2-day w Advanced option 2B□2-day w Basic option 2?□2-day w Unknown option 3B□3-day w Basic option 4A□4-day w Advanced option 4B□4-day w Basic option 4?□4-day w Unknown option 1A□1-day w Advanced option 1B□1-day w Basic option 1?□1-day w Unknown option 4L□4-day Legacy workshop 7C□7-day w Adv and Basic (TTT workshops) LG□Pseudo, catch-all workshop

278 TransectIdDomain

278.1 Description of TransectIdDomain

The TransectId domain is an integer to identify the 3 to 7 Transects that are used to describe the fire fuel characteristics of a Fire Fuel Ecology study site. The first transect is due East from the study site center. The others are laid out counter-clockwise on a hexagon circumventing the site center.

278.2 Standard checks for TransectIdDomain

Low value	
High value	
Unit	
List of values	

279 TreeCircumDomain

279.1 Description of TreeCircumDomain

The TreeCircum domain represents the circumference of a tree, or the average circumference of a collection of trees, in a biology or land cover study site. It is measured in centimeters (cm) and usually varies between 1.0 and 3000.0 cm.

279.2 Standard checks for TreeCircumDomain

Low value	qor
High value	3000.0
Unit	cm
List of values	

280 TreeDiamDomain

280.1 Description of TreeDiamDomain

The TreeDiam domain represents the diameter of a tree, or the average diameter of a collection of trees, in a biology or land cover study site. It is a derived value, calculated from student circumference measurements, measured in centimeters (cm) and usually varies between 1.0 and 1000.0 cm.

280.2 Standard checks for TreeDiamDomain

Low value	1.0
High value	1000.0
Unit	cm
List of values	

281 TreeGrassShrubDomain

281.1 Standard checks for TreeGrassShrubDomain

Low value	
High value	
Unit	identifier
List of values	T□Tree G□Grass S□Shrub

282 TreeHeightDomain

282.1 Description of TreeHeightDomain

The TreeHeight domain represents the height of a tree, or the average height of a collection of trees, in a biology or land cover study site. It is measured in meters (m) (with a decimal fraction) and usually varies between 0.1 and 99.0 meters.

282.2 Standard checks for TreeHeightDomain

Low value	0.1
High value	99.0
Unit	m
List of values	

283 TreeIdDomain

283.1 Standard checks for TreeIdDomain

Low value	

High value	
Unit	
List of values	

284 TurbidityEstDomain

284.1 Description of TurbidityEstDomain

The TurbidityEst domain is a set of codes that give a gross estimate of the turbidity of the water source from which surface water samples will be taken.

284.2 Standard checks for TurbidityEstDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Clear 2□Turbid 3□Unknown

285 TurbiditySecchiDomain

285.1 Description of TurbiditySecchiDomain

The TurbiditySecchi domain represents the depth at which a Secchi disk either appears or disappears when directly submerged into the source water of a surface water study site. It is measured in meters (with a decimal fraction) and usually varies between 0.00 and 40.00 meters.

Secchi depths are often measured from some height above the water (varying from 0 to 40 meters) and the combined distances are stored in the database along with the actual disappearing depth from the surface. At the present time this single domain is used for all such measurements, thus measurements may vary from 0 up to 80 meters.

285.2 Standard checks for TurbiditySecchiDomain

Low value	0.00
High value	80.00
Unit	m
List of values	

286 TurbidityTubeDomain

286.1 Description of TurbidityTubeDomain

The TurbidityTube domain represents the depth of water in a small test tube at which a pattern at the base of the tube disappears when water sampled from a surface water study site is poured into the tube. It is measured in centimeters (without a decimal fraction!) and usually varies between 0 and 150 cm.

In October 2001, in response to a request from Dixon, we changed the upper limit to 250 cm to accommodate a school with a very long tube.

286.2 Standard checks for TurbidityTubeDomain

Low value	0
High value	250

Unit	cm
List of values	

287 URLdomain

287.1 Description of URLdomain

The URL domain consists of character strings that represent the Universal Resource Locator (URL) of a document or other WWW object. The format of a URL is defined by the Internet Society.

287.2 Standard checks for URLdomain

Low value	
High value	
Unit	
List of values	

288 UToffsetDomain

288.1 Description of UToffsetDomain

The UToffset domain identifies the difference between UT time and Local time at a GLOBE study site. Local time is completely under the control of local authorities and cannot be easily maintained by the GLOBE database. Whenever GLOBE measurements are reported in local time, GLOBE requires that the offset from UT time also be reported.

This offset will be recorded in the database as a signed literal string of length exactly 6 characters and having the format: shh:mm, where s is a sign (+/-), and hh:mm is the hours and minutes difference from UT time. A leading blank space is interpreted as a "+" sign.

We require the colon (:) to avoid any mis-interpretation of this literal as a 4 digit number

288.2 Standard checks for UToffsetDomain

Low value	
High value	
Unit	
List of values	

289 UrbanCodeDomain

289.1 Description of UrbanCodeDomain

The UrbanCode domain is a set of codes that represent the urban environment in an area immediately surrounding a GLOBE School. The codes are defined and described in the UrbanizationCodes table.

289.2 Standard checks for UrbanCodeDomain

Low value	
High value	
Unit	code

List of values	DU□Dense Urban OU□Open Urban SB□Suburban MC□Moderate City SC□Small City TT□Town VV□Village RR□Rural

290 VendorDomain

290.1 Description of VendorDomain

The Vendor domain is a set of codes that identify vendors with whom GLOBE does business. Vendor codes are defined and described in the GlobeVendor table.

290.2 Standard checks for VendorDomain

Low value	
High value	
Unit	
List of values	

291 VisualDayDomain

291.1 Description of VisualDayDomain

The VisualDay domain identifies the Visual Day that a measurement was taken. Visual Day and Solar Day are two different alternatives for estimating local solar time based on UTtime and Longitude.

SolarDay = TRUNC(UTtime + UTtoLocalConversion, DD)
VisualDay = TRUNC((SolarDay + 12hours) - UTtoLocalConversion), DD)
UTtoLocalConversion = UTtime + (Longitude/15) (Check this??)

Visual Day is used by NASA to determine accurate visualizations of measurements for a given day regardless of the location where the measurement was observed.

291.2 Standard checks for VisualDayDomain

Low value	
High value	
Unit	day
List of values	

292 WaterAreaDomain

292.1 Description of WaterAreaDomain

The WaterArea domain represents an approximation of the surface area of the body of water identified by a surface water study site. It is measured in square kilometers (with a 3-place decimal fraction) and varies between 0 and 99999.999 square kilometers.

292.2 Standard checks for WaterAreaDomain

Low value	0.001
High value	99999.000
Unit	km2

List of values	

293 WaterContentDomain

293.1 Description of WaterContentDomain

The WaterContent domain represents measure of the water content of a soil sample. It is a ratio of the weight of the water in the soil divided by the weight of the dry soil from which the water has been removed. The ratio is dimensionless, i.e. it is not a percentage, and usually varies between 0.00 and 0.65.

In extreme cases (e.g. Iceland and Antartica) this ratio can be greater than one, thus the range limits on this ration are from 0.0 to 2.00.

293.2 Annotation for WaterContentDomain

Prior to Summer 1999, this domain was defined as a pure ratio * 100 (i.e. g/g*100) to make it look like a percentage. In August 1999 this domain was redefined to be a pure ratio (i.e. previous values all divided by 100) with the new range of values allowed as 0.00 to 2.00.

In Sept 1999 reverted back to Ratio*100 so that visualizations would work again.

Later in Sept 1999 , all systems groups agreed on a schedule that would convert all data entry pages, the database, all visualizations, and teachers guide references to "pure ratios", hopefully by the end of 1999. In the interim, there may be some mixtures of input data as some schools are entering pure ratios and some are multiplying by 100 and entering that number.

293.3 Standard checks for WaterContentDomain

Low value	0.00
High value	2.00
Unit	g/g
List of values	

294 WaterDepthDomain

294.1 Description of WaterDepthDomain

The WaterDepth domain represents an approximation of the average depth of the body of water identified by a surface water study site. It is measured in meters (with a decimal fraction) and usually varies between 0.1 and 740.0 meters.

294.2 Standard checks for WaterDepthDomain

Low value	0.1
High value	999.9
Unit	m
List of values	

295 WaterDiffDomain

295.1 Description of WaterDiffDomain

The WaterDiff domain is the subtraction of the PET total for a given 30-day period from the Precipitation total for that same 30 day period. It is measured in millimeters (mm). If WD is positive, then the environment is said to be having "wet conditions". If WD is positive, then the environment is said to be having "dry conditions".

295.2 Standard checks for WaterDiffDomain

Low value	-45000
High value	45420

Unit	mm
List of values	

296 WaterLocationDomain

296.1 Description of WaterLocationDomain

The WaterLocation domain is a set of codes that represent the various possible locations where water samples are taken from a surface water study site. Currently used values are: Outlet, Bank, Bridge, Boat, Inlet, Pier.

296.2 Standard checks for WaterLocationDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□Outlet 2□Bank 3□Bridge 4□Boat 5□Inlet 6□Pier

297 WaterSizeDomain

297.1 Description of WaterSizeDomain

The Watersize domain is a set of codes to represent approximate sizes of a standing body of water. The current choices are less than, equal to, or greater than 50m x 100m.

297.2 Standard checks for WaterSizeDomain

Low value	0
High value	9
Unit	code
List of values	0□Unchecked 1□< 50m x 100m 2□= 50m x 100m 3□> 50m x 100m

298 WaterTempDomain

298.1 Description of WaterTempDomain

The WaterTemp domain represents surface water temperature; it is also used to measure the temperature of a soil-water mixture in the laboratory.. It is measured in degrees celsius (degC) and usually varies between 0.0 and +40.0 degC.

298.2 Standard checks for WaterTempDomain

Low value	-2.0
High value	40.0
Unit	deg C
List of values	

299 WaterWidthDomain

299.1 Description of WaterWidthDomain

The WaterWidth domain represents an approximation of the width of the body of water identified by a surface water study site. It is measured in meters (with a decimal fraction) and varies between 0.0 and 9999.9 meters.

299.2 Standard checks for WaterWidthDomain

Low value	0.0
High value	9999.9
Unit	m
List of values	

300 WavelengthDomain

300.1 Description of WavelengthDomain

The Wavelength domain holds an integer value for the wavelength of light. It is used for photometer calibration in the PHOTOMETER CALIB table, and initially is set to 625 for Red and 505 for Green.

300.2 Standard checks for WavelengthDomain

Low value	
High value	
Unit	
List of values	

301 WindSpeedDomain

301.1 Description of WindSpeedDomain

[NEED TEXT] Is windspeed measured in MPH or Km/Hr? What is the range of values?

301.2 Standard checks for WindSpeedDomain

Low value	0
High value	????
Unit	?????
List of values	

302 WkshpCategoryDomain

302.1 Description of WkshpCategoryDomain

The WkshpCategory domain is a set of codes to identify the various categories of workshops held by GLOBE or its Franchises. Workshops are already categorized by WorkshopName (cf. Workshop Domain) and this categorization is a separate categorization bytype of workshop.

The categories as of October 1, 1998, are:

- Complete or Basic Workshop : Workshop covers all basic protocol areas, IDs are issued, Certificates, Evals, LandSats for participants, collect Roster Information.

- Follow-up : This could be advanced protocols, refresher, etc. Participant data would not be collected, no evaluation form, no IDs, but maybe a landsat for the actual location and some materials. Data would include Preworkshop general, materials, contact (shipping), max enrollment, actual attendance number, and short summary.

- Pre Service : A workshop which is a course designed for college students in a pre service setting. The "Pre service data form" would need to be filled out for each participant. They get certificates, but no IDs or school level Images. If selected category is "pre service"

then IDs wouldn't need to be issued. They would complete the general preworkshop, materials, contacts, credits, summary (description), and evaluation.

- Conference: Again, Participant data would not be collected, only the number expected and actually attended, General Information, no IDs, Images, etc. If materials are needed, then fill in the number and provide the shipping contact.

302.2 Standard checks for WkshpCategoryDomain

Low value	
High value	
Unit	
List of values	COMPLT□Complete or Basic FOLL-UP□Follow-Up PRE-SER□Pre-Service CONF□Conference

303 WorkshopDomain

303.1 Description of WorkshopDomain

The Workshop domain consists of names that identify general GLOBE workshops (not a session of that workshop). GLOBE workshops are defined and described in the Workshop table.

303.2 Standard checks for WorkshopDomain

Low value	
High value	
Unit	code
List of values	FIT□Field Initiated Training FRAN-TR□Franchise - Trainer FRAN-TE□Franchise - Teacher CONT□Contract INT-R□International - Regional INT-C□International - In Country GTTT□Globe Train-the-Trainer HOST□Hosted CONF□Conference SPEC□Special

304 YearDomain

304.1 Description of YearDomain

The Year domain represents a date that is truncated to its year value only. Its literal representation is as 4 digits to identify the calendar year.

304.2 Standard checks for YearDomain

Low value	
High value	
Unit	
List of values	

305 ZipCodeUS5Domain

305.1 Description of ZipCodeUS5Domain

The ZipCodeUS5 domain is the collection of all 5-digit U.S. zip codes. Nine digit US extended zip codes will use the international

PostalCode domain.

305.2 Standard checks for ZipCodeUS5Domain

Low value	
High value	
Unit	
List of values	

306 ZoomLevelDomain

306.1 Description of ZoomLevelDomain

The ZoomLevel domain represents hierarchical paths that help control navigation through the GLOBE Student Interaction (GSI) maps. It is a tree structure with nodes and
leafs. We start at the top, with the "World" zoom level and then depending on where the client wishes to go, we begin to branch off of this. So, there is a continent type level, for example "World Africa" represents the level of Africa (map), then country, "World Europe UK" for the United Kingdom (map); then if possible, a state or region level: "World NA US MD" for Maryland (map). The lowest level, the level at which there are schools is a "leaf" level which is just the lowest level in the tree with a " L" tacked on to the end of the zoom level. Each zoom level is represented by a character string that identifies each node in the path. For example, all schools in Maryland have a
"World NA US MD L" zoom level.

306.2 Standard checks for ZoomLevelDomain

Low value	
High value	
Unit	
List of values	Basic Advanced Optional Special

307 Table ACIDITY_WS

307.1 Description of table ACIDITY_WS

The AcidityWS table records soil acidity pH readings for multiple samples taken from a given Horizon at a given soil characteristics study site. The sample measurements are averaged and the average reported in the AcidityAvg attribute of the SoilData table.

308 Table ACIDITY_WS_CNT

309 Table ADMIN_ACTIVITY

310 Table AIR_TEMP

310.1 Description of table AIR_TEMP

The AirTemperature entity records daily air temperature measurements at a specific Atmosphere study site.

The GLOBE Protocol for measuring air temperatures is given on pages 2-26 through 2-28 of the GLOBE II Teacher's Guide. The student data entry sheet is given on page 2-62.

```
AirTemperature entity
  SchoolId            (PK, FK1)
  SiteId              (PK, FK1)
  DateMeasured        (PK)
  CurrentTemperature
  MaximumTemperature
  MinimumTemperature
```

The 3-part primary key associates measurements performed on a specific date with a specific GLOBE school study site.

The foreign key FK1 references the StudySite entity, and enforces a one-to-many mapping from a study site to a collection of air temperature measurements taken at that site on different days

The DateMeasured attribute identifies the Date on which air temperature measurements were made.

The CurrentTemperature attribute records the current temperature (measured in degrees Celsius); it is read within one hour of solar noon.

The MaximumTemperature and MinimumTemperature attributes record the maximum and minimum temperatures, respectively, since the last time the thermometer indicators were reset.

This entity design assumes that CurrentTemperature is recorded within one hour of local solar noon (see GLOBE II Teacher's Guide, Atmosphere Protocol 5, page 2-26), so it does not record an HourMeasured attribute.

310.2 Annotation for table AIR_TEMP

The foreign key FK1 references the StudySite entity, and enforces a one-to-many mapping from a study site to a collection of air temperature measurements taken at that site on different days. An additional constraint is needed to ensure that the referenced site is in fact a site created for Air Temperature measurements, i.e. a site whose MeasurementCode attribute is either AM or AT. There are various ways to enforce this additional constraint in the Physical Database Design.

This entity corresponds directly to the existing FSL table GLOBE2AT, but with the DateReported column removed (that attribute is maintained instead in the DataActivity entity).

311 Table AIR_TEMP_CNT

312 Table ARCHIVEMESSAGELOG

312.1 Description of table ARCHIVEMESSAGELOG

The ARCHIVEMESSAGELOG table maintains a copy of every GLOBEMail message ever sent.

313 Table ATMOSPHERE_SITE

313.1 Description of table ATMOSPHERE_SITE

The AtmosphereSite entity records Atmosphere study site characteristics that are somewhat stable throughout the lifetime of that site. If any of these characteristics should change significantly, then the site itself should be deactivated and no further measurements made because the data would no longer be interpreted under the same site characteristics. A new site could then be created with new site characteristics identified.

AtmosphereSite entity
SchoolId (PK, FK)
SiteId (PK, FK)
VersionDate (PK)
ObstacleDistance
ObstacleHeight
SurfaceCover
Comments

The primary key (PK) identifies metadata associated with a specific GLOBE study site. Site characteristics may vary over time and VersionDate identifies the characteristics in effect during a specific time period. Past versions are never deleted from the database.

The foreign key (FK) pair (SchoolId, SiteId) ensures that these characteristics map to an existing GLOBE study site, but does not require that every Atmosphere study site have a set of site characteristics associated with it. In some cases an Atmosphere study site may be created and recorded in the StudySite entity first, with the site characteristics measured and recorded in this entity later, possibly never. If it is a GLOBE requirement that every Atmosphere study site have the attributes of this entity always recorded in the database, then an additional constraint to enforce this rule is needed.

The ObstacleDistance attribute records the distance (in meters) to the nearest building or tree. The ObstacleHeight attribute records the obstacle height (in meters). The SurfaceCover attribute identifies the surface cover type of the study site. It is specified to be one of: Paved, Bare Groud, Short Grass (< 10cm), or Long Grass (> 10 cm). The Comments attribute allows further textual explantion of site characteristics that may influence student measurements. This attribute is constrained to be a character string of length no more than 200 characters.

There is currently a discussion underway to consider how to enhance the description of obstructions relevant to an Atmosphere study site and to allow identification of the location and other characteristics of multiple obstructions (See Len Gallagher email 7/18/00).

313.2 Annotation for table ATMOSPHERE_SITE

The ObstacleDistance attribute records the distance (in meters) to the nearest building or tree. The GLOBE II data entry sheet that specifies this characteristic (Teacher's Guide pg 2-65) does not record whether the obstacle is a tree or building. That information, or an indicator for additional types of obstacles, could be added as a new attribute (e.g. ObstacleCode) if it is important to make such distinctions. Alternatively, obstacles such as trees and buildings that shade an Atmosphere study site could be recorded in the candidate AllFeatures entity, with its relationship to this study site recorded in a SiteFeatureAssociation entity; both are discussed under the Study Site Features information category of the Strawman GLOBE Database Design.

The SurfaceCover attribute identifies the surface cover type of the study site. It is specified on page 2-65 of the GLOBE II Teacher's Guide to be one of: Paved, Bare Groud, Short Grass (< 10cm), or Long Grass (> 10 cm). This differs slightly form the SurfaceCover attribute of the SoilMoistureSite entity. They should probably be reconciled.

Other site characteristics that could have an effect on temperature readings (e.g. percent of daytime shade or CanopyCover) could be accommodated by adding appropriate new attributes to this entity.

This entity does not have a DatePerformed or DateRecorded attribute; if it is important to maintain that information, then these attributes could be added here, or a new SiteActivity entity, analogous to the DataActivity entity discussed under the Student Measurements information category, could be defined.

The Atmosphere Study Site Data Entry Sheet (page 2-65 of the Teacher's Guide) asks for the user to identify the most detailed MUC level for this study site. This information is now recorded in the LandCoverData entity described under the Student Measurements information category.

This AtmosphereSite entity will be populated in the same manner that the FSL SITELOCATION tables with corresponding site characteristic information are populated. The GLOBE Server will read data entry sheets, such as those currently specified in the GLOBE II Teacher's Guide, that have this site characteristic information and Insert a new instance into this entity, Update an existing instance, or report an error, as appropriate.

This entity is most directly related to the existing NOAA/FSL GLOBE2ATSITELOCATION table.

314 Table ATMOSPHERE_SITE_CNT

315 Table BAROM_PRESSURE

316 Table BAROM_PRESSURE_CNT

317 Table BUDBURST_METADATA

318 Table BUDBURST_META_CNT

319 Table BULLETINS

320 Table CERTMERGE_DIRECTOR

321 Table CERTMERGE_RUNGONLY

322 Table CERTMERGE_TEACHER

323 Table CHECKLIST_ITEM

324 Table CLOUD_OBS

324.1 Description of table CLOUD_OBS

The CloudObservation entity records daily cloud observations at a specific Atmosphere Study Site, including CloudCover and an indication of the presence of one or more cloud types. At least one observation should be taken within one hour of local solar noon, but other observations may be made during other daylight hours.

The school reports the UT time of the measurement and GLOBE derives the Solar times.

324.2 Annotation for table CLOUD_OBS

The foreign key FK1 references the StudySite entity, and enforces a one-to-many mapping from a study site to a collection of could observation measurements taken at that site on different days. An additional constraint is needed to ensure that the referenced site is in fact a site created for Cloud Observation measurements, i.e. a site whose MeasurementCode attribute is either AM or CO. There are various ways to enforce this additional constraint in the Physical Database Design.

This entity design assumes that cloud observations are recorded within one hour of local solar noon (see GLOBE II Teacher's Guide, Atmosphere Protocol 5, page 2-26), so it does not record an HourMeasured attribute. The GLOBE II Teacher's Guide (page 2-30) indicates that cloud observations may be made several times per day, whereas the GLOBE II Atmosphere Protocols 1&2 (pages 2-20 and 2-21) indicate that cloud observations should be made only once per day within one hour of local solar noon. What is the correct protocol? If measurements can be made multiple times per day, then we could add an HourMeasured attribute to the entity definition.

This entity probably corresponds to the existing FSL table GLOBE1EXPERIMENTS with Experiment='CO', and with the DateReported column removed (the DateReported column is kept instead in the DataActivity entity).

325 Table CLOUD_OBS_CNT

326 Table CONGRESS

326.1 Description of table CONGRESS

The Congress table identifies a U.S. House or Senate position, the current member of congress holding that position, and the staff member in that congressional office most closely associated (or potentially associated) with GLOBE activities.

327 Table CONGRESS_PEOPLE

328 Table CONTACT_LOG

329 Table CONTACT_TOPIC

330 Table COORDINATED_SITES

330.1 Description of table COORDINATED_SITES

The COORDINATED SITES table is a list of study sites that are to be coordinated for display purposes. Each SponsorId and CoordName identifies a Coordination from the COORDINATIONS table, and the SchoolId, SiteId pair identifies the list of sites from the STUDY SITE table that are to be associated with that Coordination. In most cases the study sites will be associated with the same GLOBE protocol, but not always.

331 Table COORDINATIONS

331.1 Description of table COORDINATIONS

The COORDINATIONS table identifies all coordinations known to the system. A coordination is a list of study sites that is maintained by the COORDINATED SITES table. The Sponsor of a coordination is a school or campaign that is identified by a SchoolId in the GLOBE SCHOOL table. The CoordName need be unique only within the list of coordinations being sponsored by the SponsorId.

332 Table COUNTRIES

333 Table COUNTRY_ATTENDANCE

333.1 Description of table COUNTRY_ATTENDANCE

The Country Attendance table identifies countries that have participants at a given workshop session. If known, the table also records the number of attendees from each country.

This table cannot be populated from existing information on teacher registrations at workshops for several reasons; either because registration lists for international workshops are not always available, or because registration lists do not list all those in attendance from other countries.

NOTE: An open question is whether or not to list the host country in this table as one of the countries that attended the workshop. Current thinking (Jo and Len, Jan '98) is Yes, do list the host country as attending even though this makes the Country Attendance table larger than necessary. It will include all of the in-country workshops with attendance just from that single country. We could omit the U.S. workshops from inclusion, because with rare exceptions only U.S. people attend U.S. workshops.

334 Table COUNTRY_CODES

335 Table COUNTRY_CONTACTS

335.1 Description of table COUNTRY_CONTACTS

The COUNTRY CONTACTS table maintains a list of all country points of contact, including one or more official Point(s) of Contact and one or more Country Coordinators for various GLOBE activities within the country.

The CountryCode column is the 2-char country code as in the COUNTRY CODES table.

The ContactCode column contains either PC or CC to indicate a country Point of Contact or a Country Coordinator. Note: As of Fall 2000, this was extended to allow the Help Desk to insert special codes to designate other people from that country to whom email questions from schools could be directed. (See Avnish email dated 8/21/00 and 8/28/00).

The SpecialText column is the special text used to introduce a point of contact or a country coordinator, only if that text differs from "Point(s) of Contact" or "Country Coordinator(s)".

The PersonName column contains the full name of the person as it should appear on a web page, with leading titles and suffixes. Later this information will be parsed for entry into separate fields of the GLOBE TEACHER table.

The Address column contains the full address for that person, as it is to appear on the web page, including titles, institutes, rooms, city, state/province, zip, country. At some later point this information will be analyzed to populate address information in GLOBE TEACHER and/or GLOBE SCHOOL tables.

The CountryName column does not necessarily match exactly any of the country names in the GLOBE COUNTRY table or the COUNTRY CODES table; instead, it is the country name as it is to appear on the Country Information web page. As time goes by, this

name will be reconciled with one of the country names in those tables.

The Phone column is the main telephone number if not more than 25 characters. Multiple phone numbers, with lead-in tags, can be put in the TaggedInfo column. If it fits in 25 characters, multiple numbers could be put on this line, e.g. 303-497-6751 or 6756.

The Fax column is the facimile phone number. Multiple numbers could be handled as above.

The Email column is the email address. If they fit, multiple email addresses could be put in this cell; otherwise, multiple email addresses would have lead-in tags and be placed in the TaggedInfo column.

The TaggedInfo column is for separate lines of tagged information for the person identified by PersonName, e.g. Cell Phone: 303-497-6751 <newline> Alternate Email: lg@nist.gov. Each line will be printed as a separate line on the web page under that person's name.

The ExtraStuff column is for other info that doesn't fit anywhere else! It is put here for further analysis. The contents of this column will NOT be printed on the web page.

336 Table COUNTRY_IMAGES

337 Table COUNTRY_LINKS

337.1 Description of table COUNTRY_LINKS

The Country Links table identifies the name and URL for items that the country wishes to have associated with its Globe Country information.

338 Table COUNTRY_REGION

339 Table COVER_WS

340 Table CURRENT_TEMP_NONNOON

341 Table DAILY_DIXON_OBS

341.1 Description of table DAILY_DIXON_OBS

The DAILY DIXON OBS snapshot contains the total Dixon count for each day data has been "observed". It is built directly from the data tables and hence contains the Dixon count logic.

A separate table does the counts for each day that data is "reported".

342 Table DAILY_DIXON_REP

342.1 Description of table DAILY_DIXON_REP

The DAILY DIXON REP snapshot contains the total Dixon count for each day data has been "reported". It is built directly from the data tables and hence contains the Dixon count logic.

A separate table does the counts for each day that data is "observed".

343 Table DATA_ACTIVITY

343.1 Description of table DATA_ACTIVITY

The DataActivity entity records the DatePerformed and DateReported of each student measurement activity that is performed for a specific GLOBE school study site.

This entity determines a multi-faceted relationship among Schools, Study Sites, and Measurement Groups. It can be used to answer many ad hoc data activity questions such as: number of measurements performed by a given school, measurement activity at a specific site or group of sites, number of schools participating in a given measurement group,schools with the most active students, average time lag in DatePerformed to DateReported, etc.

DataActivity entity
SchoolId
SiteId
MeasurementCode
TableType
TableName
DatabaseAction
DateHourPerformed
DateHourReported
ClientType

Every field is part of the primary key; this is to ensure that no student activity on any data entry table is missed.

A projection on SchoolId and MeasurementCode will produce a many-to-many relationship among GLOBE Schools and GLOBE Measurement Groups at which the school is actively recording student measurements. Other projections will produce analogous relationships.

The MeasurementCode attribute identifies the GLOBE Measurement Group that is affected by the given student measurement. It is a 2-character code defined by the MeasurementCodes table.

The TableType attribute identifies the type of table that is affected to be one of site characteristic, summary data, or worksheet.

The DatabaseAction attribute identifies the action to be one of Update, Insert, or Delete.

The DateHourPerformed attribute identifies the calendar Date and Hour (UT) that a student measurement at a given site was performed. Because no student measurements are taken more often than justhourly, the precision of this attribute is just to the nearest day and hour

The DateHourReported attribute identifies the calendar Dateand Hour (UT) that a student measurement or modification was reported to the GLOBE WWW Server. The precision of this attribute is to the nearest day and hour.

343.2 Annotation for table DATA_ACTIVITY

This is a proposed new entity that is not currently represented in the Globe database. This information could be kept with the other measurement entities, but maintaining it separately has the advantage of keeping the "real data" entities unencumbered. For example, the DateReported could be removed from the existing NOAA/FSL GLOBE2 data tables and kept here instead.

This entity could be populated at the same time that new data is being entered into any one of the other Student Measurement entities. All of the primary key attributes would be known by the Globe WWW Server as a result of the logon procedure to allow reporting of new student measurements, so the Server could automatically populate this entity without any additional student initiative.

NOTE: The exisitng FSL database (as of 2/11/98) is missing some information on about 10,000 records. This table can be automatically populated once those records are retrieved from the NGDC Archive.

344 Table DATA_ENTRY_OPTIONS

344.1 Description of table DATA_ENTRY_OPTIONS

The DATE ENTRY OPTIONS table provides a mapping from FSL-specific parameter names to descriptions in various languages of the numeric codes used as values for the parameters. The primary key fields are:

ParamName
Language
Code

To be completely useful to everyone else, this table will have to be supplemented with mappings to and from other system names, e.g. TableName/ColumnName, DomainName, GSFCParmName, NGDCParmName, etc. These mapppings may require some effort from anyone needing the code descriptions to be on-line in an Oracle table.

345 Table DB_MASTER_SITES

345.1 Description of table DB_MASTER_SITES

The DB Master Sites table identifies all Oracle instances of the GLOBE database. The "Cycle" column indicates the cycle to be production (PROD), tests (TEST), or expired (EXPD).

346 Table DEADLOG_WS

347 Table DENSITY_WS

347.1 Description of table DENSITY_WS

The DensityWS table records raw data soil density readings for multiple samples taken from a given Horizon at a given soil characteristics study site. The values reported for each sample are:

Volume of the sample (mL or cc's)
Weight of moist soil and container (grams)
Weight of dry soil and container (grams)
Weight of empty container (grams)
Weight of rocks in dry soil sample (grams)
Volume of water in cylinder before rocks added (mL)
Volume of water in cylinder after rocks added (mL)

The sample data is used to derive the volume of rocks (mL), soil water content (dimensionless), bulk density (g/mL or g/cc), and porosity (%). The sample values are averaged and the average reported in the RockVolume, WaterContent, BulkDensity, and Porosity attributes of the SoilData table.

348 Table DENSITY_WS_CNT

349 Table DIXON_SUMMARY_CNT

350 Table DIXON_SUMMARY_GROUPED

351 Table DIXON_SUMMARY_MC

352 Table DUFFLITTER_WS

353 Table EVALUATION_FORM

354 Table EVAL_QUESTIONS

355 Table EXPERIMENTLOG

356 Table FERTILITY_WS

356.1 Description of table FERTILITY_WS

The FertilityWS table records nitrogen, phosphorous, and potassium readings for multiple samples taken from a given Horizon at a given soil characteristics study site. The sample measurements are averaged and the average reported in the Nitrogen, Phosphorous, and Potassium attributes of the SoilData table.

357 Table FERTILITY_WS_CNT

358 Table FIREFUEL_CNT

359 Table FIREFUEL_SITE

360 Table FIREFUEL_TRANSECT

361 Table FLOWERLIST_BYSCHOOL

362 Table FLOWERSPECIES_COUNTS

363 Table FLOWERSPECIES_COUNTS_CNT

364 Table FOLLOWUP_TARGET

365 Table FRANCHISE_CONTACT

365.1 Description of table FRANCHISE_CONTACT

The FRANCHISE_CONTACT table identifies the two most important person contacts for each franchise that has a formal association with GLOBE. The Franchise Coordinator (FC) is responsible for overall administrative issues dealing with the franchise and its relationship with GLOBE, while the Training Contact (TR) is responsible for the franchise's training of teachers.

The Type column carries the labels FC or TR to identify which role the contact plays. Other roles may be added in the future. See the FranContactTypeDomain for other potential designations.

Note that Franchise contacts do not have PersonId's created for them, so these contacts cannot automatically participate in other database relationships without first being added to a table (e.g. GLOBE_TRAINER or GLOBE_TEACHER) that creates person ID's.

366 Table FRANCHISE_HOMEPAGE

366.1 Description of table FRANCHISE_HOMEPAGE

The FRANCHISE_HOMEPAGE table identifies all of the URL's for home pages of any of the organizations that make up a franchise.

367 Table GDS_MOISTCHG_30DAYS

368 Table GEOGRAPHIC_REGION

369 Table GLOBE1EXPERIMENTS

370 Table GLOBEDB_USERS

370.1 Description of table GLOBEDB_USERS

A base table containing contact information for all authorized Oracle GLOBE users at FSL.

371 Table GLOBE_CAMPAIGN

371.1 Description of table GLOBE_CAMPAIGN

The GLOBE_CAMPAIGN table is a listing of all GLOBE Schools that are participating in a GLOBE Campaign.

By agreement, a GLOBE Campaign is viewed as if it were a GLOBE School, thus every CampaignId must appear as a SchoolId in the GLOBE_SCHOOL table. Also, by convention, a GLOBE Campaign is considered to be a Participant in the campaign identified by its CampaignId, thus the rows in GLOBE_CAMPAIGN that identify the campaign itself have the property that CampaignId = ParticipantSchool. The Begin and End Dates of this special row give the overall time span of the campaign, whereas the Begin and End Dates for other participating schools give the time of their specific participation.

See the view GLOBE_CAMPAIGN_LIST for information extracted from GLOBE_SCHOOL about all GLOBE Campaigns. The SchoolType attribute of GLOBE_SCHOOL identifies the SchoolType to be GLOBE Campaign, i.e. code 9. The ParentOrganization attribute identifies the franchise, school, or local organization that is sponsoring the campaign. The LifeCycleCode is set to Active (A) if the campaign intends to define new study sites under the campaign ID; otherwise, it is set to Administrative (M).

371.2 Annotation for table GLOBE_CAMPAIGN

In general a GLOBE Campaign incorporates the following ideas:

1) GLOBE-a-thons are all regarded as GLOBE Campaigns and all receive a Campaign ID. A GLOBE Campaign has all of the characteristics of a GLOBE School, i.e. it has a mailing address, telephones, a director, etc., and is allowed to own Sample and Study Sites.

2) Each GLOBE Campaign is represented by a new entry in the GLOBE_SCHOOL

table. To make this work we'll need to decide on appropriate entries for all of the non-null columns in that table. A Campaign may or may not be allowed to define sample and study sites and report data at those sites. If it is allowed to define sites and report data, it's LifeCycleCode could be "Active" (A), otherwise it could be "Administrative" (M).

3) For each GLOBE Campaign, there is a list of GLOBE Schools that participate in the Campaign. This list of schools is maintained in the new table called GLOBE CAMPAIGN, which is really a many-to-many mapping from GLOBE SCHOOL to GLOBE SCHOOL. This means that schools can participate in multiple campaigns simultaneously. The time duration for each school's participation in each campaign is also maintained in this table. During that time interval each school receives full credit for all measurements made, if any, under the Campaign ID. The time duration for the entire campaign is given by the row where CampaignId = ParticipantSchool.

4) A Campaign may choose to maintain a list of participating schools even if that Campaign is an administrative one and doesn't define any new sites for reporting GLOBE data. This list could be advertising for other activities of the Campaign.

5) There is a notion of Coordinated Sites, separate from that of a Campaign. A list of coordinated sites (called a Coordination) has a "Sponsor" that must be an existing GLOBE School. Thus the sponsor of a Coordination could be a Campaign.

6) Coordinations are supported by two new tables in the architecture, one to identify the Coordinations themselves and track their dates of existence, and one to maintain the list of sites that participate in the Coordination. These are COORDINATIONS and COORDINATED SITES, respectively, in the attached graphic. The table COORDINATED SITES maintains a many-to-many relationship among Coordinations and Sample or Study Sites. For example, sites may participate in multiple coordinations.

7) Each Campaign will require a minimum of one row in the GLOBE CAMPAIGN table, namely itself as a ParticipantSchool in the Campaign. Thus a projection on the CampaignId column will produce a complete list of GLOBE Campaign ID's. Note that a Campaign has no name in the GLOBE CAMPAIGN table; instead, it's name is the SchoolName from the GLOBE SCHOOL table. Thus it's probably desirable to have a view defined over GLOBE SCHOOL to identifiy just those schools that are really campaigns. That view is labeled as GLOBE CAMPAIGN LIST on the attached graphic.

8) It's not clear whether or not it is desirable to support a hierarchy of Campaigns. for example, may Campaign A join in as a participant in Campaign B. From a database structure point of view this is not a problem, but trying to assign measurement credits accurately could turn into a recursive nightmare. I don't think this should be supported at the beginning!

372 Table GLOBE_CONCEPT

373 Table GLOBE_COUNTRY

373.1 Description of table GLOBE_COUNTRY

The GLOBE COUNTRY table is the definitive list of countries that have signed a GLOBE agreement, together with the date that agreement was signed.

It also contains information about that country such as the number of schools in the country, the beginning and end of the normal school term, a pointer to the GLOBE home page for that country, and the country's LifeCycleCode and Date of the last LifeCycle modification.

The Comments column is reserved for a specail use! It will contain any special text that introduces that country's GLOBE Home Page.

374 Table GLOBE_FRANCHISE

374.1 Description of table GLOBE_FRANCHISE

The GLOBE FRANCHISE table identifies all of the franchise organizations, or consortiuums, that have a relationship with GLOBE. If a formal agreement has been signed, then the begin and end dates of that agreement are recorded. The LifeCycle designation identifies a franchise as Candidate (C), Active (A), Inactive (I), or Administrative (M).

As of Fall 1998, all franchises are labeled as either Active or Inactive. It is not known at this time if the Administrative designation makes sense here.

In Early 2001, the term "Franchise" was changed to "US Partner". However, the GLOBE FRANCHISE and FRANCHISE CONTACT tables will remain unchanged. All new develoment will use the new term.

In April 2001, GLOBE HQ introduced the term "Affiliate" to refer to organizations that work with a US Partner to achieve common goals, but who themselves are NOT "US Partners". It is not clear yet if we need to invent a new LifeCycle code for Affiliates, or if the "Administrative (M)" code can serve this purpose. Dixon says that an "Affiliate" will work with exactly one "US Partner" so there is no need to introduce a many-to-many Affilate-Partner relationship. If we can rely on this one-to-many relationship, then the Affiliate-to-Partner relationship can be supported with the addition of just one new column, e.g. "ParentPartner"; the new column will be null if the row identifies a "US Partner" and will be non null if the row identifies an Affiliate.

375 Table GLOBE_MAILING

376 Table GLOBE_ORDER

376.1 Description of table GLOBE_ORDER

The GlobeOrder table is a listing of all GLOBE product orders related to workshop sessions and warehouses that maintain supplies of products for workshop sessions. Each row of the table identifies an order, the vendor the order is sent to, the vendor or workshop session the order is to be shipped to, and other shipping requirements and instructions. The individual line items of the order are maintained in the OrderItems table.

Orders can be of the following different types:

Warehouse to Workshop Session - An order is sent to a warehouse asking that materials or equipment be sent to a workshop session. Such orders will decrease the QuantityOnHand in the GlobeProduct table. Sometimes such orders will be sent to an intermediate place, e.g. GLOBE, a Franchise, or a Country Coordinator, for subsequent distribution to a workshop session.

Vendor to Warehouse - An order is sent to a source vendor asking that materials be sent to a warehouse. Such orders will increase the QuantityOnHand in the GlobeProduct table

Vendor to Workshop Session - An order is sent to a source vendor asking that materials be sent directly to a workshop session. Such orders will have no effect on the QuantityOnHand in the GlobeProduct table. This type of order should be quite rare!

Warehouses and source vendors are both considered to be GLOBE vendors and are listed in the GlobeVendor table.

If the order is to be shipped to a warehouse, then the OrderToVendor column identifies the source vendor that the order is sent to and the ShipToVendor identifies the warehouse.

If the order is to be shipped to a workshop session, then the OrderToVendor identifies the warehouse (or rarely the source vendor) and the ShipToSession and ShipToContact identify the workshop session and the conatct at the workshop site that the order is to be shipped to.

If an order is to be shipped to an intermediary before subsequent distribution to a workshop session, then the ShipToVendor identifies the intermediary and the ShipToSession and ShipToContact identify the ultimate workshop destination.

The Instructions column is a text field that identifies any instructions to the vendor that are included with the order.

The comments column is intended for comments to be seen only by GLOBE.

377 Table GLOBE_PRODUCT

377.1 Description of table GLOBE_PRODUCT

The GlobeProduct table identifies various commercial products used in GLOBE operations. Some products might be binders and notebooks used in teacher training workshops; other products may be modems or computer hardware and software products send to GLOBE schools.

Each product is identified by a ProductCode that is the primary key of this table and a foreign key in other tables that reference the GlobeProduct table (e.g. WorkshopSupplies).

378 Table GLOBE_SCHOOL

378.1 Description of table GLOBE_SCHOOL

The GlcbeSchool entity is the primary repository of GLOBE School inforaation. It identafies the name, address, and contact information for each school the school director, and various school attributes. The school attributes will often be used by students to help identify other GLOBE schools that are "most like" their own.

GlobeSchool entity
- SchoolId (primary key)
- SchoolName
- SchoolNameLocal
- SchoolGrades
- SchoolFunding
- SchoolType
- UrbanizationCode (FK4)
- NumberOfStudents
- AddressLine1
- AddressLine2
- City
- StateRegionCode (FK2)
- PostalCode
- CountryCode (FK1, FK2)
- Timezone
- ZoomLevel
- DirectorFullName
- DirectorJobTitle
- DirectorSalutation
- ParentOrganization
- SchoolTelephone
- SchoolFax
- SchoolEmail
- SchoolHomePage
- IPprovider
- IPpayer
- InternetAccess
- BrowserType
- LastLoginDate
- LifeCycleCode (FK5)
- CycleChangeDate
- Comments

The SchoolId attribute is a globally unique identifier for a GLOBE School. SchoolId is an 8-character field structured so that the first two characters are the country code of the country in which the school is located and the second two characters are derived from the State or Province in which the school is located. SchoolId's will be used extensively as foreign Keys in other database entities to reference a GLOBE School that is an instance of this GlobeSchool entity.

The SchoolName attribute identifies the full formal name of the school using characters from the ISO Latin-1 character set. The Latin-1 character set is an International Standard (ISO 8859-1) that defines the 191 characters that form the basis of most Latin based languages, including: Danish, Dutch, English, Faeroese, Finnish, French, German, Icelandic, Irish, Italien, Norwegian, Portuguese, Spanish, and Swedish. The maximum
length of the SchoolName attribute will be long enough to capture the full Latin-1 name of every GLOBE School.

The SchoolNameLocal attribute identifies the full formal name of the school using characters from some multi-byte character set that can represent nearly every language in the world, including: Chinese, Arabic, Greek, Russian, Korean, Japanese, etc. Initially this attribute may be represented with the 2-byte UNICODE character set, with evolution as appropriate to the new International Standard for 4-byte character sets (ISO/IEC 10646). Support for this attribute is dependent upon the character set capabilities of the underlying GLOBE WWW and Database Servers and upon the presentation capabilities of the WWW browser used by the GLOBE School. The maximum length of the SchoolNameLocal attribute will be long enough to capture the full local language name of every GLOBE School.

The SchoolGrades attribute uses a list of SchoolGradeCodes to identify the grade level and approximate ages of students at the identified GLOBE School. A value of this attribute will be a short list (from 1 to 4 characters) of school grade/age codes defined in the SchoolGradeCodes entity. The allowable codes, each with its code name and a short description, are as follows:

P	Primary	Grades K-3, approximate ages 5-10
M	Middle	Grades 4-6, approximate ages 9-13
U	Upper	Grades 7-9, approximate ages 12-16
H	High School	Grades 10-13, approximate ages 15-19
C	College/Univ	Grades 12+, approximate age >17

For example, a Web data entry form could identify the grade groupings and allow a school to pick one or more of them to identify the grades of students in that school who will be participating in GLOBE experiments. A classical grade school having grades 1 to 8 might choose codes P, M, and U, resulting in a SchoolGrades value of "PMU", or a combined Junior-Senior high school might choose codes U and H, resulting in a SchoolGrades value of "UH". To avoid non-unique representations of the same information, the GLOBE Database Server could ensure that code lists (e.g. "UH" and "HU") are always ordered and stored the same way.

The SchoolFunding attribute identifies the primary type of funding that support the school. A school is either Publicly Funded or Privately Funded.

The SchoolType attribute identifies the school type to be one of: No Special Type (this is the default), Charter School, Magnet School, Home School, School for Special Needs, District Office, Science Center, Museum, State/National Park, Other. Privately funded schools are not classified by the type of private funding they receive or by the religious group they might be affiliated with. The school is asked to choose the single type that BEST describes their school.They should not check multiple types, even though that might make sense for some schools. .

The UrbanizationCode attribute is a 2-character code that identifies the urban environment in which a given school is located. It is a foreign key reference (FK4 above) to the UrbanizationCodes entity. An urban environment may vary from dense urban, i.e. co-located with multi-story office buildings in a large city, to a small town or village. The allowable codes are defined in the UrbanizationCodes table.

The NumberOfStudents attribute identifies the number of students as an indicator of school size.

The City attribute identifies the city, town, or other political area where the GLOBE school is located.

The StateRegionCode attribute combined with the CountryCode attribute is a foreign key reference (see FK2 above) to the StateRegionCodes entity, which gives the complete name and correct spelling of the political subdivision represented by this code. In the U.S. the StateRegionCode is the standard 2-character code for identifying a state; in other countries the allowed codes are given by instances of the StateRegionCodes entity.

The PostalCode attribute is a postal mailing code, if any, for mailing addresses in the country where the school is located.

The CountryCode attribute is a foreign key reference (FK1 above) to the CountryNameCodes entity, which gives the complete name and correct spelling in various GLOBE supported languages of the country identified by this code.

The DirectorFullName attribute identifies the full name, complete with appropriate titles, if desired, of the administrative head of the identified GLOBE school. Examples are: Dr. Elizabeth L. Green; Professor Heinrich Kruger; Peter M. Jones, M.Ed.; etc. This attribute is constrained to be a character string of length no more than 30 characters.

The DirectorJobTitle attribute identifies the administrative title of the school director. Examples are: Principal, Headmaster, School Master, Chairperson, Director, etc. There is no pre-specified list of possible job titles; instead, each school is free to identify the most appropriate title for its administrative leader. This attribute is constrained to be a character string of length no more than 20 characters.

The DirectorSalutation attribute identifies a shorter form of the school director's name to be used in greetings or other references where a full name is not appropriate. Examples are: Mrs. Green, Dr. Kruger, or Mr. Jones. The main use of the DirectorSalutation will be the greeting in a letter written to the school director (e.g. Dear Mrs. Green or Dear Dr. Kruger). This attribute is constrained to be a character string of length no more than 20 characters.

The ParentOrganization attribute identifies, by name, the organization that oversees the administration of the school. For public schools it might be the name of the local city or county school system, for parochial schools it might be the name of the church, synagogue, or mosque that operates the school, or for sponsored schools it might be the name of the funding organization. This attribute is constrained to be a character string of length no more than 40 characters.

The GlobeCoordinatorId attribute is a foreign key reference (FK3 above) to the GlobeTeacher entity. It identifies the GLOBE Teacher who coordinates GLOBE activities at the identified school. This teacher will be the most frequent recipient of various mailings from GLOBE administrative offices.

The SchoolTelephone attribute identifies the full international telephone number of the school's main telephone.

The SchoolFax attribute identifies the full international telephone number of the school's Facsimile telephone, if any.

The SchoolHomePage attribute identifies the Internet URL for World Wide Web access to the school's home page, if any. For example: http://www.globe.gov.

The AppNum attribute identifies a previous primary key for this school in the Paradox database. It still has some legacy value (e.g. entry number for hard copy files in filing cabinets).

The IPprovider attribute identifies the name of the service that provides an Internet connection to the school.

The IPpayer is a code to indicate who pays for the Internet service.

The InternetAccess attribute identifies whether or not a school has Intyernet access; it is populated from the legacy Schoolflag.Messagable attribute.

The LastLoginDate attribute identifies the date of the last school login with a WWW Browser; this field is updated automatically each and every time a Globe School logs on to the FSL system.

The BrowserType attribute identifies the Web browser used by the school (e.g. Netscape or Explorer); this field is updated automatically each and every time a Globe School logs on to the FSL system.

The LifeCycleCode attribute is a 1-character code that identifies the life-cycle status of a given school; it is a foreign key reference (FK5 above) to the LifeCycleCodes table.

The CycleChangeDate attribute identifies the most recent date that the LifeCycleCode attribute changed. For "Active" GLOBE Schools, this is the date when the school was formally accepted as a GLOBE School.

The Comments attribute provides an opportunity for the identified GLOBE School to tell the world something about itself. For example, a school could record that: "We are the first school in Finland to be registered as a GLOBE school".

378.2 Annotation for table GLOBE_SCHOOL

The GlobeSchool entity has many attributes that will seldom be used together in the same applications. For example, school address and telephone information will most often be used by the GLOBE administrative staff, the GPS maximum and minimum values will be used as integrity constraints for other GPS measurements related to study sites at this school, and the school grades, school type, and number of students attributes will most often be used by students looking for schools most like their own. During the GLOBE Physical Database Design, it may be desirable to partition the GlobeSchool entity (either horizontally by rows or vertically by attributes) for performance optimization; the resulting pieces may be represented in relational tables that are owned by different groups and stored in different primary locations.

379 Table GLOBE_STARS

380 Table GLOBE_TASK

381 Table GLOBE_TEACHER

381.1 Description of table GLOBE_TEACHER

The GlobeTeacher entity is the main repository of GLOBE Teacher information. It identifies the name and personal contact information for each teacher, their life-cycle status, and the current GLOBE school with which the teacher is affiliated.

This entity does not identify the school class or classes that the teacher is associated with; instead, there exists a separate TeacherToClass association entity that maintains a many-to-many relationship between Teachers and Classes. Classes will also have associations with other entities, e.g. Study Sites, so Teachers will thereby have indirect associations with multiple other entities as well.

This entity does not identify the training status or last training date of a teacher; instead, there exists a separate Registration table that maintains a many-to-many relationship among teachers and workshop training sessions.

```
GlobeTeacher entity
  TeacherId          (primary key)
  CurrentSchoolId      (FK3)
  InternalAddress
  TeacherPrefix
  TeacherSurname
  TeacherOtherNames
```

TeacherSuffix
WorkTelephone
HomeTelephone
TeacherFax
Email
HomePage
HomeAddressLine1
HomeAddressLine2
HomeCity
StateRegionCode (FK2)
PostalCode
CountryCode (FK1, FK2)
LifeCycleCode (FK4)
CycleChangeDate
RecruitedBy
GrantTraining
Comments

The TeacherId is globally unique; it identifies a teacher permanently, and stays fixed even if a teacher moves from one school to another. TeacherId will likely consist of a field of 8 to 12 characters; part of that field may be used to identify a GLOBE Franchise organization so that Franchises could assign TeacherId's independently, without fear of duplicating the Id assigned by some other Franchise. It is possible to have TeacherId's generated automatically by the GLOBE Database Server if that is desirable, probably derived as a hash code from the teacher's name. TeacherId's will be used as Foreign Keys in other database entities to reference a GLOBE Teacher that is an instance of this GlobeTeacher entity.

The CurrentSchoolId attribute is a foreign key (FK3 above) that references the GlobeSchool entity. It identifies a single GlobeSchool that the teacher is currently associated with; the mailing address of that school serves as the preferred school mailing address for the teacher. The identification of a single school as the CurrentSchool of a teacher does not prohibit the TeacherId from being associated with School Classes at other schools in the TeacherToClass association entity; in this manner there may be an additional implicit many-to-many relationship between Schools and Teachers.

The InternalAddress attribute identifies a room number or other internal school address for the teacher.

The Teacher Prefix identifies a title used as part of the teacher's nasme, e.g. Mr. Ms. Dr. Prof. Inst., etc.

The TeacherSurname is the legal family name of the teacher.

The TeacherOtherNames identifies First and Middle names for the teacher, if desired, or initials as used my the teacher is their name. It is combined with the TeacherSurname to derive the full name of the teach when needed.

The TeacherSuffix is a trailing title that may be used by a teacher, e.g. Jr., III, Esquire, PhD, MEd, etc.

The WorkTelephone and HomeTelephone attributes identify work and home telephone numbers for the teacher. Either may be left unspecified, if desired.

The TeacherFax attribute identifies the Fax number for the teacher, if any.

The Email attribute identifies the single email address, if any, that is most appropriate for sending an email message to the teacher.

The HomePage attribute identifies the URL of a HomePage that is most suitable for the teacher; it might be the school HomePage if that contains information about teachers, or it might be a personal HomePage if that is better, or it might be left unspecified.

The Home mailing address attributes HomeAddress1, HomeAddress2, HomeCity, StateRegionCode, PostalCode, and CountryCode, serve the same purpose for the home mailing address of a teacher as do the analogous mailing address attributes for the teacher's school. It may be necessary to maintain home address information for GLOBE teachers in the database so that teachers can be contacted if necessary by GLOBE staff during school vacation times. These attributes need not be specified if they are not needed for that purpose, i.e. usually for last minute scheduling of summer GLOBE training workshops.

The LifeCycleCode attribute is a 1-character code that identifies the life-cycle status of a given teacher; it is a foreign key reference (FK4 above) to the LifeCycleCodes table.

The CycleChangeDate attribute identifies the most recent date that the LifeCycleCode attribute changed. For "Active" GLOBE Teachers, this is the date when the teacher was formally accepted as a GLOBE Teacher.

The RecruitByCode identifies by category (e.g. Web, Globe, Site) how the teacher was originally recruited. The number of codes is quite small, so a RecruitByText attribute lets the data entry form add any additional relevant information.

The GrantTraining attribute identifies if the teacher's GLOBE training was paid for by a grant, or not!

The Comments attribute provides an opportunity for the identified GLOBE Teacher to tell the world something about themselves. For example, they might state the reason for their involvement with the GLOBE project.

381.2 Annotation for table GLOBE_TEACHER

The GlobeTeacher entity has some attributes that will be used for very different purposes. For example, the personal contact information may get used mostly by GLOBE administrative staff, whereas the TeacherSalutation may get used extensively in "personalization" of GLOBE WWW Server interactions. During the GLOBE Physical Database Design, it may be desirable to partition the GlobeTeacher entity (either horizontally by rows or vertically by attributes) for performance optimization; the resulting pieces may be represented in relational tables that are owned by different groups and stored in different primary locations.

382 Table GLOBE_TRAINER

382.1 Description of table GLOBE_TRAINER

The GlobeTrainer table identifies training staff that will conduct GLOBE teacher training workshops. The relationship between trainers and teacher training workshops is maintained by the WorkshopStaff table. The Role that the trainer plays at a specific workshop is also maintained in the WorkshopStaff table.

383 Table GLOBE_VENDOR

383.1 Description of table GLOBE_VENDOR

The GlobeVendor table identifies various commercial vendors that do business with GLOBE. These may be vendors that supply training materials to teacher training workshops, vendors that supply computer and other scientific products to GLOBE schools, or warehouse vendors who store either of the above types of products in a warehouse for distribution to schools or workshop sessions.

384 Table GPS_METADATA

385 Table GPS_WORKSHEET

386 Table GRASSBIOM_WS

386.1 Description of table GRASSBIOM_WS

The GrassBiomWS table records Green and Brown grass Biomass for multiple samples at a given Land Cover or Biology study site. The multiple measurements are then averaged and the result stored in the GreenBioAvg and BrownBioAvg attributes of the GrassBiometry table.

387 Table GRASSBIOM_WS_CNT

388 Table GRASS_BIOMETRY

388.1 Description of table GRASS_BIOMETRY

The GrassBiometry table records ground cover measurements (by counting squares), derived ground cover percentages, and average Green and Brown grass Biomass at a given Land Cover or Biology study site. The Green and Brown Biomass averages are derived from data taken from multiple samples stored in the GrassBiomWS table.

389 Table GRASS_BIOMETRY_CNT

390 Table GRASS_TYPES

391 Table GREENDOWN_WS

391.1 Description of table GREENDOWN_WS

In the GREENDOWN WS table, the LeafColorCode comes from the GLOBE Plant Color Guide. The DyingState is a code that identifies one of several different stages that a leaf or blade of grass may go through during GreenDown. See the DyingState domain for a list of allowed values.

392 Table GREENDOWN_WS_CNT

393 Table GREENING_METADATA

393.1 Description of table GREENING_METADATA

In the GREENING METADATA table, a GreeningId is assigned by GLOBE. It identifies each tree or shrub, or square-meter of turf, selected for the greening protocol. There may be a number of different GreeningId's for each site, e.g. a class of 20 students may pairup to study greening on 10 different trees, shrubs, or grasses. The students can give a GreeningLabel to each tree, shrub, or grass patch. This name is just to help them remember which tree is which. GLOBE will assign the more efficient GreeningId to each item, and will maintain the association between GreeningId and GreeningLabel. It's redundant information, but the TreeGrassShrub attribute is just there to help classify the item as a Tree(T), Grass(G), or Shrub(S). It will help categorize queries on this information if it is consistently provided. It will save having to lookup the Genus/Species to see if the item is a tree, grass, or shrub. Each GreeningId can represent a different Genus/Species! Or they could all be the same. The CommonName just records what the students say is the local name of the specified Genus/Species. The DateDefined is the Date the greening metadata was observed. This date will become more important if additional metadata (e.g. tree height and diameter) is added later.

394 Table GREENING_METADATA_CNT

395 Table GREENUP_WS

395.1 Description of table GREENUP_WS

In the GREENUP WS table, the YEAR is the Calendar year in which the Greening protocol begins. If the observations overlap a calendar year, choose the year of the start of bud swelling. The GreeningCycle identifies the 1st, 2nd, etc. greening cycle that calendar year. By default it is 1. But some grasses may have multiple cycles per year, and even some trees or shrubs could have multiple cycles because of drought or disease. The LeafId identifies the leaf of a tree or shrub, or the blade of grass, to be measured. For trees and shrubs, the leaf order (e.g. 1,2,3,4) varies from outermost to innermost. Larger numbers could be used if a leaf is destroyed before completion of the protocol and a new leaf is used, e.g. leaf #5 replaces leaf #2. The students would clarify the substitution by a Comment!

The GreeningState is a code that identifies one of several different stages that a leaf or blade of grass may go through during GreenDown. See the GreeningState domain for a list of allowed values.

396 Table GREENUP_WS_CNT

397 Table GROUP_USES_COUNT

398 Table HAZE

398.1 Description of table HAZE

The HAZE table reports the results of Haze-related measurements taken during a specific time period, usually 7 to 10 minutes, one or more times per day at a specific Atmosphere study site. The UTtimeMeasured is the start time of the multiple measurements, and that value is then translated to SolarDay, Hour, and Minute for the primary key values of the table. Measurements taken just once during the time period are stored in the HAZE table and measurements sampled multiple times are stored in the HAZE WS table. The attributes ResultAOT and ResultTransPct in the HAZE table are derived from relevant measurements in the HAZE WS table. The results of the required associated Cloud Observation are stored in the CLOUD OBS table with the same primary key.

399 Table HAZE_CNT

400 Table HAZE_WS

400.1 Description of table HAZE_WS

The HAZE WS table records multiple haze measurements taken during a specific time period, usually 7 - 10 minutes, one or more times per day. The Solar times in the primary key identify the start time of the measurements, whereas the UTtimeMeasured for each instance identifies the time each set of measurements was taken, with precision up to 1 second. A single AOT and TransmissionPct representing the entire set of measurments is stored in the HAZE table. The SampleNbr has no semantics and simply identifies distinct sets of measurements. The PhotometerId points to a specific calibrated photometer and the ChannelId, Red(R) or Green (G), identifies a specific channel of that photometer. The geographic location of the study site, the air temperature and barometric presssure from the

Haze table, and the ETconstants from the PHOTOMETER CALIB table are used to calculate the AOT. The TransmissionPct is derived from the AOT.

401 Table HAZE_WS_CNT

402 Table HR_ATMOSADV_CURRENT

403 Table HR_ATMOSADV_PREVIOUS

404 Table HR_ATMOS_CURRENT

405 Table HR_ATMOS_PREVIOUS

406 Table HR_CLIMATE_CURRENT

407 Table HR_CLIMATE_PREVIOUS

408 Table HR_CLOUDS_CURRENT

409 Table HR_CLOUDS_PREVIOUS

410 Table HR_EARTHSYS_CURRENT

411 Table HR_EARTHSYS_PREVIOUS

412 Table HR_HYDRO_CURRENT

413 Table HR_HYDRO_PREVIOUS

414 Table HR_LCBIO_CURRENT

415 Table HR_LCBIO_PREVIOUS

416 Table HR_OVERALL_CURRENT

417 Table HR_OVERALL_PREVIOUS

418 Table HR_PANGLOBE_CURRENT

419 Table HR_PANGLOBE_PREVIOUS

420 Table HR_SOILCHAR_CURRENT

421 Table HR_SOILCHAR_PREVIOUS

422 Table HR_SOILMT_CURRENT

423 Table HR_SOILMT_PREVIOUS

424 Table HUMIDITY

424.1 Description of table HUMIDITY

The HUMIDITY table records the Humidity measurements taken daily, within 1 hour of local solar noon, at a specific Atmosphere study site. The DryBulb and WetBulb temperatures are used to calculate the Humidity. The procedure used to calculate the Humidity is identified by HumidityMthd. The DewPoint is derived from ??? In the future, multiple measurements per day could be accepted simply by adding SolarHour and SolarMinute to the primary key.

425 Table HUMIDITY_CNT

426 Table HUMMINGBIRD_COUNTS

427 Table HUMMINGBIRD_COUNTS_CNT

428 Table HUMMINGBIRD_SITE

429 Table HUMMINGBIRD_SITE_CNT

430 Table HUMMINGBIRD_SPECIES

431 Table INFILTRATION_WS

431.1 Description of table INFILTRATION_WS

The InfiltrationWS table records raw flow rate data using to determine the rate of absorption of water into soil (mm/min). Infiltration flow rates are more fully described under the SoilInfiltration table. Up to nine measurements are taken for each of several different samples. Each set of measurements determines a curve (distance vs time) from which a saturated flow rate can be derived.

431.2 Annotation for table INFILTRATION_WS

The foreign key FK1 references the StudySite entity, and enforces a one-to-many mapping from a study site to a collection of air temperature measurements taken at that site on different days. An additional constraint is needed to ensure that the referenced site is in fact a site created for Air Temperature measurements, i.e. a site whose MeasurementCode attribute is either AM or AT. There are various ways to enforce this additional constraint in the Physical Database Design.

This entity corresponds directly to the existing FSL table GLOBE2AT, but with the DateReported column removed (that attribute is maintained instead in the DataActivity entity).

432 Table INFILTRATION_WS_CNT

433 Table INTERNATIONAL_LOG

433.1 Description of table INTERNATIONAL_LOG

The InternationalLog table identifies all correspondence related to International workshop sessions. It includes invitation mailings to partner and non-partner countries as well as logistics letters to the workshop site in the host country.

434 Table INVESTIGATION_CODES

434.1 Description of table INVESTIGATION_CODES

The InvestigationCodes entity defines a 5-character code to represent GLOBE Investigations, and maps those codes to appropriate Names and Descriptions as presented in the GLOBE Teacher's Guide.

The GLOBE Investigation Codes are as follows:

SCHOL Overall GLOBE School Investigations
ATMOS Atmosphere Investigation
HYDRL Hydrology Investigation
LCBIO LandCover/Biology Investigation
SOILS Soil Investigation

The "SCHOL" code is used to identify an investigation activity that does not fit into one of the specifically defined Investigation categories, e.g. establishing the overall 15 km x 15 km GLOBE School Study Site and taking the GPS measurements to locate the main entrance to the school.

The Description attribute may give a textual description of the identified Investigation, derived from text in the appropriate Teacher's Guide, or it may be a direct reference into the appropriate sections of that guide. The choice is determined by whether or not the Teacher's Guides have been successfully integrated into the GLOBE Database.

435 Table LANDCOVER_CODES

435.1 Description of table LANDCOVER_CODES

The LandCoverCodes entity maps a Modified UNESCO Classification Scheme (MUC) Code to a MUC Name and a MUC Description. It captures all of the information in the Glossary of Terms from the GLOBE II Teacher's Guide (pages 8-41 through 8-63). It is also an appropriate location for maintaining non-English descriptions of MUC Codes if that is desirable.

LandCoverCodes entity
- MUCcode (primary key)
- MUCname
- ClassLevel
- Description
- Description-French
- Description-Russian
- Description-Arabic
- Description-Chinese

The primary key identifies a specific MUC code; these codes vary from 1-digit in length to 4-digits.

The MUCname attribute identifies the name of a given MUCcode; it is the name given by the "Glossary of Terms in the Modified Classification Scheme" presented in the GLOBE II Teacher's Guide.

The ClassLevel attribute identifies the "Class Level" of the MUC code as presented in the GLOBE II Teacher's Guide. It is constrained to be a single digit, either "1", "2", "3", or "4". It is also constrained to be equal to the number of digits in the MUC code being described.

The Description attribute contains the descriptive English language text for each MUC code as presented in the GLOBE II Teacher's Guide. The other associated Description attributes give MUC code descriptions in other GLOBE supported languages, if available.

436 Table LAND_COVER

436.1 Description of table LAND_COVER

The LandCover table records either a MUCcode or an AndersonCode estimate of the land cover for a given study site. Land cover codes may be defined for many different types of study sites, including atmosphere, biology, land cover, soil, and surface water.

Anderson codes are used for GLOBE Phase I study sites and MUC codes are used for GLOBE II study sites.

437 Table LAND_COVER_CNT

438 Table LEAF_COLOR

438.1 Description of table LEAF_COLOR

The LEAF_COLOR table identifies the list of colors used by GLOBE in the Phenology Greening protocol for tree leaves and grasses. As of March 2001 there were 24 colors in the list.

439 Table LEARNING_STANDARD

440 Table LETTERMERGE_DIRECTOR

441 Table LETTERMERGE_RUNGONLY

442 Table LETTERMERGE_TEACHER

443 Table LIFECYCLE_CODES

443.1 Description of table LIFECYCLE_CODES

The LifeCycleCodes entity describes codes used to identify the life-cycle status of a given GLOBE school or teacher. The life-cycle status of a school or teacher usually ranges from "candidate" to "active participant" to "inactive" or "retired". Other possibilities are "administrative", for a school, school office, or teacher that plays only an administrative support role and never intends to become an official GLOBE school or teacher, or "other", for a life-cycle status not captured by one of the specific codes.

The defined codes as of July 1997 are:

P Pre-Candiate -- This is the status assigned to a brand new teacher or school that has just been added to the database; a new SchoolId or TeacherId has been assigned to the new entity but no checks have yet been made to determine if it duplicates some previous entry. After duplicate checking is completed, the status changes from P to C.

C Candidate -- This is the status of a school or teacher that has expressed an interest in the GLOBE program, but has not yet completed GLOBE training.

A Active -- This is the status of a teacher who has completed GLOBE training, or of a school with at least one teacher having completed GLOBE training, It is not necessary that the school has begun sending data, the sending of data is recorded in a different attribute.

I Inactive -- This is the status of a teacher who has retired or otherwise left the GLOBE program, or of a scxhool that has stopped sending data.

M Administrative -- This is the status of GLOBE staff, advisors, and other people that are listed in the GlobeTeacher table, who are not classroom teachers of GLOBE students.

O Other -- This identifies a status not covered by any of the above.

```
LifeCycleCodes entity
  LifeCycleCode        (primary key)
  LifeCycleName
  Description
  Description-French
  Description-Russian
  Description-Arabic
  Description-Chinese
```

The LifeCycleCode attribute is a specific 1-character code; it is the primary key of this entity.

The LifeCycleName attribute identifies the name of a given LifeCycleCode; the list of allowable codes and names, with a brief description, is as follows:

```
C   Candidate     A candidate school or teacher already involved
                  in the application process toward becoming an
                  active registered GLOBE school or teacher.

A   Active        An officially accepted and recognized GLOBE
                  school or teacher.

I   Inactive      A formerly active GLOBE school or teacher that is
                  now inactive, or retired, and no longer participates
                  in GLOBE projects.

M   Admin         An administrative school, school office, or teacher
                  that is involved in GLOBE projects but is not a
                  candidate to become an active GLOBE school or teacher.

O   Other         Some other life-cycle status not captured by one of
                  the specifically defined codes.
```

The Description attribute contains the descriptive English language text for each LifeCycleCode; the other associated Description attributes give life-cycle status code descriptions in other GLOBE supported languages, if available.

444 Table LILAC_METADATA

444.1 Description of table LILAC_METADATA

The LILAC METADATA table records information about individual shrubs planted at a Lilac Phenology study site. The information includes whether the lilac shrub is Common or Clonal, its Genus/Species, the date it is planted, the date it dies or is removed from the experiment, its initial Height at planing time, and its relative location from the center of the group of shrubs.

The center of the site is the point where the GPS reading is taken to determine the site location.

The initial shrub height is recorded in order to help estimate the age of common lilacs that are purchased from a commercial nursery.

The relative location is recorded to help identify a shrub if its ID is lost or forgotten. It might also help explain any observed differences in phenophases among shrubs of the same type. See the DirectionAngle and DirectSource domains for further explanation.

445 Table LILAC_METADATA_CNT

446 Table LILAC_PHENOPHASE

446.1 Description of table LILAC_PHENOPHASE

The LILAC PHENOPHASE table records the 5 important phenophase Dates (and associated Last Observation Days) for each shrub at the lilac study site. The important phenophase events are First Leaf, Full Leafing, first Bloom, Full Bloom, and End of Bloom.

The ShrubId can be used to identify the shrub metadata in the LILAC METADATA table.

The LastObservation dates help determine the accuracy of the observed phenophase dates. The most accurate observations are those where the phenophase date minus the last observation date is as small as possible.

447 Table LILAC_PHENOPHASE_CNT

448 Table LISTLEVEL

448.1 Description of table LISTLEVEL

The LISTLEVEL table is used to build the lists of countries and state/provinces for countries in the FSL text-based school search interface. This table may get replaced in Spring 2001 by a more universal school search mechanism.

449 Table MAILING_EFFECT

450 Table MAILING_STATUS

451 Table MALE_BEHAVIOR

452 Table MASSMAIL

452.1 Description of table MASSMAIL

The MASSMAIL table contains a copy of all mass/bulk mail messages sent out by HQ staff, Partners, and Country Coordinators.

453 Table MEASURE_AT_SITE

454 Table MEASURE_CODES

454.1 Description of table MEASURE_CODES

The MeasurementCodes entity defines a 2-character code to represent GLOBE Measurement groups as defined by protocols in a specific GLOBE Investigation.

The 2-part primary key allows the same code to be used to represent measurements in different GLOBE Phases. For example, "SI" may be used to represent "Species Identification" measurements in both GLOBE I and in GLOBE II, but the description of that measurement may differ in the GLOBE I and GLOBE II Teacher's Guides.

The GLOBE Measurement Codes as of Sept '98 include:

GP GLOBE School Study Site (15km x 15km) GPS Measurement

AT Air Temperature Measurements from Atmosphere Investigation
CO Cloud Observation Measurements from Atmosphere Investigation
PR Precipitation Rain Measurements from Atmosphere Investigation
PS Precipitation Solid Measurements from Atmosphere Investigation
SW Surface Water Measurements from Hydrology Investigation
LC Land Cover MUC Measurements from LandCover/Biology Investigation
SI Species Identification Measurements from Land Cover/Biology
LL Land Cover QuaLitative
LN Land Cover QuaNtitative
TB Tree Biometry Measurements from Land Cover/Biology Investigation
GB Grass Biometry Measurements from Land Cover/Biology Investigation
SM Soil Moisture Measurements from Soil Investigation
SC Soil Characterization Measurements from Soil Investigation
SF Soil InFiltration measurements
BB BudBurst phenology measurements
SP Snowpack phenology measurements
BI Biometry measurements from GLOBE I+
BM Biometric measurements from GLOBE I

The GlobePhase attribute identifies the version of the GLOBE Teacher's Guide from which the Name and Description of the identified measurement was first taken. Later phases are expected to be upward compatible with it.

The Description attribute may give a textual description of the identified Measurement group, derived from text in the appropriate Teacher's Guide.

The IsCurrent attribute identifies whether or not the identified MeasureCode is currently being used by GLOBE Data Entry pages and reporting mechanisms.

The TeachersGuidePages attribute identifies pages in the current "hard-copy" version of the GLOBE Teachers Guide where the protocols for that code are described. These values will be updated with each new published version of the Teacher's guide. A history of past pages is not kept.

The TeachersGuideURL attribute points to the "on-line" version of the GLOBE Teachers Guide where the protocols for that code are described. These values will be updated with each new published on-line version of the Teacher's guide. A history of past references to older on-line versions is not kept.

455 Table MEASURE_COUNTS

456 Table MEASURE_GROUP

457 Table MESSAGELOG

457.1 Description of table MESSAGELOG

The MESSAGELOG table is the collection of all current GLOBEMail messages. Rows are deleted from this table as schools delete their GLOBEMail.

A Trigger fires after Insert on this table to add a new row to ARCHIVEMESSAGELOG.

458 Table MICRO_CLIMATE

458.1 Description of table MICRO_CLIMATE

Mike and Sally (written 7/1/02),

The MICRO CLIMATE table was created a while back and didn't get documented as well as it should have. Here's what I remember. I'll put this text in the design document so we'll remember the original intent!

The intent is that the MICRO CLIMATE table be a base table, not a snapshot or materialized view. But the content of the table would be derived almost completely from other tables, primarily GDS MOISTCHG 30DAYS, and would get updated from that snapshot periodically, at least every several days. The intent is to keep a permanent record of GDS MoistureChg information only for schools that have active Phenology sites.

The column "RadiusKm" is intended as a variable that takes on only a very limited number of values. If RadiusKm is 0, then the

micro-climate information comes only from a single Atmosphere site, namely the one pointed to by the metadata for that phenology site. In that case one would not use GDS_MOISTCHG_30DAYS to populate MICRO_CLIMATE. If RadiusKm is equal to R (where R is an approximation of the radius of the rectangular region on the globe determined by rounding off latitude and longitude to 1 decimal digit), then MICRO_CLIMATE would be populated directly from GDS_MOISTCHG_30DAYS, since that snapshot calculates its values from all Atmosphere sites from any school that falls into the region determined by truncating (or rounding - I forget) latitude and longitude to 1 decimal digit.

Some things to consider

1) I defined the views and snapshots leading up to GDS_MOISTCHG_30DAYS some time ago, and those definitions should be reviewed to determine if they're still the best way to get this information. I know that WaterEquiv for Snow has changed since these definitions!

2) GDS_MOISTCHG_30DAYS should be sensitive to both Northern and Southern hemispheres, calculating info from Jan 1 to June 30 in the northern Hemisphere and from July 1 to Dec 30 in the Southern hemisphere.

3) GDS_MOISTCHG_30DAYS is sensitive to the time it takes to calculate the snapshot. I don't know how well it's been performing lately. Initially, it did very well for the first few months in a phenology season, but then took progressively longer as we neared the end of the season.

4) If MICRO_CLIMATE gets updated daily from GDS_MOISTCHG_30DAYS, then one may get some potentially duplicate information for MICRO_CLIMATE. This happens because "RecentDay" in GDS_MOISTCHG_30DAYS actually varies over 3-4 days, to account for schools that don't take atmosphere measurements every day. Thus it's possible that on day x you'll try to populate MICRO_CLIMATE with information that was already inserted on day x-1 or x-2. There are various ways to get around this problem, but I didn't think it through completely. One must be careful not to solve this problem in a way that misses good information that comes in a day or two later than expected!

5) Using the above approach, MICRO_CLIMATE is NOT sensitive to atmosphere data that gets updated more than 30 days after the measure day. Thus occasionally, one may want to re-calculate MICRO_CLIMATE from scratch! But that would require writing a modification to GDS_MOISTCHG_30DAYS that substitutes a specific Day for SYSDATE in the calculations of the views leading up to that snapshot, especially in PHN_JoinThree_VU. We may want to do this anyway in order to populate MICRO_CLIMATE with historical information.

NOTE There's no pride of ownership in the views and/or snapshots leading up to GDS_MOISTCHG_30DAYS. They were created initially as a quick-and-dirty way to get interesting GDS data for a larger number of Phenology sites. There may be better ways to do it now!

Sally -- my original thoughts in creating GDS_MOISTCHG_30DAYS was to provide data that could be used for a visualization (based solely on GLOBE data) that showed greening on the globe coming down from the North (or up from the South!). One could set the parameters for "greening" colors based on different values of GDS and then watch a visualization that showed the green creeping down (or up!) the globe over a 3 or 4 month period. I was hoping such a visualization could be used to get more schools interested in Phenology! Then MICRO_CLIMATE was created for each Phenology site, so that each site could anticipate (based on historial GDS compared to current GDS!) when it's various tree or shrub species would do their thing.

-- Len

459 Table NESTING_REPORT

460 Table NEW_PARTNER_CHECKLIST

461 Table NWS_MEASUREMENTS

461.1 Description of table NWS_MEASUREMENTS

The NWS_MEASUREMENTS table records daily Max and Min temperature and daily precipitation at more than 12000 weather sites around the world. See the descriptions of NWS_SITES for additional site information. No distinction is made between Precip_Rain and Precip_Solid as is done at GLOBE study sites.

461.2 Annotation for table NWS_MEASUREMENTS

The data types of this table are GLOBE data types, not necessarily NWS data types.

462 Table NWS_SITES

462.1 Description of table NWS_SITES

The NWS_SITES table records more than 12000 weather stations around the world. The table is maintained by NASA Goddard with input from various sources.

462.2 Annotation for table NWS_SITES

The data types of this table are GLOBE data types, not necessarily NWS data types.

The semantics of StationCode are not yet known (Aug 1999).

>I believe that the siteid and stationcodes are assigned for
>each station by the WMO (World Meteorological Organization). I'm
>sure they use some kind of logical scheme, but I don't know
>what it is. In general, the letter-identifiers (like KRIC for
>Richmond) are based either on the city or airport name. Some are
>obvious (like RIC for RIChmond), others are more convoluted
>(like IAD for Dulles Airport - maybe the IAD stands for
>"International Airport Dulles"; I'm not sure). Some are
>changed because of conflicts with existing names - for example
>Albany NY (ALB) and Albany GA (ABY).
>
>The leading letter (like K) I believe is based on the area, most
>US stations start with K for example.

463 Table ORDER_ITEMS

463.1 Description of table ORDER_ITEMS

The OrderItems table identifies the individual items in a GLOBE purchase order. Each item is subject to the delivery and/or shipping dates identified in the parent GlobeOrder table.

463.2 Annotation for table ORDER_ITEMS

Must define a TRIGGER so that when a Quantity is ordered for a workshop session the WarehouseQuantity of the product (or all products in a kit) is either increased or decreased in the GlobeProduct table. If the OrderType is Warehouse-to-Workshop, then the WarehouseQuantity is decreased; if the OrderType is Vendor-to-Warehouse, then the WarehouseQuantity is increased. If the product ordered is a kit made up of other items, e.g. Teacher's Guide Kit, then the WarehouseQuantity must be adjusted for every item in the kit.

The ItemStatus and StatusDate columns are under discussion for possible addition to this table. The fields would be left null on the initial order, and could be filled in later if the item is Back-Ordered, Not-Available, Partially-Filled, Received-in-Full, etc.

The value of ItemStatus would be over-written by each new entry, so no history would be kept in those fields. However, we could define a Trigger on the table so that whenever ItemStatus or StatusDate is Updated, the Trigger would Append the current values of both fields to the Comments.

464 Table OZONE

464.1 Description of table OZONE

The OZONE table records Ozone related measurements taken once daily at a specific Atmosphere study site. An Ozone measurement requires about one hour of elapsed time to complete and it is important to record other atmosphere conditions both at the start of the exposure time period and at the end of that time period. The two required Cloud Observations are stored in the CLOUD_OBS table under the primary key identifier determined by the SolarHour/Minute of the Start and the SolarHour/Minute of the Stop. The Solar times are derived from the UT start and stop times supplied by the school, with precision to the minute. The Ozone PPB is derived from the exposed film according to the OzoneMethod identified in the Teacher's Guide. At present there is only one method, identified by OzoneMethod=1.

465 Table OZONE_CNT

466 Table PALS_INFO

467 Table PALS_PROTOCOLS

468 Table PARTICLE96_WS

468.1 Description of table PARTICLE96_WS

The SandSiltClayWS table records raw data soil settling readings for multiple samples taken from a given Horizon at a given soil characteristics study site. The values reported for each sample are:

Volume of the sample (mL or cc's)
Settled after 40 seconds (mL)
Settled after 30 minutes (mL)

The sample data is used to derive the percent of sand, silt, and clay in the soil sample. The sample values are averaged and the average reported in the SandPct, SiltPct, and ClayPct attributes of the SoilData table.

469 Table PARTICLE96_WS_CNT

470 Table PARTICLE_WS

470.1 Description of table PARTICLE_WS

The Particle worksheet records specific gravity readings at different times for a soil and water mixture with mutiple soil samples taken from a given horizon of a given soil characterization study site.

Data entry fields are:

Weight of soil sample (grams)
Temperature of Soil and Water after mixed together in 500 mL cylinder.
Specific gravity (using hydrometer) at 30 seconds.
Specific gravity (using hydrometer) at 12 minutes.
Specific gravity (using hydrometer) at 47 minutes

The specific gravity readings are used to estimate the percentages of sand, silt, and clay in the soil sample. Note: NOT SURE THIS IS HOW USED! Other uses may be possible too!

471 Table PARTICLE_WS_CNT

472 Table PARTNER_GOALS

473 Table PARTNER_IMPLEMENTATION_PLAN

474 Table PARTNER_LIAISON

475 Table PARTNER_SCHOOLTYPES

475.1 Description of table PARTNER_SCHOOLTYPES

The PARTNER SCHOOLTYPES table identifies the school types within Training and Follow Up Target areas, that a U.S. Partner will focus on. The choices are taken from the SchoolType domain.

476 Table PARTNER_STRATEGIES

477 Table PARTNER_TO_AFFILIATE

478 Table PET_CHART

479 Table PHENGARDEN_CONTENT

480 Table PHENGARDEN_SITE

481 Table PHENOLOGICAL_GARDEN

482 Table PHENOLOGY_CNT

483 Table PHENOLOGY_SHRUB

484 Table PHENOLOGY_SITE

485 Table PHENOLOGY_SITE_CNT

486 Table PHENOLOGY_WORKUP

486.1 Description of table PHENOLOGY_WORKUP

The PHENOLOGY WORKUP table is a viewed table or a snapshot table of relevant quantities derived from the associated Air Temperature, Rain, and Snow atmosphere measurements, and the PET CHART table. It contains data for each calendar day from the beginning of the growing season (Jan 1 or July 1) up to the current day (i.e. the day the view or snapshot is executed). It is critically important that none of the days be missing for the 30-days preceding budburst. All values can be derived from Atmosphere measurements if the schools collect all required atmosphere measurements daily! A visualization of how the cumulative sums of GDS and PET change during the growing season may be very interesting for analysis.

A possible view definition is given in the Annotation section following.

486.2 Annotation for table PHENOLOGY_WORKUP

```
CREATE SNAPSHOT Phenology  Workup
REFRESH COMPLETE
START WITH '09-MAR-98' NEXT SYSDATE + 1
 AS
(SELECT
   Study  Site.SchoolId AS SchoolId,
   Study  Site.SiteId AS SiteId,
   Air  Temp.SolarDay AS SolarDay,
   (MaxTemp + MinTemp)/2 AS AvgTemp,
   GREATEST(0, (MaxTemp + MinTemp)/2) AS GDSvalue,
   (P1.PETvalue
           + (
              (MaxTemp + MinTemp)/2 - FLOOR( (MaxTemp + MinTemp)/2 )
            )
           * (P2.PETvalue - P1.PETvalue)
   ) AS PETvalue,
   Precip  Rain.RainAmount AS NewRain,
   Precip  Solid.NewAmount AS NewSnow,
   Precip  Solid.RainEquivalent AS RainEquiv,
   Precip  Solid.CurrentDepth AS Snowpack,
   Precip  Solid.CurrentDepth/5 AS WaterEquiv,
   (Precip  Rain.RainAmount
            + NVL(Precip  Solid.RainEquivalent, 0)
            - (
               P1.PETvalue
                  + (
                     (MaxTemp + MinTemp)/2 - FLOOR( (MaxTemp + MinTemp)/2 )
                   )
             )
            * (P2.PETvalue - P1.PETvalue)
   ) AS WaterDiff
 FROM Study  Site, Phenology  Site, Air  Temp, Precip  Rain, Precip  Solid,
        PET  Chart P1, PET  Chart P2
 WHERE
   Study  Site.InvestigCode = 'PHN'
  AND
   Study  Site.DateDeactivated IS NULL
  AND
   Study  Site.SchoolId = Air  Temp.SchoolId
  AND
   Air  Temp.SchoolId = Precip  Rain.SchoolId
  AND
   Air  Temp.SchoolId = Precip  Solid.SchoolId(+)
  AND
```

```
  Study  Site.SchoolId = Phenology  Site.SchoolId
 AND
  Air  Temp.SiteId = Phenology  Site.TempLink
 AND
  Phenology  Site.RainLink = Precip  Rain.SiteId
 AND
  NVL(Phenology  Site.SnowLink,'NULLL') = NVL(Precip  Solid.SiteId,'NULLL')
 AND
  VersionDate = (SELECT MAX(VersionDate)
              FROM Phenology  Site
              WHERE
                Study  Site.SchoolId = Phenology  Site.SchoolId
              AND
                Study  Site.SiteId = Phenology  Site.SiteId
             )
 AND
  Air  Temp.SolarDay >= VersionDate
 AND
  Air  Temp.SolarDay = Precip  Rain.SolarDay
 AND
  Air  Temp.SolarDay = Precip  Solid.SolarDay(+)
 AND
  TRUNC((MaxTemp + MinTemp)/2) = P1.AvgTemp
 AND
  TRUNC((MaxTemp + MinTemp)/2) + 1 = P2.AvgTemp
 AND
  (
    (
      TRUNC(Air  Temp.SolarDay + 183, 'YEAR') = TRUNC(SYSDATE, 'YEAR')
    AND
      TRUNC(SYSDATE + 183, 'YEAR') = TRUNC(SYSDATE, 'YEAR')
    )
   OR
    (
      TRUNC(Air  Temp.SolarDay - 183, 'YEAR') = TRUNC(SYSDATE, 'YEAR')
    AND
      TRUNC(SYSDATE - 183, 'YEAR') = TRUNC(SYSDATE, 'YEAR')
    )
  )
);
```

487 Table PHENOLOGY_WS

488 Table PHENOLOGY_WS_CNT

489 Table PHOTOMETER_CALIB

489.1 Description of table PHOTOMETER_CALIB

The PHOTOMETER CALIB table records the calibration of each photometer used by a GLOBE school to take Haze measurements. Every Haze voltage measurement recorded in the HAZE WS table identifies a specific PhotometerId and ChannelId, so that an appropriate ET constant can be used to help calculate the haze AOT. Sometimes the photometer channels need to be re-calibrated, so the initial calibration, or recalibration, Date is part of the primary key of this table. The Date of the measurement will fall between two CalibDate's, thereby identifying the appropriate ET constant to use, although sometimes this may be the later CalibDate rather than the earlier one. The procedure for which Date to use is not yet fully specified.

489.2 Annotation for table PHOTOMETER_CALIB

At 09:51 PM 4/18/00 -0400, David R. Brooks wrote:
> Calibration data for each instrument should include:
>1. the full model and serial number of the instrument, using the format I
>have devised. For example, instrument RG2-041 is a two-channel (red and
>green) instrument with a digital panel meter (the 2). The number (041 in
>this example) is assigned consecutively in a single series for all GLOBE
>sun photometers. That is, every instrument, regardless of its "model
>number" designation, has a unique serial number.
>2. The extraterrestrial constant, as determined by a Langley plot
>calibration or by transfer from a primary reference instrument. A
>two-channel instrument will have two such constants.
>3. The dark voltage for the instrument. Again, a two-channel instrument

>will have two such constants. (In our new PC board design, we are using a
>separate op-amp for each channel, so each channel is essentially completely
>independent from the other and can therefore produce a different dark
>voltage under calibration conditions.)
>4. The serial number(s) of the sun photometer(s) from which this instrument
>was calibrated. For reference instruments that have been calibrated at MLO
>or (perhaps in the future) some other location, we would enter a code such
>as "MLO" in place of a sun photometer serial number. It is possible that
>each channel of a two-channel instrument will be calibrated from a
>different source. I have done transfer calibrations for a two-channel
>instrument using one sun photometer for the green channel and a different
>one for the red channel calibration.
>5. The date(s) of the most recent calibration(s). Again, these dates might
>conceivably be different for each channel of a two-channel instrument. For
>reference instruments, Forrest and I will need to keep a record of all
>calibrations, but I guess this doesn't need to be part of the GLOBE data
>base of operational instruments used in GLOBE schools.
>
>Of all these values, only the extraterrestrial constant is required to do
>the AOT calculations. That is, this value needs to be stored in a "lookup
>table" that can be accessed quickly when students report measurements,
>including the serial number of the instrument they are using. The other
>values, however, should be part of the permanent documentation record for
>each instrument.

490 Table PHOTO_ENVELOPE

491 Table PHOTO_INFO

492 Table POST_CHECKLIST

492.1 Description of table POST_CHECKLIST

The PostWkshpChecklist table identifies a list of action items that must be completed after each workshop session is completed. The action items may differ for each workshop type (i.e. TRAD, FIT, TTT, INT) and new action items may be created at any time by the administrators of each workshop type.

To create an action item it is only necessary to insert a new row into this table with a different value for the PostActionItem column.

Initially, the post-workshop action items are:

Evaluation - Copies of all participant and trainer evaluation forms.

Inventory - A post-workshop accounting of all materials and equipment that are remaining after the workshop has taken place.

Report - A brief summary of the workshop, lessons learned, etc.

Roster - A final attendance roster of all participants who attended the workshop.

493 Table PRECIP_RAIN

493.1 Description of table PRECIP_RAIN

The PrecipRain table records daily rain amounts and rain acidity for rain water collected in the rain gauge at a given atmosphere study site.

The data fields are:

Rain amount (mm)
Trace or Missing flag (T or M)
Days accumulated
Acidity (pH)
Acidity Method used

494 Table PRECIP_RAIN_CNT

495 Table PRECIP_SOLID

495.1 Description of table PRECIP_SOLID

The PrecipSolid table records daily solid precipitation amounts (snow, sleet, etc) and the acidity of the sample taken from the snow board at a given atmosphere study site.

The data fields are:

New amount (mm)
Daily Trace or Missing flag (T or M)
Days accumulated
Current total depth (mm)
Total depth Trace or Missing Flag (T or M)
Rain equivalent (mm)
Acidity (pH)
Acidity Method used (choice of 3)

496 Table PRECIP_SOLID_CNT

497 Table PRECIP_SOLID_WS

498 Table PRE_CHECKLIST

498.1 Description of table PRE_CHECKLIST

The PreWkshpChecklist table identifies a list of action items that must be completed prior to each workshop session. The action items may differ for each workshop type (i.e. TRAD, FIT, TTT, INT) and new action items may be created at any time by the administrators of each workshop type.

To create an action item it is only necessary to insert a new row into this table with a different value for the PreActionItem column.

Initially, the pre-workshop action items are:

Contact Information - All relevant contact information for the site. Corresponds with contact codes.

Invitation Letter - The letter sent from the site to the teacher who will be attending training.

Agenda - The agenda that will be followed during the workshop session.

499 Table PROTOCOL_TO_CONTACT

500 Table PROTOCOL_USES_GROUPS

501 Table REGISTRATION

501.1 Description of table REGISTRATION

The Registration table maintains a many-to-many relationship among teachers and teacher training workshop sessions.

A registration StatusCode identifies the status of the relationship between a given teacher and a given workshop session. Current codes are:

P1 First Preference Workshop Session
P2 Second Preference Workshop Session
P3 Third Preference Workshop Session
TR Tentative Registration Received
RF Registration Meal Fee Received
RC Registration Cancelled
NS No Show
AW Attended Workshop - Then Withdrew
CW Completed Workshop

502 Table REGISTRATION_FOR_PROTOCOL

503 Table S2S_ACTIVITY

504 Table S2S_PARTICIPANT

505 Table SALINITY_LOOKUP

505.1 Description of table SALINITY_LOOKUP

The SALINITY_LOOKUP table gives a relationship among water temperature, specific gravity, and salinity. The values contained in the table are equivalent to Table HYD-P-3 presented in the GLOBE 1997 Teacher's Guide.

505.2 Annotation for table SALINITY_LOOKUP

We need a text file of the content of Table HYD-P-3 from the GLOBE 1997 Teacher's Guide in order to populate this table.

506 Table SCHLTCHR_MINCWDATE

506.1 Description of table SCHLTCHR_MINCWDATE

A listing of SchoolId's and TeacherId's, for teachers currently associated with the school, along with the DATE at which the teacher first completed Globe-Training.

507 Table SCHOOL7_A

507.1 Description of table SCHOOL7_A

A view of 7 identifying attributes from the Globe School base table, restricted to schools having Life Cycle Code A as "Active".

508 Table SCHOOL7_CGTS

508.1 Description of table SCHOOL7_CGTS

A view of the Globe School base table, restricted to schools that have a Current, Globe-Trained Staff member.

509 Table SCHOOL7_CP

509.1 Description of table SCHOOL7_CP

A view of 7 identifying attributes from the Globe School base table, restricted to schools having Life Cycle Codes C or P as "Candidate" or "Pre-candidate", respectively.

510 Table SCHOOL7_IO

510.1 Description of table SCHOOL7_IO

A view of 7 identifying attributes from the Globe School base table, restricted to schools having Life Cycle Codes I or O as "Inactive" or "Other", respectively.

511 Table SCHOOL7_M

511.1 Description of table SCHOOL7_M

A view of 7 identifying attributes from the Globe School base table, restricted to schools having Life Cycle Code M as "adMinistrative".

512 Table SCHOOL7_N_CGTS

512.1 Description of table SCHOOL7_N_CGTS

A view of the Globe School base table, restricted to schools that do NOT have a Current, Globe-Trained Staff member.

513 Table SCHOOLFLAGS

514 Table SCHOOLGRADE_CODES

514.1 Description of table SCHOOLGRADE_CODES

The SchoolGradeCodes entity describes codes used to identify the grade level and approximate ages of students at a given GLOBE School. It is likely that students will use these values to help locate other schools that are "like" themselves, thereby fostering school-to-school communication and possible relationships as "school partners".

SchoolGradeCodes entity
GradeCode (primary key)
GradeName
Description
Description-French
Description-Russian
Description-Arabic
Description-Chinese

The GradeCode attribute is a specific 1-character code; it is the primary key of this entity.

The GradeName attribute identifies the name of a given GradeCode; the list of allowable codes and names, with a brief description, is as follows:

P Primary Grades K-3, approximate ages 5-10
M Middle Grades 4-6, approximate ages 9-13
U Upper Grades 7-9, approximate ages 12-16
H High Grades 10-13, approximate ages 15-19
C College/University, approximates ages 18+

The Description attribute contains the descriptive English language text for each GradeCode; the other associated Description attributes give school grade code descriptions in other GLOBE supported languages, if available.

514.2 Annotation for table SCHOOLGRADE_CODES

These codes and names are not aligned with any agreed international categorization of school grades; if a suitable categorization exits, then it should be used instead.

515 Table SCHOOLZOOM

516 Table SCHOOL_CONGRESS

516.1 Description of table SCHOOL_CONGRESS

The SchoolCongress table maintains a many-to-many relationship among GLOBE Schools and members of the U.S. Congress. The criteria for inclusion in this relationship is that a GLOBE school have students from the home District of a House member.

517 Table SCHOOL_EQUIP

517.1 Description of table SCHOOL_EQUIP

The SchoolEquipment table identifies scientific equipment and computer hardware and software that is supplied by GLOBE to various schools. It identifies the equipment by an EquipCode attribute (that references the GlobeProduct table?) and tracks the date that the identified item was sent to the associated GLOBE school.

518 Table SCHOOL_GRANTINFO

518.1 Description of table SCHOOL_GRANTINFO

The SchoolGrantInfo table records the GrantNumber, GrantStatus, and GrantScore for any U.S. school that applied for and received a grant in the early years of the GLOBE program. After 1997 grants were no longer available to any school; however, some schools have not yet invoked their grant privileges, so the GrantStatus attribute identifies those schools that can still use grant money for GLOBE teacher training.

519 Table SCHOOL_HTTPINFO

520 Table SCHOOL_MINCWDATE

520.1 Description of table SCHOOL_MINCWDATE

A listing of SchoolId's along with the DATE at which a current staff member at that school first completed Globe-Training.

521 Table SCHOOL_NBRCGTS

521.1 Description of table SCHOOL_NBRCGTS

A listing of SchoolId's along with the Number of Current, Globe-Trained Staff associated with that school.

522 Table SCHOOL_TRAINEDSTAFF

522.1 Description of table SCHOOL_TRAINEDSTAFF

A join of the Registration and TeacherToSchool base tables, restricted to current teachers at the identified school who have completed GLOBE Training.

523 Table SCHOOL_ZOOMLEVEL

523.1 Description of table SCHOOL_ZOOMLEVEL

The SCHOOL_ZOOMLEVEL table assigns a "zoom level" to each GLOBE school. This identifies which map or fixed school list for a region the school belongs on. This table may be replaced in Spring 2001 by a more universal mechanism.

524 Table SESSION_PROTOCOLS

525 Table SESSION_TO_PARTNER

526 Table SITE_CODES

526.1 Description of table SITE_CODES

The SiteCodes entity defines a 3-character code to represent GLOBE sample or study sites. The relationship among Globe Investigations, Globe Sites, and Globe Measurements is given in the Investigation_Tree table.

The GLOBE Site Codes are listed in the SiteCodes domain.

A special SiteCode is used to identify the Main Entrace of a GLOBE School as described by the GPS Investigation protocol of the GLOBE Teachers Guide. The special site code for the main entrance is SCH and the measurement code for that measurement is FD for Front Door. All GPS readings captured by the School Location Data Entry Sheet will be stored in the Site_Location table with SiteId = 'SCH-01', and the Data_Activity table will record that action with siteId = 'SCH-01' and with MeasureCode = 'FD'."

527 Table SITE_LOCATION

527.1 Description of table SITE_LOCATION

The SiteLocation entity records GPS locations, when they are known, for GLOBE Study Sites. In some cases these GPS values are determined at the time of Study Site creation and in some cases they are determined by later GPS measurements. If Study Sites are not in an open area, then GPS locations can be derived by using the "Offset GPS Measurements" protocol defined in the GLOBE Teacher's Guide. If GPS receivers are not available, then the Latitude, Longitude, and Elevation of a Study Site can be estimated by other means, but it is desirable that these measurements be accurate within 0.01 arc minute, which translates to about 20 meters on the Earth's surface.

```
SiteLocation entity
  SchoolId      (PK, FK)
  SiteId        (PK, FK)
  Latitude
  Longitude
  Elevation
```

The 2-part combined primary and foreign keys ensure that there is at most one GPS position recorded in the Globe database for each

GLOBE Study Site.

The foreign key FK references the StudySite entity, and establishes a one-to-one mapping from the SiteLocation entity into the StudySite entity. There is no additional constraint to require that every study site has a GPS location; we could add such a constraint if that becomes a GLOBE requirement. With the FK constraint only, it is still possible for a study site to be represented in the StudySite entity without having a GPS location for it in the SiteLocation entity.

Standard representations of Latitude, Longitude, and Elevation are defined in International Standard ISO 6709:1993. We define positive and negative values for Latitude and Longitude consistent with that standard. The Elevation attribute identifies the relative elevation of a study site (measured in meters) either above (+ values) or below (- values) mean sea level.

527.2 Annotation for table SITE_LOCATION

This entity does not have a DatePerformed or DateRecorded attribute; if it is important to maintain that information, then these attributes could be added here, or a new SiteActivity entity (analogous to the DataActivity entity under Student Measurements) could be created just for this purpose.

A common application of this entity will be for placement of student measurements at a specific location on the world globe. For example, by joining this entity with the Air Temperature entity (joining on SchoolId and SiteId) one is able to associate temperature measurements with specific Latitude and Longitude readings.

This SiteLocation entity will be populated in the same manner that existing NOAA/FSL tables with SITELOCATION information are populated. The GLOBE Server will read data entry sheets that have GPS information and Insert a new instance into this entity, Update an existing instance (e.g. with a more precise GPS value), or report an error, as appropriate.

528 Table SITE_LOCATION_CNT

529 Table SITE_OBSTRUCTION

530 Table SITE_PHOTO

531 Table SITE_PHOTO_CNT

532 Table SITE_PROXIMITY

532.1 Description of table SITE_PROXIMITY

The SiteProximity entity maintains a many-to-many relationship among "near by" study sites that are within the same parent GLOBE School study site. If two study sites are in close proximity to one another, and if the location of one site can be clearly identified relative to the known location of the other site, then a proximity relationship from the known site to the related site can be maintained by the SiteProximity entity.

One of the goals of this entity is to provide a database structure for determining whether or not the following criteria for Selecting Your GLOBE School Study Sites, identified in pages 1-11 to 1-13 of the GLOBE II Teacher's Guide, are satisfied:

1) The Atmosphere site should be within 100 m of the Soil Moisture site.

2) One Soil Characteristics site should be within the Biology site.

3) One Soil Characteristics site should be located near to the Soil Moisture site.

4) One Soil Moisture site should be within 100m of a rain gauge Atmosphere site.

5) One Soil Moisture site should be within the Biology site.

6) One Soil Moisture site should be near to a Surface Water site.

7) The Biology site should ideally be at the center of a Land Cover site.

Another goal of this entity is to encourage the participation of mathematics classes in GLOBE School measurement activities.

Associated learning activities might include introduction to surveying and mapping techniques and use of engineering measurement tools such as rod, chain, level, and transit. Students in advanced classes (i.e. high school level) are encouraged to use geometric and trigonometric methods (see Working With Angles on pages 6-30 to 6-35 of the GLOBE II Teacher's guide) to map these relative location attributes to Latitude, Longitude, and Elevation attributes for the related site. A protocol for determining Offset GPS Measurements from a given GPS location using distance and direction "offsets" is given on pages 6-36 to 6-38 in the GLOBE II

Teacher's guide. The derived geographic location values can then be compared with GPS measured values and the reasons for any differences analyzed.

Since GPS measurements are often only accurate within 15 to 30 meters, these local proximity measurements may sometimes be more valuable in studying intersite relationships than are the GPS values.

```
SiteProximity entity
  SchoolId          (PK, FK1, FK2)
  KnownSiteId         (PK, FK1
  RelatedSiteId       (PK,     FK2)
  Distance
  Direction
  ElevationChange
```

The three-part primary key ensures that there will be at most two sets of proximity measurements for any pair of local study sites. For example, measurements may be recorded for SiteB related to a known SiteA, or for SiteA related to a known SiteB. Database constraints will be used to ensure that if both proximity relationships are stored in the database, then their relative location values will be consistent. Potential inconsistencies can be avoided by up-front agreement on which sites should always be considered "known" and which sites should be "related"; for example, Soil Moisture sites could always be regarded as related to the known Atmosphere, Biology, Soil Characteristics, or Surface Water study sites that they are in close proximity to.

The foreign key FK1 references the StudySite entity, thereby enforcing a one-to-many mapping from the StudySite entity to this SiteProximity entity; indirectly, this provides a mapping from a known study site to a collection of related sites in the same "proximity".

The foreign key FK2 references the StudySite entity, thereby enforcing a one-to-many mapping from the StudySite entity to this SiteProximity entity; indirectly, this provides a mapping from a related study site to a collection of known sites in the same "proximity".

In addition, the site proximity relationships determined by the primary and foreign keys form a directed graph over related study sites that can be traversed to determine and/or analyze recursive site dependencies. This might be exploited in a structured way to maintain a hierarchy of links among all study sites related to a given known site. For example, starting with a known Biology study site, this entity might provide links to related Surface Water and Soil Characteristics study sites, which in turn might provide links to related Soil Moisture study sites.

The Distance attribute records the distance (in meters) measured along a straight line between the centers of the known site and the related site. The center of a site is the point where a GPS reading for the site would be taken; usually the geometric center of polygon shaped study sites.

The Direction attribute records a compass reading (in degrees) from the known site to the related site, measured clockwise relative to Magnetic North.

The ElevationChange attribute records the vertical change in elevation (in meters) from the known site to the related site.

532.2 Annotation for table SITE_PROXIMITY

The StudySite entity has DateActivated and DateDeactivated attributes. Thus every Site-Site pairing in this entity has these time attributes associated with it, thereby defining the valid time of the Site-Site proximity relationship. No further time duration attributes should be needed.

The SiteProximity entity is most directly related to the Soil Characterization Study Site Data Entry Sheet (on page 5-85 of the GLOBE II Teacher's Guide) and to the Soil Moisture Study Site Data Entry Sheet (on page 5-86 of the Teacher's guide). The Soil Moisture sheet asks for the distance and relative direction of the site to the raingauge and to the soil characterization hole. The Soil Characterization sheet asks for the user to check any of the choices from: Near SM site, Near SW site, In or near Biology site, or Other. In each case, with additional optional measurements for elevation change and distance, the WWW Server could record information from these data entry sheets in the database via the SiteProximity entity.

533 Table SITE_PROXIMITY_CNT

534 Table SKY_OBSCURED

534.1 Description of table SKY_OBSCURED

The SKY OBSCURED table records the reason that a Cloud Observation could not be taken because the sky was obscured. There could be multiple reasons and each possible reason is checked.

535 Table SKY_OBSCURED_CNT

536 Table SMGRAV_WS

536.1 Description of table SMGRAV_WS

The SMGravWS table records raw data used to determine the water content of soil. The four methods used are described under the SoilMoisture table.

The Protocol attribute identifies the protocol used (Star, Transect, Hole)

The Depth attribute (for the Star or Transect protocols) is either 5cm or 10cm; the Depth for the Hole protocol is either: 10cm, 30cm, 60cm, or 90cm.

The Position attribute (for the Star or Transect protocols) identifies the the "position" from which the soil sample was taken; the Position is always 1 for the Hole protocol.

The CanNumber attribute identifies a specific sample; it is analogous to the SampleNbr in several other student data worksheet tables.

The other data fields are:

Total wet weight of soil and can (grams)
Dry weight of soil and can (grams)
Weight of can (grams)
Drying time (minutes)
Drying method (several choices)
Gravimetric soil moisture content (dimensionless ratio derived from above values)

537 Table SMGRAV_WS_CNT

538 Table SMGYPS_WS

538.1 Description of table SMGYPS_WS

The SMGypsWS table records raw data used to determine the water content of soil using the Gypsum Blocks protocol. Other protocols for determining the water content of soil are described under the SoilMoisture table.

The data fields are:

Depth (10cm, 30cm, 60cm, 90cm)
Meter reading (between 0 and 100)
Date of curve used to calibrate meter reading (Installation date of gypsum blocks)
The calculated soil water content (dimensionless ratio between 0 and 100).

539 Table SMGYPS_WS_CNT

540 Table SNAP$_PHENOLOGY_WORKUP

541 Table SNOWPACK_WATER

542 Table SOILCHAR_SITE

542.1 Description of table SOILCHAR_SITE

The SoilCharSite entity records characteristics of a Soil Characteristics study site that are constant throughout the lifetime of that site. Some soil characteristics may change over time and are recorded in the SoilCharData entity, whereas others are stable for the lifetime of the study site and are recorded in this entity. If any of these stable characteristics should change significantly, then the site itself should be deactivated and no further measurements made because the data would no longer be interpreted under the same site characteristics. A new site could then be created with new stable site characteristics identified.

SoilCharSite entity
SchoolId (PK, FK1)
SiteId (PK, FK1)
SlopeOfSite
SampleSource
ParentMaterial
Comments

The primary and foreign key pair, ensures that there is at most one set of site characteristics recorded in the Globe database for each Soil Characteristics study site. Site characteristics do not vary over time.

The foreign key FK1 references the StudySite entity, and enforces a one-to-one mapping from this SoilCharSite entity into the StudySite entity.

The SlopeOfSite attribute records the general slope (measured in degrees) of the terrain surrounding a Soil Characteristics study site; it is presented in the student data entry sheet for soil characterizations on page 5-83 of the GLOBE II Teacher's Guide.

The SampleSource attribute identifies the type of collection method used to collect soil samples. It is specified on page 5-85 of the GLOBE II Teacher's Guide to be one of: Soil Pit, Auger Hole, Surface (<10 cm), Excavation, or Other.

The ParentMaterial attribute identifies the parent material of the soil samples. It is specified on page 5-85 of the GLOBE II Teacher's Guide to be one of: Bedrock, Glacial Deposit, Volcanic Deposit, Stream Deposit, Wind Blown Sand, Ancient Lake Deposit, Other, or Don't Know.

The Comments attribute allows further textual explantion of site characteristics that may influence student measurements.

Other site characteristics that could have an effect on soil characteristics measurements (e.g. undisturbed soil vs landfill) could be accommodated by adding appropriate new attributes to this entity.

542.2 Annotation for table SOILCHAR_SITE

The foreign key FK1 references the StudySite entity, and enforces a one-to-one mapping from this SoilCharSite entity into the StudySite entity. An additional constraint is needed to ensure that the referenced site is in fact a Soil Characteristics study site, i.e. a site whose MeasurementCode attribute is equal to SC. There are various ways to enforce this additional constraint in the Physical Database Design.

The foreign key FK1 does not require that every Soil Characteristics study site have a set of site characteristics associated with it; in many cases a Soil Characteristics study site may be created and recorded in the StudySite entity first, with the site characteristics measured and recorded in this entity later, possibly never. If it is a GLOBE requirement that every Soil Characteristics study site have the attributes of this entity always recorded in the database, then an additional constraint to enforce this rule is needed.

This entity does not have a DatePerformed or DateRecorded attribute; if it is important to maintain that information, then these attributes could be added here, or a new SiteActivity entity, analogous to the DataActivity entity discussed under the Student Measurements information category, could be defined.

The Soil Characterization Study Site Data Entry Sheet (page 5-85 of the Teacher's Guide) asks for the user to check any of the choices from: Near SM site, Near SW site, In or near Biology site, or Other. Because there may be multiple study sites that are effected by these questions, and because relative site location is also important, these characteristics are now recorded in the SiteProximity entity described below. Some modification of the student data entry sheets will be needed in order to capture this additional information.

This entity is most directly related to the existing NOAA/FSL GLOBE2SCSITELOCATION table.

543 Table SOILCHAR_SITE_CNT

544 Table SOILMOISTURE_SITE

544.1 Description of table SOILMOISTURE_SITE

The SoilMoistureSite entity records Soil Moisture study site characteristics that are constant throughout the lifetime of that site. If any of these characteristics should change significantly, then the site itself should be deactivated and no further measurements made because the data would no longer be interpreted under the same site characteristics. A new site could then be created with new site characteristics identified.

```
SoilMoistureSite entity
  SchoolId          (PK, FK1)
  SiteId            (PK, FK1)
  SurfaceCondition
  SurfaceCover
  CanopyCover
  PercentSand
  PercentSilt
  PercentClay
  RockEstimation
  RootEstimation
```

Comments

The primary and foreign key pair, ensures that there is at most one set of site characteristics recorded in the Globe database for each Soil Moisture study site. Site characteristics do not vary over time.

The foreign key FK1 references the StudySite entity, and enforces a one-to-one mapping from this SoilMoistureSite entity into the StudySite entity.

The SurfaceCondition attribute records the type of ground surface at the study site. It is specified on page 5-86 of the GLOBE II Teacher's Guide to be one of: Natural, Plowed, Graded, Backfill, Compacted, or Other.

The SurfaceCover attribute records the surface cover of the study site. It is specified on page 5-86 of the GLOBE II Teacher's Guide to be one of: Base Soil, Short Grass (< 10 cm), or Long Grass (> 10 cm). This differs slightly from the SurfaceCover attribute of the AtmosphereSite entity. They should probably be reconciled.

The CanopyCover attribute records the tree canopy cover of the study site. It is specified on page 5-86 of the GLOBE II Teacher's Guide to be one of: Open, Some Trees Within 30m, or Canopy Overhead.

The PercentSand, PercentSilt, and PercentClay attributes record estimates for the percentage of these soils in the area of the soil moisture measurements. They are specified on page 5-86 of the Teacher's guide.

The RockEstimation attribute gives a loose estimate of the number of rocks present in the immediate vicinity of the study site. It is specified on page 5-86 of the GLOBE II Teacher's Guide to be one of: None, Few, Many.

The RootEstimation attribute gives a loose estimate of the number of tree roots present in the immediate vicinity of the study site. It is specified on page 5-86 of the GLOBE II Teacher's Guide to be one of: None, Few, Many.

The Comments attribute allows further textual explantion of site characteristics that may influence student measurements.

Other site characteristics that could have an effect on soil moisture readings could be accommodated by adding appropriate new attributes to this entity.

544.2 Annotation for table SOILMOISTURE_SITE

The foreign key FK1 references the StudySite entity, and enforces a one-to-one mapping from this SoilMoistureSite entity into the StudySite entity. An additional constraint is needed to ensure that the referenced site is in fact a Soil Moisture study site, i.e. a site whose MeasurementCode attribute is equal to SM. There are various ways to enforce this additional constraint in the Physical Database Design.

The foreign key FK1 does not require that every Soil Moisture study site have a set of site characteristics associated with it; in many cases a Soil Moisture study site may be created and recorded in the StudySite entity first, with the site characteristics measured and recorded in this entity later, possibly never. If it is a GLOBE requirement that every Soil Moisture study site have the attributes of this entity always recorded in the database, then an additional constraint to enforce this rule is needed.

The SurfaceCover attribute records the surface cover of the study site. It is specified on page 5-86 of the GLOBE II Teacher's Guide to be one of: Base Soil, Short Grass (< 10 cm), or Long Grass (> 10 cm). This differs slightly from the SurfaceCover attribute of the AtmosphereSite entity. They should probably be reconciled.

This entity does not have a DatePerformed or DateRecorded attribute; if it is important to maintain that information, then these attributes could be added here, or a new SiteActivity entity, analogous to the DataActivity entity discussed under the Student Measurements information category, could be defined.

The Soil Moisture Study Site Data Entry Sheet (page 5-86 of the Teacher's guide) asks for the distance and relative direction of the site to the raingauge and to the soil characterization hole. The relative direction choices are one of: N, NE, E, SE, S, SW, W, NW. Because there may be multiple study sites that are effected by these questions, and because relative site location is also important, these characteristics are now recorded in the SiteProximity entity described below. Some modification of the student data entry sheets will be needed in order to capture this additional information.

The Soil Moisture Study Site Data Entry Sheet (page 5-87 of the Teacher's Guide) asks for the user to identify the most detailed MUC level for this study site. This information is now recorded in the LandCoverData entity described under the Student Measurements information category.

This entity is most directly related to the existing NOAA/FSL GLOBE2SMSITELOCATION table.

545 Table SOILMOISTURE_SITE_CNT

546 Table SOILTEMP_DAILYINFO

546.1 Description of table SOILTEMP_DAILYINFO

The SoilTemp table records daily soil temperature measurements for 5cm and 10 cm near-surface depths at a given Surface Moisture (SM) or Atmosphere (AT) Study Site.

The data fields are:

Hour measured (UT hour)
Depth of sample (5cm or 10 cm)
Current temperature (degrees celsius)
Thermometer type (several choices)

546.2 Annotation for table SOILTEMP_DAILYINFO

The foreign key FK1 references the StudySite entity, and enforces a one-to-many mapping from a study site to a collection of soil temperature measurements taken at that site on different days. An additional constraint is needed to ensure that the referenced site is in fact a site created for Soil Temperature measurements, i.e. a study site whose MeasurementCode attribute is either AM, AT, or SM. There are various ways to enforce this additional constraint in the Physical Database Design.

547 Table SOIL_CHAR

547.1 Description of table SOIL_CHAR

The SoilData table records soil characteristics for each soil Horizon at a given soil characteristics study site.

The data fields are:

Top depth (cm).
Bottom depth (cm).
Moisture estimate (3 choices)
Water content (calculated from worksheet table)
Structure estimate (several choices)
Consistence estimate (several choices)
Dominant color code (from list of Munsell colors)
CoDominant color code (from list of Munsell colors)
Texture estimate (several choices)
Rock estimate (3 choices)
Rock volume (calculated from worksheet table)
Root estimate (3 choices)
Free carbonates (3 choices)
Nitrogen content (derived from worksheet table)
Phosphorous content (derived from worksheet table)
Potassium content (derived from worksheet table)
Sand percent (derived from worksheet table)
Silt percent (derived from worksheet table)
Clay percent (derived from worksheet table)
Acidity (derived from worksheet table)

548 Table SOIL_CHAR_CNT

549 Table SOIL_INFILTRATION

549.1 Description of table SOIL_INFILTRATION

The SoilInfiltration table records the rate at which water is absorbed into saturated soil (mm/minute) at a given soil or surface water study site. Infiltration rate is determined in a controlled experiment by multiple measurements of water absorbtion from two concentric cylinders. Water in the outer cylinder helps to saturate the soil and provides experimental control. Absorption of water from the inner cylinder simulates the rate at which water is absorbed into the soil.

There are three flow regimes to consider:

Unsaturated flow - the initial flow rate is high as the dry soil pores fill with water.

Saturated flow: - the flow rate is steady as water moves into the soil at a rate determined by soil texture and structure.

Ponding - the flow rate approaches zero when the ground becomes totally saturated
and is no longer able to conduct water through its pores.

The saturated flow rate recorded in this table is derived from multiple measurements over time of flow rates through the three regimes described above. The raw flow rate data is stored in the InfiltrationWS table.

549.2 Annotation for table SOIL_INFILTRATION

The foreign key FK1 references the StudySite entity, and enforces a one-to-many mapping from a study site to a collection of Soil Infiltration measurements taken at that site on different days. An additional constraint is needed to ensure that the referenced site is in fact a site created for Soil Infiltration measurements, i.e. a study site whose MeasurementCode attribute is either SM or SC. There are various ways to enforce this additional constraint in the Physical Database Design.

550 Table SOIL_INFILTRATION_CNT

551 Table SOIL_MOISTURE

551.1 Description of table SOIL_MOISTURE

The SoilMoisture table records the water content of the soil for different depths at a given soil study site. Water content is a dimensionless ratio (between 0 and 70) that is the ratio of: (weight of water in soil)/(weight of dry soil).

The soil moisture can be estimated using one of three Gravimetric protocols:

Star protocol (Globe II Teacher's Guide page 5-34)
Transect protocol (Globe II Teacher's Guide page 5-35)
Hole protocol (Globe II Teacher's Guide page 5-35)

or by the Gypsum Block protocol (Globe II Teachers Guide pages 5-38 to 5-41).

The data fields are:

Moisture conditions estimate (Dry, Moist, Wet) - this is daily metadata
Water content of soil (dimensionless ratio derived from a worksheet table)
Protocol used (Star, Transect, or Hole Gravimetric, or Calibrated Gypsum Block)

552 Table SOIL_MOISTURE_CNT

552.1 Annotation for table SOIL_MOISTURE_CNT

This table is not included in any DixonCnt calculations, because the two worksheet CNT tables do all of the counts that are needed.

553 Table SOIL_TEMP

554 Table SOIL_TEMP_CNT

555 Table STAFF_PROTOCOLS

556 Table STATE_LEA_CODE

556.1 Description of table STATE_LEA_CODE

The STATE LEA CODES record the Local Education Administration unit in each state. An LEA can be a School District, a ReligiousOrganization, a Charter School Board, etc.

557 Table STATE_PROV

557.1 Description of table STATE_PROV

The Regions table identifies a State, Province, Region, Territory, or other political subdivision within a GLOBE country, specifies a code to represent that region, and maps that code to the proper spelling of the region name in English and in other languages supported by GLOBE.

A two-part primary key maintains a one-to-many mapping from Countries to Regions, so RegionCodes need only be unique within an individual country.

In the United States, RegionCode is the 2-digit state code or territory code. In other countries a RegionCode, if appropriate, may be chosen in consultation with the country coordinator for that GLOBE country. In some countries a RegionCode is meaningless; in those situations a RegionCode attribute can always be left unspecified in the database.

Representation of a RegionCode attribute in the database may be with a datatype having more than two characters in order to accommodate longer codes from some countries.

The RegionType attribute gives the name of the type of political subdivision for that region of the identified country, e.g.: State, Province, Territory, District, Protectorate, etc.

The RegionName attribute gives the correct English spelling of the identified region within a GLOBE country. The other associated RegionName attributes give the correct spelling of the region name in other languages supported by GLOBE.

558 Table STATE_PROV_REGION

559 Table STRATEGY_ITEM

560 Table STUDY_SITE

560.1 Description of table STUDY_SITE

The StudySite entity identifies and categorizes study sites at a given GLOBE School. A GLOBE Teacher at a given school will create and name each of these study sites, and indicate the SiteCode's for which student measurements will be conducted.

```
StudySite entity
  SchoolId        (PK, FK)
  SiteId          (PK     )
  SiteName
  SiteCode
  DateActivated
  DateDeactivated
  Comments
```

The two-part primary key maintains a one-to-many relationship from Globe Schools to Study Sites defined at that school; it allows SiteId's to be assigned uniquely within a GLOBE School, rather than globally. The foreign key FK references the GlobeSchool entity and enforces a one-to-many mapping from the GlobeSchool entity to the StudySite entity.

The SiteCode attribute identifies the GLOBE investigation under which this study site is established. The SiteCode's are defined by the SiteCode Domain. The DateActivated and DateDeactivated attributes identify the time period during which the static characteristics of a site and the student measurements taken at a site are valid. For example, if a Surface Water study site is significantly altered or destroyed by urban construction, then that site should be deactivated (but kept in the database) and no further student data measurements accepted for it. The Comments attribute provides a mechanism and an opportunity for GLOBE Teachers to explain to the world the reasons why this site was selected for student measurements.

560.2 Annotation for table STUDY_SITE

SiteId could be a pure number, rarely (or never) seen by end users, or it could have some semantic content embedded into it, like the MeasurementCode, to make it more meaningful when it appears as a foreign key in other entities. This choice is an emotional issue, with pros and cons, that we should not let consume us; either approach works.

This StudySite entity could be combined with the SiteLocation entity, which is described separately. They are kept separate in order to encourage new study site identification even long before GPS coordinates are known. For example, a teacher could look at the Lansat Thematic Mapper (TM) scene of the 15km x 15km GLOBE School Study Site and identify Land Cover Study Sites to be gradually studied over the next few years. If a DateActivated is in the future, it could be recognized as a notice of future intent rather than past fact.

Backward compatibility with the existing GLOBE database is possible (but likely non-trivial) by mapping the existing EXPERIMENTERID, SITENAME, and/or SITENUMBER in many NOAA/FSL tables to SchoolId, SiteName, and SiteId, respectively. Minor modifications to existing new study site creation sheets in the GLOBE II Teacher's Guide could make them suitable to use, along with other information known by the Globe Server, for populating this new entity. In addition, whenever a GLOBE School Coordinator for a GLOBE School connects to the GLOBE WWW Server, they could be offered an option to "Register New Study Sites" at that school. Under certain circumstances, this study site creation privilege may be extended to any GLOBE Teacher at a given GLOBE School.

561 Table SURFACEWATER_SITE

561.1 Description of table SURFACEWATER_SITE

The SurfaceWaterSite entity records SurfaceWater study site characteristics that are constant throughout the lifetime of that site. If any of these characteristics should change significantly, then the site itself should be deactivated and no further measurements made because the data would no longer be interpreted under the same site characteristics. A new site could then be created with new site characteristics identified.

SurfaceWaterSite entity
- SchoolId (PK, FK1)
- SiteId (PK, FK1)
- MovingWaterType
- MovingWaterWidth
- StandingWaterType
- StandingWaterArea
- StandingWaterDepth
- LocationType
- TurbidityType
- BottomVisibleFlag
- BankMaterial
- BedrockType
- OxygenKitManufacturer
- OxygenKitModel
- AlkalinityKitManufacturer
- AlkalinityKitModel
- Comments

The primary and foreign key pair, ensures that there is at most one set of site characteristics recorded in the Globe database for each Surface Water study site. Site characteristics do not vary over time.

The foreign key FK1 references the StudySite entity, and enforces a one-to-one mapping from this SurfaceWaterSite entity into the StudySite entity.

The MovingWaterType attribute is specified on page 3-68 of the GLOBE II Teacher's Guide to be one of: Stream, River, Other, or Missing.

The StandingWaterType attribute is specified on page 3-68 of the GLOBE II Teacher's Guide to be one of: Pond, Lake, Reservoir, or Other.

The LocationType attribute is specified on page 3-68 of the GLOBE II Teacher's Guide to be one of: Outlet, Bank, Bridge, Boat, or Inlet.

The TurbidityType attribute is specified on page 3-68 of the GLOBE II Teacher's Guide to be one of: Clear, Turbid, or Don't Know.

The BankMaterial attribute is specified on page 3-68 of the GLOBE II Teacher's Guide to be one of: Soil, Rock, Concrete, or Vegetated Bank.

The BedrockType attribute is specified on page 3-68 of the GLOBE II Teacher's Guide to be one of: Granite, Lime Stone, Volcanics, Mixed Sediments, or Don't Know.

The OxygenKitManufacturer attribute is specified on page 3-69 of the GLOBE II Teacher's Guide to be one of: LaMotte, Hoch, or Other.

The OxygenKitModel is specified on page 3-69 of the GLOBE II Teacher's Guide to be text provided by the user.

The AlkalinityKitManufacturer attribute is specified on page 3-69 of the GLOBE II Teacher's Guide to be one of: LaMotte, Hoch, or Other.

The AlkalinityKitModel is specified on page 3-69 of the GLOBE II Teacher's Guide to be text provided by the user.

The Comments attribute allows further textual explantion of site characteristics that may influence student measurements. This attribute is constrained to be a character string of length no more than 200 characters.

Other site characteristics that could have an effect on soil moisture readings could be accommodated by adding appropriate new attributes to this entity.

561.2 Annotation for table SURFACEWATER_SITE

The foreign key FK1 references the StudySite entity, and enforces a one-to-one mapping from this SurfaceWaterSite entity into the StudySite entity. An additional constraint is needed to ensure that the referenced site is in fact a SurfaceWater study site, i.e. a site whose MeasurementCode attribute is equal to SW. There are various ways to enforce this additional constraint in the Physical Database Design.

The foreign key FK1 does not require that every SurfaceWater study site have a set of site characteristics associated with it; in many cases a SurfaceWater study site may be created and recorded in the StudySite entity first, with the site characteristics measured and recorded in this entity later, possibly never. If it is a GLOBE requirement that every SurfaceWater study site have the attributes of this entity always recorded in the database, then an additional constraint to enforce this rule is needed.

This entity does not have a DatePerformed or DateRecorded attribute; if it is important to maintain that information, then these attributes could be added here, or a new SiteActivity entity, analogous to the DataActivity entity discussed under the Student Measurements information category, could be defined.

The Hydrology Study Site Data Entry Sheet on page 3-68 of the GLOBE II Teacher's Guide asks for a Classification of the sampled water body as Moving Water or Standing Water. If it is Standing Water, then the data entry sheet asks for an approximation of the water body's area and average depth. This information is captured in the existing FSL GLOBE2SWSITELOCATION table and in the above attributes for moving water or standing water. However, if multiple Surface Water study sites are located around the same body of water, then this information is replicated on each study site data entry sheet, and there is no guarantee that the replicated data will be consistent. It might be better to consider the body of water as a study site "Feature" that is identified and described only once, independently of any specific study site, and then "associated" with those sites that it is relevant to. If this alternative approach is desirable, then the Feature name and description could be recorded in the AllFeatures entity, its dimensions recorded in the FeatureDimensions entity, and its location relative to associated Surface Water study sites recorded in the SiteToFeature association entity. Each of these potential new entity types is discussed under the Study Site Features information category of the Strawman GLOBE Database Design.

This entity is most directly related to the existing NOAA/FSL GLOBE2SWSITELOCATION table.

562 Table SURFACEWATER_SITE_CNT

563 Table SURFACE_WATER

563.1 Description of table SURFACE_WATER

The SurfaceWater table records weekly or monthly (or daily?) surface water measurements for temperature, acidity, dissolved oxygen, alkalinity, nitrate dilution, electrical conductivity, and turbidity of fresh water sources, as well as salinity and high and low tide measurements for salt water sources.

The turbidity measurements can be made by either a Secchi Disk protocol or a Turbidity Tube protocol. In either case, multiple measurements are recorded in a worksheet table with only derived results posted in this table.

Sometimes the turbidity measurements will exceed the length of the measurement tool; for eample, the Secchi disk may not disappear before all the rope is let out, or the pattern at the bottom of the tube may always be visible. In these cases, the greater than (GT) flag is set to YES to indicate that the measurement recorded is just an upper estimate, and that the actual measurement would have been greater.

564 Table SURFACE_WATER_CNT

565 Table SW_CALIBRATION

565.1 Description of table SW_CALIBRATION

The SW Calibration table records various calibrations of standards used in Surface Water measurements for salinity, alkalinity, and nitrates. Normally, these calibrations are made at least once every 6 months. See the Surface Water calibration Darta Entry Sheet or the Teacher's Guide protocols for further descriptions of the individual calibrations.

566 Table T2T_MESSAGE

566.1 Description of table T2T_MESSAGE

The T2T MESSAGE table is used to store messages entered through the Educator's Forum web pages.

567 Table TASK_TO_CONCEPT

568 Table TASK_TO_LEARNING

569 Table TEACHER_AGT

569.1 Description of table TEACHER_AGT

A view of the Globe Teacher base table, restricted to teachers who are Active and Globe-Trained.

570 Table TEACHER_EVALUATION

571 Table TEACHER_TO_SCHOOL

571.1 Description of table TEACHER_TO_SCHOOL

The TeacherToSchool table maintains a many-to-many relationship between GlobeTeacher and GlobeSchool. Its primary purpose is to record a historical trace of GLOBE teachers as they move from one school to another, but it can also be used to track teachers that are active at multiple schools simultaneously, e.g. a resource teacher in a county school system who helps to establish GLOBE activities at multiple schools.

The table records a DateBegin and a DateEnd for the association of a teacher with a school. If the DateEnd is NULL, than the teacher is currently active at the associated school.

The TeacherRole column identifies the role played by the teacher at the associated school. Values are from the TeacherRole domain.

572 Table TEMPERATURE_LOGGER

573 Table TEMPERATURE_LOGGER_CNT

574 Table TEMP_EXTREMES

575 Table THERMOMCALIB_WS

576 Table THERMOM_CALIB

577 Table THERMOM_RESET

578 Table TM_IMAGE

578.1 Description of table TM_IMAGE

The TM IMAGE table represents the status of satellite images that have been requested or sent to a school, showing the school's location at the center of the image. The processing of information for this table usually follows the following steps:

1) GLOBE decides which schools should be sent a satellite image and makes a request to EROS. The request includes the SchoolId, the Latitude and Longitude of the school (from SiteId='SCH-01' in the SITE LOCATION table), and the school's mailing address from the GLOBE SCHOOL table. Every request results in a new row being entered into the TM IMAGE table, with values for SchoolId and RequestDate, and all other columns left null.

2) EROS receives the requests from GLOBE and within an appropriate time period sends back status information for every request made. The return information includes the SchoolId and RequestDate, to identify a row in TM IMAGE, along with a non-null value for either ShipDate or Status. The possible values for Status are listed in the ImgStatusDomain.

3) If ShipDate is non-null and Status is Null, then EROS also supplies values for the remaining columns in the TM IMAGE table.

4) If ShipDate is Null and Status is non-null (i.e. Delay or Failed), then all image attribute columns are set to null, and, if appropriate, a Comment is added to further explain the reason that an image was not shipped.

5) GLOBE May unilaterally send EROS a request that an image shipping request be canceled, by sending EROS a list of (SchoolId,RequestDate) pairs. EROS will attempt to stop processing the image shipment. No further action need be taken by EROS. GLOBE will change the Status to Cancel.

579 Table TRAINER_EVALUATION

580 Table TRAINER_LANGUAGES

581 Table TRAINER_PROTOCOLS

582 Table TRAINER_TRAINING

583 Table TRAINING_TARGET

584 Table TREEBIOM_WS

584.1 Description of table TREEBIOM_WS

The TreeBiomWS table records multiple student measurements of tree height and tree circumference for both the dominant and codominant genus and species of trees at a given Land Cover or Biology study site.

The height and circumference values are then averaged and the results stored in the DomHeightAvg, DomCircumAvg, CodomHeightAvg, and CodomCircumAvg attributes of the TreeBiometry table.

585 Table TREEBIOM_WS_CNT

586 Table TREE_BIOMETRY

586.1 Description of table TREE_BIOMETRY

The TreeBiometry table records tree canopy measurements (by counting squares) and average tree height and circumference measurements for both the dominant and codominant genus and species of trees at a given Land Cover or Biology study site. The height and circumference averages are derived from data taken from multiple samples stored in theTreeBiomWS table.

587 Table TREE_BIOMETRY_CNT

588 Table TREE_TYPES

589 Table TURBIDITY_WS

589.1 Description of table TURBIDITY_WS

The TurbidityWS table records multiple sampling values for turbidity for each measurement day with averaged values carried over to the SurfaceWater table.

Turbidity is measured by one of the following methods:

Secchi Disk Method

A Secchi Disk is lowered into the water source with a depth measurement (meters) recorded for when the disk first disappears and a second depth measurement (meters) recorded for when the disk reappears.

Turbidity Tube Method

A sample of the water source is returned to the laboratory and poured into a turbidity tube; the turbidity measurement is the depth of water in the tube (mm) at which the pattern on the bottom of the tube disappears. In some cases, where the water is very clear, the pattern at the bottom of the tube does not disappear, even when the tube is filled to capacity. In such cases, the TubeGTFlag is set to "Yes", indicating that the tube was full, the TubeDepth measurement was equal to the TubeLength, and the pattern at the bottom was still visible. In those cases, the TubeLength is just a derived value equal to the measured TubeDepth. In other situations, the TubeLength is set to the null value.

590 Table TURBIDITY_WS_CNT

591 Table URBANIZATION_CODES

591.1 Description of table URBANIZATION_CODES

The UrbanizationCodes entity describes codes used to identify the urban environment in which a GLOBE School is located. An urban environment may vary from dense urban, i.e. co-located with multi-story office buildings in a large city, to a small town or village. It is likely that students will use these values to help locate other schools that are "like" themselves, thereby fostering school-to-school communication and possible relationships as "school partners".

UrbanizationCodes entity
- UrbanizationCode (primary key)
- UrbanizationName
- Description
- Description-French
- Description-Russian
- Description-Arabic
- Description-Chinese

The UrbanizationCode attribute is a specific 2-character code; it is the primary key of this entity.

The UrbanizationName attribute identifies the name of a given UrbanizationCode; the list of allowable codes and names, with a brief description, is as follows:

DU	Dense Urban	Large city (>500K) with dense central city atmosphere
OU	Open Urban	Large city (>500K) but co-located with open areas (parks, forests).
SB	Suburban	Satellite to large city, within 30km
MC	Moderate City	Moderate sized city (between 100K and 500K)
SC	Small City	Small city (between 30K and 100K)
TT	Town	Independent town (less than 30K)
VV	Village	Unincorporated village (less than 3-5K)
RR	Rural	School located away from any urban area

The Description attribute contains the descriptive English language text for each UrbanizationCode; the other associated Description attributes give urbanization code descriptions in other GLOBE supported languages, if available.

591.2 Annotation for table URBANIZATION_CODES

These codes and names are not aligned with any agreed sociological categorization of urban areas; if a suitable categorization exits, then it should be used instead.

592 Table USREGIONS

593 Table WEEKLY_ADDDROP_RPT

593.1 Description of table WEEKLY_ADDDROP_RPT

A view of the REGISTRATION base table, joined with several attributes from the Globe Teacher and Globe School base tables. Shows Add/Drops within the last 6 days.

594 Table WORKSHOP

594.1 Description of table WORKSHOP

The Workshop table identifies and describes general GLOBE workshops (not a session of a workshop). Workshops are also classified as to type, e.g. Traditional, Field Initiated Training, Franchise, and International. Separate subtables may be required at some future time in order to capture all of the attributes relevant to each workshop classification. Initially, the workshop types are:

CONT	Contract
FIT	Field Initiated Training
FRAN-TR	Franchise-Trainer

FRAN-TE	Franchise-Teacher
INT-R	International - Regional
INT-C	International - In Country
GTTT	GLOBE Train the Trainer
HOST	Hosted
CONF	Conference
SPEC	Special

595 Table WORKSHOP_CONTACTS

595.1 Description of table WORKSHOP_CONTACTS

The WorkshopContacts table identifies all of the session site personnel for various items, including: lodging, site administration, site coordination, computer rentals, remote sensing equipment, public affairs, shipping, and hotel. Each row of the table gives complete address, telephone, and other contact information for an identified contact person. The ContactCode is part of the primary key and its potential values are described by the ContactCodesDomain. The initial Contact Codes are:

LODGING	Lodging Contact
ADMIN	Administrative Contact
CONTRACT	Contract Contact
W-COORD	Workshop Coordinator
C-COORD	Country Coordinator
US-EMB	U.S. Embassy Contact
COMPUTER	Computer Contact
R-SENS	Remote Sensing Contact
PRESS	Public Affairs Contact
SHIP-S	Ship Small Items Contact
SHIP-L	Ship Large Items Contact
HOTEL	Hotel Contact

596 Table WORKSHOP_DROPADD

596.1 Description of table WORKSHOP_DROPADD

A view of the REGISTRATION base table, joined with several identifying attributes from the Globe Teacher base table. Only the REGISTRATION items will be Updatable, not the Teacher items.

597 Table WORKSHOP_SESSION

597.1 Description of table WORKSHOP_SESSION

The WorkshopSession table identifies each of the sessions at which a given Globe workshop type is to be offered. Session information includes location, dates, enrollment limits, budget, hotel rates, meal fees, and academic credits being offered.

Every workshop session must have a non-null workshop name to indicate the general category of workshop it is. The possibilities are defined in the Workshop Domain.

Every workshop session must have a non-null CountryCode to indicate which country will host the workshop.

Every workshop session must have a non-null BeginDate. However, it may have a null EndDate. A null value for EndDate is a subtle hint that the Date given for BeginDate is really just an estimate. It mainly applies to past international workshops where information is very sketchy, or to planned future workshops where exact Dates have not yet been fixed. If a workshop has a split session over non-consecutive days, then the EndDate is the last day of the session and SplitSessionText describes the schedule for meetings.

A required value for BeginDate also means we can have a required value for Season, because Season and BeginDate must be consistent. Season serves as a category to simplify some common queries; it is defined by the Season Domain.

Every workshop session must have a non-null value for CurrencyCountry. This identifies the currency that is assumed for all of the workshop fees identified by other numeric columns in this table, i.e. MealFee and CreditFee, as well as the interpretation of any textual numbers in the HotelRateRange column.

597.2 Annotation for table WORKSHOP_SESSION

We need to decide how to represent teachers who need to be trained, but who have not specified any preferences for exisitng workshop sessions, or whose preferences were to workshops already held, or who registered for a workshop but failed to attend or complete the workshop and have no other preferences remaining.

We could create a superfluous "TBD" session to link such teachers to. The TBD identification could be used to identify candidate

participants for future workshops.

Alternatively, we could derive this same information by searching on GlobeTeacher rows with LifeCycleCode = 'Candidate' and where the TeacherId does not appear in any row of the Registration table.

598 Table WORKSHOP_STAFF

598.1 Description of table WORKSHOP_STAFF

The WorkshopStaff table maintains a many-to-many mapping among GLOBE training staff and GLOBE workshop sessions. It also identifies the role to be played by each staff person at each associated workshop session. The roles are limited to those defined by numeric codes in the TrainerRoleCode domain.

599 Table WWW_USERS

599.1 Description of table WWW_USERS

A snapshot table derived from a Union view of all Teachers, Schools, and Franchises that are authorized to access GLOBE WWW pages for some form of Update.

600 Table ZIP_INFO

601 View ACIDITY_WS_DR1

602 View ACIDITY_WS_DR2

602.1 SQL Query for view ACIDITY_WS_DR2

```
select ACIDITY_WS_DR1.*, (SpHSampCnt) as DixonCnt
from ACIDITY_WS_DR1
```

603 View ACIDITY_WS_DU1

604 View ACIDITY_WS_DU2

604.1 SQL Query for view ACIDITY_WS_DU2

```
select ACIDITY_WS_DU1.*, (SpHSampCnt) as DixonCnt
from ACIDITY_WS_DU1
```

605 View ACIDITY_WS_MC1

605.1 SQL Query for view ACIDITY_WS_MC1

```
select 'SC' as MeasureCode, COUNT(SoilAndWaterpH+DistilledWaterpH) as SpHSampCnt
from GLOBE.ACIDITY_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.H  or  izonNbr = S.H or izonNbr
```

606 View ACIDITY_WS_MC2

606.1 SQL Query for view ACIDITY_WS_MC2

```
select ACIDITY_WS_MC1.*, (SpHSampCnt) as DixonCnt
from ACIDITY_WS_MC1
```

607 View ACIDITY_WS_RPRT_VU

607.1 SQL Query for view ACIDITY_WS_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from ACIDITY_WS
group by SchoolID
```

608 View ACIDITY_WS_VU1

608.1 SQL Query for view ACIDITY_WS_VU1

```
select
 W.SchoolId,
 W.SiteId,
 TRUNC(S.UTtimeMeasured,'MM') AS Month,
 'SC' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT(SoilAndWaterpH+DistilledWaterpH) AS SpHSampCnt,
 MIN(S.UTtimeMeasured) AS BegDate,
 MAX(S.UTtimeMeasured) AS EndDate,
 MIN(S.UTtimeReported) as BegRDate,
 MAX(S.UTtimeReported) as EndRDate,
 ROUND(AVG(W.UTtimeReported-S.UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.ACIDITY_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.HorizonNbr = S.HorizonNbr and
W.UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by W.SchoolId, W.SiteId, TRUNC(S.UTtimeMeasured,'MM')
```

609 View ACIDITY_WS_VU2

609.1 SQL Query for view ACIDITY_WS_VU2

```
select ACIDITY_WS_VU1.*, (SpHSampCnt) as DixonCnt
from ACIDITY_WS_VU1
```

610 View AIR_TEMP_DR1

611 View AIR_TEMP_DR2

611.1 SQL Query for view AIR_TEMP_DR2

```
select AIR_TEMP_DR1.*, (curtcnt+maxtcnt+mintcnt) as DixonCnt
from AIR_TEMP_DR1
```

612 View AIR_TEMP_DU1

613 View AIR_TEMP_DU2

613.1 SQL Query for view AIR_TEMP_DU2

```
select AIR_TEMP_DU1.*, (curtcnt+maxtcnt+mintcnt) as DixonCnt
from AIR_TEMP_DU1
```

614 View AIR_TEMP_MC1

614.1 SQL Query for view AIR_TEMP_MC1

```
select 'AT' as MeasureCode, COUNT(*) as RowCnt, COUNT(CurrentTemp) as CurTCnt, Count(MaxTemp) as
MaxTCnt, COUNT(MinTemp) as MinTCnt
from GLOBE.AIR_TEMP
```

615 View AIR_TEMP_MC2

615.1 SQL Query for view AIR_TEMP_MC2

```
select AIR_TEMP_MC1.*, (curtcnt+maxtcnt+mintcnt) as DixonCnt
from AIR_TEMP_MC1
```

616 View AIR_TEMP_RPRT_VU

616.1 SQL Query for view AIR_TEMP_RPRT_VU

```
select Distinct as SchoolID, 'AT' as MeasureCOde, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from AIR_TEMP
group by SchoolID
```

617 View AIR_TEMP_VU1

617.1 SQL Query for view AIR_TEMP_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay,'MM') AS Month,
 'AT' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(CurrentTemp) AS CurTCnt,
 MIN(CurrentTemp) as CurTMin,
 MAX(CurrentTemp) as CurTMax,
```

```
 ROUND(AVG(CurrentTemp),1) as CurTAvg,
 ROUND(STDDEV(CurrentTemp),1) as CurTStd,
 Count(MaxTemp) AS MaxTCnt,
 MIN(MaxTemp) as MaxTMin,
 MAX(MaxTemp) as MaxTMax,
 ROUND(AVG(MaxTemp),1) AS MaxTAvg,
 ROUND(STDDEV(MaxTemp),1) as MaxTStd,
 COUNT(MinTemp) AS MinTCnt,
 MIN(MinTemp) as MinTMin, MAX(MinTemp) as MinTMax,
 ROUND(AVG(MinTemp),1) AS MinTAvg,
 ROUND(STDDEV(MinTemp),1) as MinTStd,
 COUNT(MaxTemp+MinTemp) AS MeanTCnt,
 ROUND(AVG((MaxTemp+MinTemp)/2.),1) AS MeanTAvg,
 ROUND(AVG((MaxTemp-MinTemp)),1) AS TRangeAvg,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) as BegRDate,
 MAX(UTtimeReported) as EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
 from
 GLOBE.AIR_TEMP
 where
 UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
 group by
 SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

618 View AIR_TEMP_VU2

618.1 SQL Query for view AIR_TEMP_VU2

```
select AIR_TEMP_VU1.*, (curtcnt+maxtcnt+mintcnt) as DixonCnt
from AIR_TEMP_VU1
```

619 View ATMOSADV_HR1

619.1 SQL Query for view ATMOSADV_HR1

```
select HRnbr, SchoolId, 'Acidity' as Measurement from Acidity_HR4
UNION ALL
select HRnbr, SchoolId, 'Humidity' as Measurement from Humidity_HR4
UNION ALL
select HRnbr, SchoolId, 'Ozone' as Measurement from Ozone_HR4
UNION ALL
select HRnbr, SchoolId, 'Aerosol' as Measurement from Aerosol_HR4
```

620 View ATMOSADV_HR2

620.1 SQL Query for view ATMOSADV_HR2

```
select
 HRnbr,
 SchoolId,
 1 as DixonCnt,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+ 2*(HRnbr-1)*31,'MM') as BeginMonth,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM') as EndMonth
from ATMOSADV_HR1
group by
 HRnbr,
 SchoolId,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+2*(HRnbr-1)*31,'MM'),
 TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM')
having Count(*) >= 2
```

621 View ATMOSADV_HR3

621.1 SQL Query for view ATMOSADV_HR3

```
select
 AA.HRnbr,
 AA.SchoolId,
 AA.DixonCnt,
```

```
 AA.BeginMonth,
 AA.EndMonth
from
 ATMOSADV_HR2 AA,
 GLOBE.HR_ATMOS_CURRENT AC
where
 AA.HRnbr = AC.HRnbr AND
 AA.SchoolId = AC.SchoolId
```

622 View ATMOSPHERE_SITE_DR1

622.1 SQL Query for view ATMOSPHERE_SITE_DR1

```
select TRUNC(UTtimeStamp,'DD') as Day, 'MD' as MeasureCode, COUNT(*) as RowCnt,
COUNT(ObstacleDistance) as ObstDistCnt, COUNT(ObstacleHeight) as ObstHtCnt,
SUM(DECODE(SurfaceCover,0,0,1)) AS SurfCovCntFROM GLOBE.ATMOSPHERE_SITEWHERE UTtimeStamp < TRUNC
(SYSDATE,'DD') + 22/24GROUP BY TRUNC as (UTtimeStamp,'DD')
```

623 View ATMOSPHERE_SITE_DR2

623.1 SQL Query for view ATMOSPHERE_SITE_DR2

```
select ATMOSPHERE_SITE_DR1.*, (obstdistcnt+obsthtcnt+surfcovcnt) as DixonCnt
from ATMOSPHERE_SITE_DR1
```

624 View ATMOSPHERE_SITE_DU1

624.1 SQL Query for view ATMOSPHERE_SITE_DU1

```
select TRUNC(VersionDate,'DD') as Day, 'MD' as MeasureCode, COUNT(*) as RowCnt,
COUNT(ObstacleDistance) as ObstDistCnt, COUNT(ObstacleHeight) as ObstHtCnt,
SUM(DECODE(SurfaceCover,0,0,1)) AS SurfCovCntFROM GLOBE.ATMOSPHERE_SITEWHERE UTtimeStamp < TRUNC
(SYSDATE,'DD') + 22/24GROUP BY TRUNC as (VersionDate,'DD')
```

625 View ATMOSPHERE_SITE_DU2

625.1 SQL Query for view ATMOSPHERE_SITE_DU2

```
select ATMOSPHERE_SITE_DU1.*, (obstdistcnt+obsthtcnt+surfcovcnt) as DixonCnt
from ATMOSPHERE_SITE_DU1
```

626 View ATMOSPHERE_SITE_MC1

626.1 SQL Query for view ATMOSPHERE_SITE_MC1

```
select 'MD' as MeasureCode, COUNT(*) as RowCnt, COUNT(ObstacleDistance) as ObstDistCnt,
COUNT(ObstacleHeight) as ObstHtCnt, SUM(DECODE(SurfaceCover,0,0,1)) AS SurfCovCntFROM as
GLOBE.ATMOSPHERE_SITE
```

627 View ATMOSPHERE_SITE_MC2

627.1 SQL Query for view ATMOSPHERE_SITE_MC2

```
select ATMOSPHERE_SITE_MC1.*, (obstdistcnt+obsthtcnt+surfcovcnt) as DixonCnt
from ATMOSPHERE_SITE_MC1
```

628 View ATMOSPHERE_SITE_RPRT_VU

628.1 SQL Query for view ATMOSPHERE_SITE_RPRT_VU

```
select Distinct as SchoolID, 'AT' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from ATMOSPHERE_SITE
group by SchoolID
```

629 View ATMOSPHERE_SITE_VU1

629.1 SQL Query for view ATMOSPHERE_SITE_VU1

```
select
 Schoolid,
 Siteid, T
 RUNC(VersionDate,'MM') AS Month,
 'MD' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(ObstacleDistance) AS ObstDistCnt,
 COUNT(ObstacleHeight) AS ObstHtCnt,
 SUM(DECODE(SurfaceCover,0,0,1)) AS SurfCovCnt,
 MIN(VersionDate) AS BegDate,
 MAX(VersionDate) AS EndDate,
 MIN(UTtimestamp) as BegRDate,
 MAX(UTtimestamp) as EndRDate,
 ROUND(AVG(UTtimestamp-VersionDate),2) AS LagTimeAvg
from GLOBE.ATMOSPHERE_SITE
where UTtimeStamp < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(VersionDate,'MM')
```

630 View ATMOSPHERE_SITE_VU2

630.1 SQL Query for view ATMOSPHERE_SITE_VU2

```
select
 ATMOSPHERE_SITE_VU1.*,
 (obstdistcnt+obsthtcnt+surfcovcnt) AS DixonCnt
from ATMOSPHERE_SITE_VU1
```

631 View ATMOS_HR1

631.1 SQL Query for view ATMOS_HR1

```
select
2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.Month) as BeginMonth,
MAX(T.Month) as EndMonth,
SUM(T.TempDays) as TempDays,
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) as PrecipDays,
SUM(DX.DixonCnt) as DixonCnt
From ATMOS_PRECIP_HR2 P, ATMOS_TEMP_HR1 T, ATMOS_HRDX DX
Where
P.SchoolId = T.SchoolId
AND
P.SiteId = T.SiteId
AND
P.Month = T.Month
AND
DX.SchoolId = T.SchoolId
AND
DX.SiteId = T.SiteId
AND
DX.Month = T.Month
Group By T.SchoolId, T.SiteId,
2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
HAVING
MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
AND
SUM(T.TempDays) >= 4*30*(.75)
AND
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) >= 4*30*(.95)
UNION ALL
select
2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.Month) as BeginMonth,
MAX(T.Month) as EndMonth,
```

```
SUM(T.TempDays) as TempDays,
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) as PrecipDays,
SUM(DX.DixonCnt) as DixonCnt
From ATMOS_PRECIP_HR2 P, ATMOS_TEMP_HR1 T, ATMOS_HRDX DX
Where
P.SchoolId = T.SchoolId
AND
P.SiteId = T.SiteId
AND
P.Month = T.Month
AND
DX.SchoolId = T.SchoolId
AND
DX.SiteId = T.SiteId
AND
DX.Month = T.Month
Group By T.SchoolId, T.SiteId,
2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
HAVING
MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
AND
SUM(TempDays) >= 4*30*(.75)
AND
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) >= 4*30*(.95)
```

632 View ATMOS_HR2

632.1 SQL Query for view ATMOS_HR2

```
select
 ATMOS_HR1.*,
 DECODE(SITE_LOCATION.DataSource,0,'Unchecked',1,'GPS',2,'Other') as LocMethod
from GLOBE.ATMOS_HR1, GLOBE.SITE_LOCATION
where
 ATMOS_HR1.SchoolId = SITE_LOCATION.SCHOOLID AND
 ATMOS_HR1.SiteId = SITE_LOCATION.SITEID
```

633 View ATMOS_HR3

633.1 SQL Query for view ATMOS_HR3

```
select
 HRnbr,
 SchoolId,
 SUM(DixonCnt) as DixonCnt,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+ 2*(HRnbr-1)*31,'MM') as BeginMonth,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM') as EndMonth
from ATMOS_HR2
where LocMethod = 'GPS' or LocMethod = 'Other'
group by
 HRnbr,
 SchoolId,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+2*(HRnbr-1)*31,'MM'),
 TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM')
```

634 View ATMOS_HRDX

634.1 SQL Query for view ATMOS_HRDX

```
select
 SchoolId,
 SiteId,
 Month,
 'AT_PR_PS' AS Scope,
 SUM(DixonCnt) AS DixonCnt
from GLOBE.DIXON_SUMMARY_VU2
where
 Month > = TO_DATE ( '1999 02 01' , 'YYYY MM DD' ) AND
 Month < TRUNC ( SYSDATE , 'MM' ) AND
 MeasureCode IN ( 'AT' , 'PR' , 'PS' )
group by SchoolId, SiteId, Month, 'AT_PR_PS'
```

635 View ATMOS_PRECIP_HR1

635.1 SQL Query for view ATMOS_PRECIP_HR1

```
select "SCHOOLID","SITEID","MONTH","SUMRDAYS","SUMSDAYS","SUMREDAYS" From ATMOS_PR_HR1
UNION ALL
select "SCHOOLID","SITEID","MONTH","SUMRDAYS","SUMSDAYS","SUMREDAYS" From ATMOS_PS_HR1
```

636 View ATMOS_PRECIP_HR2

636.1 SQL Query for view ATMOS_PRECIP_HR2

```
select
 SchoolId,
 SiteId,
 Month,
 SUM(SumRDays) as SumRDays,
 SUM(SumSDays) as SumSDays,
 SUM(SumREDays) as SumREDays
from ATMOS_PRECIP_HR1
group by SchoolId, SiteId, Month
```

637 View ATMOS_PR_HR1

637.1 SQL Query for view ATMOS_PR_HR1

```
select SchoolId, SiteId, Month, SumRDays, 0 as SumSDays, 0 as SumREDays
from GLOBE.PRECIP_RAIN_CNT
where Month > =  TO_DATE  (
'1999
02
01'
,
'YYYY
MM
DD'
)  AND  Month  <  TRUNC  (
SYSDATE
,
'MM'
)
```

638 View ATMOS_PS_HR1

638.1 SQL Query for view ATMOS_PS_HR1

```
select SchoolId, SiteId, Month, 0 as SumRDays, SumSDays, SumREDays
from GLOBE.PRECIP_SOLID_CNT
where Month > =  TO_DATE  (
'1999
02
01'
,
'YYYY
MM
DD'
)  AND  Month  <  TRUNC  (
SYSDATE
,
'MM'
)
```

639 View ATMOS_TEMP_HR1

639.1 SQL Query for view ATMOS_TEMP_HR1

```
select
 ATC.SchoolId,
```

```
 ATC.SiteId,
 Month,
 CurTCnt,
 MaxTCnt,
 MinTCnt,
 Round((MaxTCnt + MinTCnt)/2) as TempDays
from
 GLOBE.AIR_TEMP_CNT ATC,
 GLOBE.SITE_LOCATION SL
where
 ATC.SchoolId = SL.SchoolId AND
 ATC.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 Month >= TO_DATE('19990201','YYYYMMDD')
```

640 View ATMOS_TEMP_HR2

640.1 SQL Query for view ATMOS_TEMP_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.TempDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 GLOBE.ATMOS_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.TempDays) >= 4*30*(.70)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.TempDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 GLOBE.ATMOS_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.TempDays) >= 4*30*(.70)
```

641 View AcidityRain_HR1

641.1 SQL Query for view AcidityRain_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(Acidity) as GoodDays,
 Count(*) as PhDays
from
 GLOBE.PRECIP_RAIN
where
 SolarDay >= TO_DATE('20010201','YYYYMMDD') AND
 RainAmount >= 5.0
group by SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

642 View AciditySnow_HR1

642.1 SQL Query for view AcidityySnow_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(Acidity) as GoodDays,
 Count(*) as PhDays
from
 GLOBE.PRECIP_SOLID
where
 SolarDay >= TO_DATE('20010201','YYYYMMDD') AND
 RainEquivalent >= 5.0
group by SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

643 View Acidity_HR2

643.1 SQL Query for view Acidity_HR2

```
select
 SchoolId,
 SiteId,
 Month,
 SUM(GoodDays) as GoodDays,
 SUM(PhDays) as PhDays
from
 (select * from AcidityRain_HR1
  UNION ALL
  select * from AcidityySnow_HR1)
group by SchoolId, SiteId, Month
```

644 View Acidity_HR3

644.1 SQL Query for view Acidity_HR3

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 SUM(T.PhDays) as PhDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Acidity_HR2 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 >= 1
 AND
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('19990401','YYYYMMDD')))/2)
 AND
 SUM(T.GoodDays)/SUM(T.PhDays) >= 0.8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 SUM(T.PhDays) as PhDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Acidity_HR2 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) >= 1
 AND
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) <=
```

```
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('19990401','YYYYMMDD')))/2)
 AND
 SUM(T.GoodDays)/SUM(T.PhDays) >= 0.8
```

645 View Acidity_HR4

645.1 SQL Query for view Acidity_HR4

```
select distinct
 HRnbr,
 SL.SchoolId,
 'Acidity' as Measurement,
 BeginMonth,
 EndMonth
from
 Acidity_HR3 HR3,
 GLOBE.SITE_LOCATION SL
where
 HRnbr >= 13 AND
 HR3.SchoolId = SL.SCHOOLID AND
 HR3.SiteId = SL.SITEID AND
 SL.DataSource IN (1,2)
```

646 View Aerosol_HR1

646.1 SQL Query for view Aerosol_HR1

```
select
 SchoolId,
 SiteId,
 SolarDay
from
 GLOBE.HAZE_WS
where
 SolarDay >= TO_DATE('20010201','YYYYMMDD') AND
 DarkVoltage is not null AND
 SunlightVoltage is not null
group by SchoolId, SiteId, SolarDay
```

647 View Aerosol_HR2

647.1 SQL Query for view Aerosol_HR2

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(*) as GoodDays
from
 Aerosol_HR1
group by
 SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

648 View Aerosol_HR3

648.1 SQL Query for view Aerosol_HR3

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Aerosol_HR2 T
```

```
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.GoodDays) >= 4*30*(.30)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Aerosol_HR2 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.GoodDays) >= 4*30*(.30)
```

649 View Aerosol_HR4

649.1 SQL Query for view Aerosol_HR4

```
select distinct
 HRnbr,
 SL.SchoolId,
 'Aerosol' as Measurement,
 BeginMonth,
 EndMonth
from
 Aerosol_HR3 HR3, GLOBE.SITE_LOCATION SL
where
 HRnbr >= 13 AND
 HR3.SchoolId = SL.SCHOOLID AND
 HR3.SiteId = SL.SITEID AND
 SL.DataSource IN (1,2)
```

650 View BAROM_PRESSURE_VU1

650.1 SQL Query for view BAROM_PRESSURE_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay,'MM') AS Month,
 'BP' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(BarometricMb) AS MbCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
FROM GLOBE.BAROM_PRESSURE
WHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

651 View BAROM_PRESSURE_VU2

651.1 SQL Query for view BAROM_PRESSURE_VU2

```
select BAROM_PRESSURE_VU1.*,
 MbCnt AS DixonCnt
from BAROM_PRESSURE_VU1
```

652 View BBatmosLink_HR0

652.1 SQL Query for view BBatmosLink_HR0

```
select
 BB.SchoolId,
 BB.SiteId,
 MAX(PS.VersionDate) as VersionDate
from
 Budburst_HR1 BB,
 GLOBE.PHENOLOGY_SITE PS
where
 BB.SchoolId = PS.SchoolId AND
 BB.SiteId = PS.SiteId AND
 BB.Day >= PS.VersionDate
group by BB.SchoolId, BB.SiteId
```

653 View BBatmosLink_HR1

653.1 SQL Query for view BBatmosLink_HR1

```
select
 BB.SchoolId,
 BB.SiteId,
 BB.Day,
 PS.AtmosLink
from
 Budburst_HR1 BB,
 GLOBE.PHENOLOGY_SITE PS,
 BBatmosLink_HR0 PSL
where
 BB.SchoolId = PS.SchoolId(+) AND
 BB.SiteId = PS.SiteId(+) AND
 PS.SchoolId = PSL.SchoolId AND
 PS.SiteId = PSL.SiteId AND
 PS.VersionDate = PSL.VersionDate
```

654 View BBrain_HR0

654.1 SQL Query for view BBrain_HR0

```
select
 L.SchoolId,
 L.SiteId,
 L.Day,
 Sum(DaysAccumulated) as RainDays
from
 BBatmosLink_HR1 L,
 GLOBE.PRECIP_RAIN T
where
 L.SchoolId = T.SchoolId AND
 L.AtmosLink = T.SiteId AND
 T.SolarDay >= L.Day - 30 AND
 T.SolarDay <= L.Day
group by L.SchoolId, L.SiteId, L.Day
```

655 View BBsnow_HR0

655.1 SQL Query for view BBsnow_HR0

```
select
 L.SchoolId,
 L.SiteId,
 L.Day,
 Sum(DaysAccumulated) as SnowDays
from
 BBatmosLink_HR1 L,
 GLOBE.PRECIP_SOLID T
where
 L.SchoolId = T.SchoolId AND
 L.AtmosLink = T.SiteId AND
```

```
 T.SolarDay >= L.Day - 30 AND
 T.SolarDay <= L.Day
group by L.SchoolId, L.SiteId, L.Day
```

656 View BBtemp_HR1

656.1 SQL Query for view BBtemp_HR1

```
select
 L.SchoolId,
 L.SiteId,
 L.Day,
 Count(*) as TempDays
from
 BBatmosLink_HR1 L,
 GLOBE.AIR_TEMP T
where
 L.SchoolId = T.SchoolId AND
 L.AtmosLink = T.SiteId AND
 T.SolarDay >= L.Day - 30 AND
 T.SolarDay <= L.Day AND
 T.MaxTemp is not null AND
 T.MinTemp is not null
group by L.SchoolId, L.SiteId, L.Day
having Count(*) >= 25
```

657 View BBzPrecip_HR1

657.1 SQL Query for view BBzPrecip_HR1

```
select
 R.SchoolId,
 R.SiteId,
 R.Day,
 RainDays,
 SnowDays,
 RainDays + SnowDays as GoodDays
from
 BBrain_HR0 R,
 BBsnow_HR0 S
where
 R.SchoolId = S.SchoolId(+) AND
 R.SiteId = S.SiteId(+) AND
 R.Day = S.Day(+) AND
 (RainDays + SnowDays) >= 31
```

658 View BUDBURST_META_DR1

659 View BUDBURST_META_DR2

659.1 SQL Query for view BUDBURST_META_DR2

```
select BUDBURST_META_DR1.*, (HtCnt+CmCnt) as DixonCnt
from BUDBURST_META_DR1
```

660 View BUDBURST_META_DU1

661 View BUDBURST_META_DU2

661.1 SQL Query for view BUDBURST_META_DU2

```
select BUDBURST_META_DU1.*, (HtCnt+CmCnt) as DixonCnt
from BUDBURST_META_DU1
```

662 View BUDBURST_META_VU1

662.1 SQL Query for view BUDBURST_META_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(UTTimeReported,'MM') AS MONTH,
  'PH' AS MeasureCode,
  COUNT(*) AS RowCnt,
  COUNT(DECODE(Genus4,NULL,0,1)*DECODE(Species4,NULL,0,1)) GSpCnt,
  COUNT(Height) AS HtCnt,
  COUNT(Circumference) AS CmCnt,
  MIN(UTtimeReported) AS BegDate,
  MAX(UTtimeReported) AS EndDate,
  MIN(UTTimeReported) AS BegRDate,
  MAX(UTTimeReported) AS EndRDate,
  ROUND(NVL(AVG(UTTimeReported-DateMeasured),9999),2) AS LagTimeAvg
FROM
  BUDBURST_METADATA
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD')+ 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(UTTimeReported,'MM')
```

663 View BUDBURST_META_VU2

663.1 SQL Query for view BUDBURST_META_VU2

```
SELECT
  BUDBURST_META_VU1.*,
  (GSpCnt+HtCnt+CmCnt) AS DixonCnt
FROM BUDBURST_META_VU1
```

664 View BudBurst_HR1

664.1 SQL Query for view BudBurst_HR1

```
select
 BB.SchoolId,
 BB.SiteId,
 BB.Year,
 MIN(BB.BudburstDay) as BeginDay,
 MAX(BB.BudburstDay) as Day
from
 GLOBE.PHENOLOGY_WS BB,
 GLOBE.SITE_LOCATION SL
where
 BB.SchoolId = SL.SchoolId AND
 BB.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 BB.BudburstDay >= TO_DATE('20000601','YYYYMMDD')
group by BB.SchoolId, BB.SiteId, BB.Year
having
 Count(*) >= 4
```

665 View BudBurst_HR2

665.1 SQL Query for view BudBurst_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId, Max(Day) as Day
from Budburst_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId, Max(Day) as Day
from Budburst_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId, Max(Day) as Day
from Budburst_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId, Max(Day) as Day
from Budburst_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId, Max(Day) as Day
from Budburst_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId, Max(Day) as Day
from Budburst_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

666 View CERTMERGE_DIRECTOR_VU

666.1 SQL Query for view CERTMERGE_DIRECTOR_VU

```
select
 CS.SchoolId as "SchoolId",
 CS.TchrCnt as "TchrCnt",
 CS.TeacherList as "TeacherList",
 HR_CERTIFICATE_INFO_VU2.*,
 HR_CERTIFICATE_INFO_VU1.*,
 HR_CERTIFICATE_INFO_VU3.*,
 RC.NewRung as "NewRung",
 RC.OldRung as "OldRung",
 DS.BegDate as "FirstObs",
 DS.EndDate as "LastObs",
 DS.DixonCnt as "TotalObs",
 ROUND(MONTHS_BETWEEN(DS.EndDate, DS.BegDate)) as "ObsMonths",
 ROUND(DS.DixonCnt/MONTHS_BETWEEN(DS.EndDate, DS.BegDate)) as "ObsRate",
 AC."AllCnt",
 AC."AllYearCnt",
 AC."AllNewCnt",
 AC."AllPrevCnt"
from
 GLOBE.HR_CERTIFICATE_TEACHERS CS,
 GLOBE.HR_CERTIFICATE_INFO_VU1,
 GLOBE.HR_CERTIFICATE_INFO_VU2,
 GLOBE.HR_CERTIFICATE_INFO_VU3,
 HR_ALLCNT_SUMS AC,
 GLOBE.HR_RUNGCHG RC,
 GLOBE.DIXON_SUMMARY_CNT DS
where
 CS.SchoolId = HR_CERTIFICATE_INFO_VU1."Vu1SchoolId" AND
 CS.SchoolId = HR_CERTIFICATE_INFO_VU2."Vu2SchoolId" AND
 CS.SchoolId = HR_CERTIFICATE_INFO_VU3."Vu3SchoolId" AND
 CS.SchoolId = AC."SchoolId" AND
 CS.SchoolId = RC.SchoolId ( + ) AND
 CS.SchoolId = DS.SchoolId
```

667 View CERTMERGE_RUNGONLY_VU

667.1 SQL Query for view CERTMERGE_RUNGONLY_VU

```
select
 RC.SchoolId as "SchoolId",
 RC.TotalObs as "TotalObs",
 RC.NewRung as "NewRung",
 RC.OldRung as "OldRung",
 RC.FirstObs as "FirstObs",
 RC.LastObs as "LastObs",
 ROUND(MONTHS_BETWEEN(RC.LastObs, RC.FirstObs)) as "ObsMonths",
 ROUND(RC.TotalObs/MONTHS_BETWEEN(RC.LastObs, RC.FirstObs)) as "ObsRate",
 NVL(AC."AllCnt",0) as "AllCnt",
 NVL(AC."AllYearCnt",0) as "AllYearCnt",
 NVL(AC."AllNewCnt",0) as "AllNewCnt",
 NVL(AC."AllPrevCnt",0) as "AllPrevCnt",
 GS.SchoolName as "SchoolName",
 GS.AddressLine1 as "AddressLine1",
 GS.AddressLine2 as "AddressLine2",
 GS.City as "City",
 GS.StateProvCode as "StateProvCode",
 GS.PostalCode as "PostalCode",
 GS.CountryCode as "CountryCode",
 GC.GlobeName as "CountryName",
 GS.DirectorFullName as "DirectorFullName",
 GS.DirectorSalutation as "DirectorSalutation",
 GS.LifeCycleCode as "LifeCycleCode",
 GS.CycleChangeDate as "CycleChangeDate"
from
 GLOBE.HR_RUNGONLY_CERTIFICATE RC,
```

```
 GLOBE.GLOBE_SCHOOL GS,
 GLOBE.GLOBE_COUNTRY GC,
 HR_ALLCNT_SUMS AC
where
 RC.SchoolId = GS.SchoolId AND
 GS.CountryCode = GC.CountryCode AND
 RC.SchoolId = AC."SchoolId"(+)
```

668 View CERTMERGE_TEACHER_VU

668.1 SQL Query for view CERTMERGE_TEACHER_VU

```
select
 CS.SchoolId as "SchoolId",
 GT.TeacherId as "TeacherId",
 GT.TeacherPrefix as "Pre",
 GT.TeacherOtherNames as "FirstName",
 GT.TeacherSurname as "LastName",
 GT.TeacherSuffix as "Suf",
 GT.InternalAddress as "InternalAddress",
 GT.LifeCycleCode as "TeachLCcode",
 GT.CycleChangeDate as "TeachLCDate",
 HR_CERTIFICATE_INFO_VU2.*,
 HR_CERTIFICATE_INFO_VU1.*,
 HR_CERTIFICATE_INFO_VU3.*,
 RC.NewRung as "NewRung",
 RC.OldRung as "OldRung",
 DS.BegDate as "FirstObs",
 DS.EndDate as "LastObs",
 DS.DixonCnt as "TotalObs",
 ROUND(MONTHS_BETWEEN(DS.EndDate, DS.BegDate)) as "ObsMonths",
 ROUND(DS.DixonCnt/MONTHS_BETWEEN(DS.EndDate, DS.BegDate)) as "ObsRate",
 AC."AllCnt",
 AC."AllYearCnt",
 AC."AllNewCnt",
 AC."AllPrevCnt"
from
 GLOBE.HR_CERTIFICATES CS,
 GLOBE.TEACHER_TO_SCHOOL TS,
 GLOBE.GLOBE_TEACHER GT,
 GLOBE.HR_CERTIFICATE_INFO_VU1,
 GLOBE.HR_CERTIFICATE_INFO_VU2,
 GLOBE.HR_CERTIFICATE_INFO_VU3,
 HR_ALLCNT_SUMS AC,
 GLOBE.HR_RUNGCHG RC,
 GLOBE.DIXON_SUMMARY_CNT DS
where
 CS.SchoolId = TS.SchoolId AND
 TS.TeacherId = GT.TeacherId AND
 TS.DateEnd IS NULL AND
 GT.LifeCycleCode <> 'I' AND
 CS.SchoolId = HR_CERTIFICATE_INFO_VU1."Vu1SchoolId" AND
 CS.SchoolId = HR_CERTIFICATE_INFO_VU2."Vu2SchoolId" AND
 CS.SchoolId = HR_CERTIFICATE_INFO_VU3."Vu3SchoolId" AND
 CS.SchoolId = AC."SchoolId" AND
 CS.SchoolId = RC.SchoolId (+) AND
 CS.SchoolId = DS.SchoolId
```

669 View CLIMATE_1_HR1

669.1 SQL Query for view CLIMATE_1_HR1

```
select
 SchoolId,
 SolarDay as Day,
 MAX(Cover*Type) as Clouds
from
  (select
    A.SchoolId,
    A.SiteId,
    A.SolarDay,
    DECODE(CloudCover,0,0,null,0,1) as Cover,
    DECODE(Cirrus,null,0,1) as Type
   from
    GLOBE.CLOUD_OBS A,
    GLOBE.SITE_LOCATION B
```

```
   where
    A.SolarDay >= TO_DATE('19980601','YYYYMMDD') AND
    A.UTtimeReported >= TO_DATE('19980601','YYYYMMDD') AND
    A.SchoolId = B.SchoolId AND
    A.SiteId = B.SiteId AND
    B.DataSource IN (1,2)
   )
Group By SchoolId, SolarDay
```

670 View CLIMATE_1_HR2

670.1 SQL Query for view CLIMATE_1_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 SUM(Clouds) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_1_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 SUM(Clouds)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 SUM(Clouds) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_1_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 SUM(Clouds)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 SUM(Clouds) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_1_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 SUM(Clouds)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId,
```

```
 SUM(Clouds) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_1_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 SUM(Clouds)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId,
 SUM(Clouds) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_1_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 SUM(Clouds)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 SUM(Clouds) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_1_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 SUM(Clouds)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

671 View CLIMATE_2_HR1

671.1 SQL Query for view CLIMATE_2_HR1

```
Select
 SchoolId,
 SolarDay as Day,
 MAX(MaxT*MinT) as Temp
From
(Select
 A.SchoolId,
 A.SolarDay,
 DECODE(MaxTemp,null,0,1) as MaxT,
 DECODE(MinTemp,null,0,1) as MinT
From
 GLOBE.AIR_TEMP A,
 GLOBE.SITE_LOCATION B
Where
 A.SolarDay >= TO_DATE('19980601','YYYYMMDD') AND
 A.UTtimeReported >= TO_DATE('19980601','YYYYMMDD') AND
 A.SchoolId = B.SchoolId AND
```

```
 A.SiteId = B.SiteId AND
 B.DataSource IN (1,2)
)
Group By SchoolId, SolarDay
```

672 View CLIMATE_2_HR2

672.1 SQL Query for view CLIMATE_2_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 SUM(Temp) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_2_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 SUM(Temp)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 SUM(Temp) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_2_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 SUM(Temp)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 SUM(Temp) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_2_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 SUM(Temp)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId,
 SUM(Temp) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_2_HR1 A
```

```
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 SUM(Temp)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId,
 SUM(Temp) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_2_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 SUM(Temp)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 SUM(Temp) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_2_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 SUM(Temp)>=(0.7)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

673 View CLIMATE_3_HR1

673.1 SQL Query for view CLIMATE_3_HR1

```
Select
 SchoolId,
 Month as Day,
 MAX(LEAST(GREATEST(SumRDays, SumSDays, SumREDays), 31)) as Precip
from
(select
  SchoolId,
  SiteId,
  Month,
  SumRDays,
  0 AS SumSDays,
  0 AS SumREDays
 From
  GLOBE.PRECIP_RAIN_CNT
 Where
 Month >= TO_DATE('19980601','YYYYMMDD')
UNION ALL
 select
  SchoolId,
  SiteId,
  Month,
```

```
  0 AS SumRDays,
  SumSDays,
  SumREDays
 From GLOBE.PRECIP_SOLID_CNT
 Where
 Month >= TO_DATE('19980601','YYYYMMDD')
)
Group By SchoolId, Month
```

674 View CLIMATE_3_HR2

674.1 SQL Query for view CLIMATE_3_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 SUM(Precip) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_3_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 SUM(Precip)>=(0.95)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 SUM(Precip) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_3_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 SUM(Precip)>=(0.95)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 SUM(Precip) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_3_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 SUM(Precip)>=(0.95)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId,
```

```
 SUM(Precip) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_3_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 SUM(Precip)>=(0.95)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId,
 SUM(Precip) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_3_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 SUM(Precip)>=(0.95)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 SUM(Precip) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_3_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 SUM(Precip)>=(0.95)*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

675 View CLIMATE_4_HR1

675.1 SQL Query for view CLIMATE_4_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') AS Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(WaterTemp) AS WTempCnt,
 Count(Acidity) AS WpHCnt,
 COUNT(TurbiditySecchi) AS SecCnt,
 COUNT(TurbidityTube) AS TubeCnt,
 COUNT(Conductivity) AS ConCnt,
 COUNT(Alkalinity) AS AlkCnt,
 COUNT(DissolvedOxy) AS OxyCnt,
 COUNT(HydAvgSalinity) AS HydAvgSalCnt,
 COUNT(TitrSalinity) AS TitrSalCnt,
 COUNT((Nitritenitrate - NitriteOnly)) AS NO3Cnt
from GLOBE.SURFACE_WATER
where
```

```
 DayMeasured >= TO_DATE('19980601','YYYYMMDD') AND
 DayMeasured < TRUNC(SYSDATE,'MM') + 6
group by
 SchoolId, SiteId, TRUNC(DayMeasured,'WW')
Having
 TRUNC(DayMeasured,'WW') >= TO_DATE('19980601','YYYYMMDD')
 AND TRUNC(DayMeasured,'WW') < TRUNC(SYSDATE,'MM')
```

676 View CLIMATE_4_HR2

676.1 SQL Query for view CLIMATE_4_HR2

```
select DISTINCT
 SchoolId,
 Week as Day,
 DECODE(SUM(SecCnt+A.TubeCnt),0,0,1)+DECODE(SUM(OxyCnt),0,0,1)+DECODE(SUM(AlkCnt),0,0,1)
    +DECODE(SUM(NO3Cnt),0,0,1)+DECODE(SUM(HydAvgSalCnt+TitrSalCnt+ConCnt),0,0,1)
    +DECODE(SUM(WTempCnt),0,0,1)+DECODE(SUM(WpHCnt),0,0,1) as ParmCnt
from CLIMATE_4_HR1 A
Group By SchoolId, Week
```

677 View CLIMATE_4_HR3_2parms

677.1 SQL Query for view CLIMATE_4_HR3_2parms

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 2
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 2
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
```

```
Where ParmCnt >= 2
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 2
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 2
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 2
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

678 View CLIMATE_4_HR3_4parms

678.1 SQL Query for view CLIMATE_4_HR3_4parms

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 4
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 4
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 4
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 4
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
```

```
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 4
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 4
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

679 View CLIMATE_4_HR3_6parms

679.1 SQL Query for view CLIMATE_4_HR3_6parms

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 6
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
```

```
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 6
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 6
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 6
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId,
COUNT(Day) as WeekCnt,
MIN(TRUNC(Day)) as FirstDay,
MAX(TRUNC(Day)) as LastDay
From CLIMATE_4_HR2 A
Where ParmCnt >= 6
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
COUNT(Day)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

```
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as WeekCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_4_HR2 A
 Where ParmCnt >= 6
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 COUNT(Day)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

680 View CLIMATE_4_HR4

680.1 SQL Query for view CLIMATE_4_HR4

```
Select DISTINCT
 A.SchoolId,
 A.Month as Day
From GLOBE.SURFACE_WATER_CNT A, GLOBE.SITE_LOCATION B
Where
 A.SchoolId=B.SchoolID AND
 A.SiteId=B.SiteId AND
 B.DataSource IN (1,2) AND
 A.Month >= TO_DATE('19980601','YYYYMMDD') AND
 A.Month < TRUNC(SYSDATE,'MM')
```

681 View CLIMATE_4_HR5

681.1 SQL Query for view CLIMATE_4_HR5

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId
 From CLIMATE_4_HR4 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 COUNT(Day) = 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId
 From CLIMATE_4_HR4 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 COUNT(Day) = 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
```

```
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId
 From CLIMATE_4_HR4 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 COUNT(Day) = 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId
 From CLIMATE_4_HR4 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 COUNT(Day) = 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId
 From CLIMATE_4_HR4 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 COUNT(Day) = 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId
 From CLIMATE_4_HR4 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 COUNT(Day) = 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

682 View CLIMATE_5_HR1

682.1 SQL Query for view CLIMATE_5_HR1

```
Select
 SchoolId,
 DayMeasured as Day
From
(Select DISTINCT
```

```
 A.SchoolId,
 A.DayMeasured,
 A.Depth
From
 GLOBE.SMGRAV_WS A,
 GLOBE.SITE_LOCATION B,
 SOILCHAR_HR1 C
Where
 A.DayMeasured >= TO_DATE('19980601','YYYYMMDD') AND
 A.SchoolId = B.SchoolId AND
 A.SiteId = B.SiteId AND
 A.SchoolId = C.SchoolId AND
 A.SiteId = C.SiteId AND
 B.DataSource IN (1,2)
)
Group By SchoolId, DayMeasured
Having COUNT(Depth) >= 2
```

683 View CLIMATE_5_HR2

683.1 SQL Query for view CLIMATE_5_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_5_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 COUNT(Day)>=12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_5_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 COUNT(Day)>=12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_5_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 COUNT(Day)>=12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
```

```
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_5_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 COUNT(Day)>=12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_5_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 COUNT(Day)>=12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 COUNT(Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_5_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 COUNT(Day)>=12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

684 View CLIMATE_6_HR1

684.1 SQL Query for view CLIMATE_6_HR1

```
Select
 SchoolId,
 SiteId,
 DayMeasured as Day
From
(Select
 A.SchoolId,
 A.SiteId,
```

```
 A.DayMeasured,
 A.Depth
From
 GLOBE.SMGYPS_WS A,
 GLOBE.SITE_LOCATION B,
 SOILCHAR_HR1 C
Where
 A.DayMeasured >= TO_DATE('19980601','YYYYMMDD') AND
 A.SchoolId = B.SchoolId AND
 A.SiteId = B.SiteId AND
 A.SchoolId = C.SchoolId AND
 A.SiteId = C.SiteId AND
 B.DataSource IN (1,2)
)
Group By SchoolId, SiteId, DayMeasured
Having COUNT(Depth) >= 3
```

685 View CLIMATE_6_HR2

685.1 SQL Query for view CLIMATE_6_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 COUNT(DISTINCT Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_6_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 COUNT(DISTINCT Day)>=0.7*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 COUNT(DISTINCT Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_6_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 COUNT(DISTINCT Day)>=0.7*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 COUNT(DISTINCT Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_6_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 COUNT(DISTINCT Day)>=0.7*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId,
 COUNT(DISTINCT Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_6_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 COUNT(DISTINCT Day)>=0.7*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId,
 COUNT(DISTINCT Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_6_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 COUNT(DISTINCT Day)>=0.7*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId,
 COUNT(DISTINCT Day) as DayCnt,
 MIN(TRUNC(Day)) as FirstDay,
 MAX(TRUNC(Day)) as LastDay
 From CLIMATE_6_HR1 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 COUNT(DISTINCT Day)>=0.7*365 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

686 View CLIMATE_7_HR1

686.1 SQL Query for view CLIMATE_7_HR1

```
Select
 A.SchoolId,
 A.SiteId,
 TRUNC(A.SolarDayMeas,'WW') as Week,
 COUNT(A.Depth) as DepthCnt,
 B.DataSource
From
 GLOBE.SOILTEMP_DAILYAVG A,
 GLOBE.SITE_LOCATION B
```

```
Where
 A.SchoolId = B.SchoolId AND
 A.SiteId = B.SiteId AND
 A.SolarDayMeas >= TO_DATE('19980601','YYYYMMDD')
Group By
 A.SchoolId, A.SiteId, TRUNC(A.SolarDayMeas,'WW'), B.DataSource
Having
 COUNT(A.Depth)>=2
```

687 View CLIMATE_7_HR2

687.1 SQL Query for view CLIMATE_7_HR2

```
Select DISTINCT
 SchoolId,
 Week as Day
From
 CLIMATE_7_HR1
Where
 DataSource IN (1,2)
```

688 View CLIMATE_7_HR3

688.1 SQL Query for view CLIMATE_7_HR3

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 Count(*) as WeekCnt,
 MIN(TRUNC(Day)) as FirstWeek,
 MAX(TRUNC(Day)) as LastWeek
 From CLIMATE_7_HR2 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 Count(*)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 Count(*) as WeekCnt,
 MIN(TRUNC(Day)) as FirstWeek,
 MAX(TRUNC(Day)) as LastWeek
 From CLIMATE_7_HR2 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 Count(*)>=(0.75)*52 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 Count(*) as WeekCnt,
 MIN(TRUNC(Day)) as FirstWeek,
 MAX(TRUNC(Day)) as LastWeek
 From CLIMATE_7_HR2 A
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
```

```
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
HAVING
Count(*)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId,
Count(*) as WeekCnt,
MIN(TRUNC(Day)) as FirstWeek,
MAX(TRUNC(Day)) as LastWeek
From CLIMATE_7_HR2 A
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
Count(*)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId,
Count(*) as WeekCnt,
MIN(TRUNC(Day)) as FirstWeek,
MAX(TRUNC(Day)) as LastWeek
From CLIMATE_7_HR2 A
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
Count(*)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
A.SchoolId,
Count(*) as WeekCnt,
MIN(TRUNC(Day)) as FirstWeek,
MAX(TRUNC(Day)) as LastWeek
From CLIMATE_7_HR2 A
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
HAVING
Count(*)>=(0.75)*52 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

689 View CLIMATE_7_HR4

689.1 SQL Query for view CLIMATE_7_HR4

```
Select
 A.SchoolId,
 A.SiteId,
```

```
 A.SolarDayMeas as Day,
 Count(DISTINCT A.Depth) as DepthCnt,
 Count(A.SampleNbr) as SampleCnt,
 B.DataSource
From
 GLOBE.SOILTEMP_DIURNAL A,
 GLOBE.SITE_LOCATION B
Where
 A.SchoolId = B.SchoolId AND
 A.SiteId = B.SiteId AND
 A.SolarDayMeas >= TO_DATE('19980601','YYYYMMDD')
Group By
 A.SchoolId, A.SiteId, A.SolarDayMeas, B.DataSource
Having
 Count(Distinct A.Depth) >= 2 AND
 Count(A.SampleNbr) >= 8
```

690 View CLIMATE_7_HR5

690.1 SQL Query for view CLIMATE_7_HR5

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId,
 Count(*) as DiurnalCnt,
 MIN(TRUNC(Day)) as FirstDiurnal,
 MAX(TRUNC(Day)) as LastDiurnal
 From CLIMATE_7_HR4 A
 Where DataSource IN (1,2)
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 Count(*)>=3 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId,
 Count(*) as DiurnalCnt,
 MIN(TRUNC(Day)) as FirstDiurnal,
 MAX(TRUNC(Day)) as LastDiurnal
 From CLIMATE_7_HR4 A
 Where DataSource IN (1,2)
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 Count(*)>=3 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId,
 Count(*) as DiurnalCnt,
 MIN(TRUNC(Day)) as FirstDiurnal,
 MAX(TRUNC(Day)) as LastDiurnal
 From CLIMATE_7_HR4 A
 Where DataSource IN (1,2)
 Group By A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 Count(*)>=3 AND
```

```
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId,
Count(*) as DiurnalCnt,
MIN(TRUNC(Day)) as FirstDiurnal,
MAX(TRUNC(Day)) as LastDiurnal
From CLIMATE_7_HR4 A
Where DataSource IN (1,2)
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
Count(*)>=3 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId,
Count(*) as DiurnalCnt,
MIN(TRUNC(Day)) as FirstDiurnal,
MAX(TRUNC(Day)) as LastDiurnal
From CLIMATE_7_HR4 A
Where DataSource IN (1,2)
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
Count(*)>=3 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
A.SchoolId,
Count(*) as DiurnalCnt,
MIN(TRUNC(Day)) as FirstDiurnal,
MAX(TRUNC(Day)) as LastDiurnal
From CLIMATE_7_HR4 A
Where DataSource IN (1,2)
Group By A.SchoolId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
HAVING
Count(*)>=3 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

691 View CLIMATE_7_HR6

691.1 SQL Query for view CLIMATE_7_HR6

```
Select
 A.HRnbr,
 A.SchoolId,
```

```
 A.WeekCnt,
 B.DiurnalCnt
From
 CLIMATE_7_HR3 A,
 CLIMATE_7_HR5 B
Where
 A.HRnbr = B.HRnbr AND
 A.SchoolId = B.SchoolId
```

692 View CLIMATE_8_HR1

692.1 SQL Query for view CLIMATE_8_HR1

```
select
```

693 View CLIMATE_ALL_HR0

693.1 SQL Query for view CLIMATE_ALL_HR0

```
Select HRnbr, SchoolId, 1 as ItemCnt, 'Clouds' as Item
 From CLIMATE_1_HR2
UNION
 Select HRnbr, SchoolId, 1 as ItemCnt, 'Temp' as Item
 From CLIMATE_2_HR2
UNION
 Select HRnbr, SchoolId, 1 as ItemCnt, 'Precip' as Item
 From CLIMATE_3_HR2
UNION
 Select A.HRnbr, A.SchoolId, 1 as ItemCnt, 'SW_2parms' as Item
 From CLIMATE_4_HR3_2parms A, CLIMATE_4_HR5 B
 Where A.HRnbr=B.HRnbr AND A.SchoolId=B.SchoolId
UNION
 Select A.HRnbr, A.SchoolId, 1 as ItemCnt, 'SW_4parms' as Item
 From CLIMATE_4_HR3_4parms A, CLIMATE_4_HR5 B
 Where A.HRnbr=B.HRnbr AND A.SchoolId=B.SchoolId
UNION
 Select A.HRnbr, A.SchoolId, 1 as ItemCnt, 'SW_6parms' as Item
 From CLIMATE_4_HR3_6parms A, CLIMATE_4_HR5 B
 Where A.HRnbr=B.HRnbr AND A.SchoolId=B.SchoolId
UNION
 Select HRnbr, SchoolId, 1 as ItemCnt, 'SM_Grav' as Item
 From CLIMATE_5_HR2
UNION
 Select HRnbr, SchoolId, 1 as ItemCnt, 'SM_Gypsum' as Item
 From CLIMATE_6_HR2
UNION
 Select HRnbr, SchoolId, 1 as ItemCnt, 'SoilTemp' as Item
 From CLIMATE_7_HR6
```

694 View CLIMATE_ALL_HR1

694.1 SQL Query for view CLIMATE_ALL_HR1

```
select
 HRnbr,
 SchoolId,
 SUM(ItemCnt) as ItemCnt
from CLIMATE_ALL_HR0
having SUM ( ItemCnt )  > = 3
group by HRNbr, SchoolId
```

695 View CLIMATE_ALL_HR2

695.1 SQL Query for view CLIMATE_ALL_HR2

```
select
 A.HRnbr,
 A.SchoolId,
 A.ItemCnt, 1 as DixonCnt,
 TRUNC(TO_DATE('19980601','YYYYMMDD')+2*(A.HRnbr-1)*31,'MM') as BeginMonth,
```

```
 TRUNC(TO_DATE('19980601','YYYYMMDD')+(2*(A.HRnbr+5)-1)*31,'MM') as EndMonth
from CLIMATE_ALL_HR1 A
```

696 View CLOUDS_HR0

696.1 SQL Query for view CLOUDS_HR0

```
select SchoolId, SiteId, SolarDay, COUNT(DECODE(CloudCover,9,NULL,1)) as CCovCnt, COUNT(Cirrus) as
CTypCnt, DECODE(COUNT(DECODE(CloudCover,9,NULL,1)*Cirrus),null,0,0,0,1) as HRdays
from GLOBE.CLOUD_OBS
where SolarDay >= TO_DATE  (
'1999
02
01','YYYY
MM
DD'
)  And  UTtimeRep or ted >= TO_DATE  (
'1999
02
01','YYYY
MM
DD'
)  And  SolarDay  <  TRUNC  (
SYSDATE,'MM'
)
group by SchoolId, SiteId, SolarDay
```

697 View CLOUDS_HR1

697.1 SQL Query for view CLOUDS_HR1

```
select
SchoolId,
SiteId,
TRUNC(SolarDay,'MM') AS Month,
SUM(CCovCnt) + SUM(CTypCnt) AS DixonCnt,
SUM(HRdays) AS HRdays
from CLOUDS_HR0
group by SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

698 View CLOUDS_HR2

698.1 SQL Query for view CLOUDS_HR2

```
select
2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MMDD'))/4,1)) +1 as HRnbr,
T.SchoolId, T.SiteId, MIN(T.Month) as BeginMonth,
MAX(T.Month) as EndMonth,
'CO' as Scope,
SUM(T.DixonCnt) as DixonCnt,
SUM(T.HRDays) as HRDays
FROM CLOUDS_HR1 T
Group By T.SchoolId, T.SiteId,  2*TRUNC (Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY
MMDD'))/4,1)) +1
HAVING MIN(T.Month) < TRUNC (TRUNC(SYSDATE,'MM') - 2*33,'MM') AND SUM (T.HRdays) > (0.7) * (120)
ALL
select
2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MMDD'))+2)/4,1)) as HRnbr,
T.SchoolId, T.SiteId, MIN(T.Month) as BeginMonth, MAX(T.Month) as EndMonth,
'CO' as Scope, SUM(T.DixonCnt) as DixonCnt, SUM(T.HRDays) as HRDays
FROM CLOUDS_HR1 T
Group By  T.SchoolId, T.SiteId,  2*TRUNC (Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY
MMDD'))+2)/4,1))
HAVING MIN(T.Month) > TO_DATE ('1999 02 01','YYYY MM DD') AND SUM (T.HRdays) > (0.7) * (120)
```

699 View CLOUDS_HR3

699.1 SQL Query for view CLOUDS_HR3

```
select CLOUDS_HR2.*, DECODE(SITE_LOCATION.DataSource,0,'Unchecked',1,'GPS',2,'Other') as LocMethod
from CLOUDS_HR2, GLOBE.SITE_LOCATION
where CLOUDS_HR2.SchoolID = SITE_LOCATION.SCHOOLID  AND  CLOUDS_HR2.SiteID  =  SITE_LOCATION.SITEID
```

700 View CLOUDS_HR4

700.1 SQL Query for view CLOUDS_HR4

```
select HRnbr, SchoolId, SUM(DixonCnt) as DixonCnt,
TRUNC(TO_DATE('19990201','YYYYMMDD')+ 2*(HRnbr-1)*31,'MM') as BeginMonth,
TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM') as EndMonth
from CLOUDS_HR3
where LocMethod = 'GPS'
or LocMethod = 'Other'
group by HRnbr, SchoolId,
TRUNC(TO_DATE('19990201','YYYYMMDD')+2*(HRnbr-1)*31,'MM'),
TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM')
```

701 View CLOUD_OBS_DR1

702 View CLOUD_OBS_DR2

702.1 SQL Query for view CLOUD_OBS_DR2

```
select CLOUD_OBS_DR1.*, (ccovcnt+ctypcnt) as DixonCnt
from CLOUD_OBS_DR1
```

703 View CLOUD_OBS_DU1

704 View CLOUD_OBS_DU2

704.1 SQL Query for view CLOUD_OBS_DU2

```
select CLOUD_OBS_DU1.*, (ccovcnt+ctypcnt) as DixonCnt
from CLOUD_OBS_DU1
```

705 View CLOUD_OBS_MC1

705.1 SQL Query for view CLOUD_OBS_MC1

```
select 'CO' as MeasureCode, COUNT(*) as RowCnt, COUNT(DECODE(CloudCover,9,NULL,1)) as CCovCnt,
COUNT(Cirrus) as CTypCnt
from GLOBE.CLOUD_OBS
```

706 View CLOUD_OBS_MC2

706.1 SQL Query for view CLOUD_OBS_MC2

```
select CLOUD_OBS_MC1.*, (ccovcnt+ctypcnt) as DixonCnt
from CLOUD_OBS_MC1
```

707 View CLOUD_OBS_NOON

707.1 SQL Query for view CLOUD_OBS_NOON

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay,'MM') AS Month,
 'CO' AS MeasureCode,
 COUNT(DECODE(CloudCover,9,null,null,null,1)) AS NCCovCnt,
 COUNT(Cirrus) AS NCTypCnt,
 ROUND(AVG(DECODE(Cloudcover,0,5,1,30,2,70,3,95,4,0,5,5,6,17.5,7,37.5,NULL)),1) AS NCCovAvg,
```

```
 COUNT(DECODE(CloudCover,8,1,null)*DECODE(Obscured,0,null,null,null,1)) AS NObscureCnt,
 SUM(Cirrus) AS NCiSum,
 SUM(CirroCumulus) AS NCCSum,
 SUM(CirroStratus) AS NCSSum,
 SUM(AltoStratus) AS NASSum,
 SUM(AltoCumulus) AS NACSum,
 SUM(Cumulus) AS NCuSum,
 SUM(NimboStratus) AS NNSSum,
 SUM(Stratus) AS NStSum,
 SUM(StratoCumulus) AS NSCSum,
 SUM(CumuloNimbus) AS NCNSum,
 MIN(UTTimeMeasured) AS NBegDate,
 MAX(UTTimeMeasured) AS NEndDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS NLagTimeAvg
from GLOBE.CLOUD_OBS
where
 UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24 AND
 SolarNoonDiff < 5400 AND
 SolarNoonDiff > -5400
group by SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

708 View CLOUD_OBS_RPRT_VU

708.1 SQL Query for view CLOUD_OBS_RPRT_VU

```
select Distinct as SchoolID, 'CO' as MeasureCode, Min(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from CLOUD_OBS
group by SchoolID
```

709 View CLOUD_OBS_VU1

709.1 SQL Query for view CLOUD_OBS_VU1

```
select SchoolId, SiteId, TRUNC(SolarDay,'MM') AS Month, 'CO' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(DECODE(CloudCover,9,null,null,null,1)) AS CCovCnt,
 COUNT(Cirrus) AS CTypCnt,
 ROUND(AVG(DECODE(Cloudcover,0,5,1,30,2,70,3,95,4,0,5,5,6,18,7,38,NULL)),1) AS CCovAvg,
 COUNT(DECODE(CloudCover,8,1,null)*DECODE(Obscured,0,null,null,null,1)) AS ObscureCnt,
 SUM(Cirrus) AS CiSum, SUM(CirroCumulus) AS CCSum, SUM(CirroStratus) AS CSSum,
 SUM(AltoStratus) AS ASSum, SUM(AltoCumulus) AS ACSum, SUM(Cumulus) AS CuSum,
 SUM(NimboStratus) AS NSSum, SUM(Stratus) AS StSum, SUM(StratoCumulus) AS SCSum,
 SUM(CumuloNimbus) AS CNSum,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.CLOUD_OBS
where
 UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

710 View CLOUD_OBS_VU2

710.1 SQL Query for view CLOUD_OBS_VU2

```
select
 A.*,
 (A.CCovCnt+A.CTypCnt+A.ObscureCnt) AS DixonCnt,
 B.NCCovCnt,
 B.NCTypCnt,
 B.NCCovAvg,
 B.ObscureCnt,
 B.NCiSum,
 B.NCCSum,
 B.NCSSum,
 B.NASSum,
 B.NACSum,
 B.NCuSum,
 B.NNSSum,
 B.NStSum,
```

```
 B.NSCSum,
 B.NCNSum,
 B.NBegDate,
 B.NEndDate
from CLOUD_OBS_VU1 A, CLOUD_OBS_NOON B
where
 A.SchoolId = B.SchoolId ( + ) AND
 A.SiteId = B.SiteId ( + ) AND
 A.Month = B.Month ( + )
```

711 View CURRENT_TEMP_NONNOON_VU1

711.1 SQL Query for view CURRENT_TEMP_NONNOON_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(UTTimeMeasured,'MM') Month,
  'AT' as MeasureCode,
  COUNT(*) RowCnt,
  COUNT(currenttemp) NNCurTCnt,
  MIN(CurrentTemp) as NNCurTMin,
  MAX(CurrentTemp) as NNCurTMax,
  ROUND(AVG(CurrentTemp),1) as NNCurTAvg,
  ROUND(STDDEV(CurrentTemp),1) as NNCurTStd,
  MIN(UTTimeMeasured) AS BegDate,
  MAX(UTTimeMeasured) AS EndDate,
  MIN(UTTimereported) AS BegRDate,
  MAX(UTTimeReported) AS EndRDate,
  ROUND(AVG(UTTimeReported-UTTimeMeasured),2) AS LagTimeAvg
from
  CURRENT_TEMP_NONNOON
where
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
group by
  SchoolId, SiteId, TRUNC(UTTimeMeasured,'MM')
```

712 View CURRENT_TEMP_NONNOON_VU2

712.1 SQL Query for view CURRENT_TEMP_NONNOON_VU2

```
select
  CURRENT_TEMP_NONNOON_VU1.*,
  nncurtcnt as dixoncnt
from
 CURRENT_TEMP_NONNOON_VU1
```

713 View Canopy_HR1

713.1 SQL Query for view Canopy_HR1

```
select
 TB.SchoolId,
 TB.SiteId,
 DayMeasured as Day
from
 GLOBE.TREE_BIOMETRY TB,
 GLOBE.SITE_LOCATION SL
where
 TB.SchoolId = SL.SchoolId AND
 TB.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 TB.DayMeasured >= TO_DATE('20010201','YYYYMMDD') AND
 TB.CanopyPlus is not null AND
 TB.CanopyMinus is not null
```

714 View Canopy_HR2

714.1 SQL Query for view Canopy_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from Canopy_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from Canopy_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from Canopy_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from Canopy_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from Canopy_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

```
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from Canopy_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

715 View Canopy_HR3

715.1 SQL Query for view Canopy_HR3

```
select
 HRnbr,
 SchoolId
from
 Canopy_HR2
group by HRnbr, SchoolId
having count(*) >= 2
```

716 View CloudCover_HR1

716.1 SQL Query for view CloudCover_HR1

```
select distinct
 CC.SchoolId,
 CC.SiteId,
 SolarDay as DayMeasured
from
 GLOBE.CLOUD_OBS CC,
 GLOBE.SITE_LOCATION SL
where
 CC.SchoolId = SL.SchoolId AND
 CC.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 CC.SolarDay >= TO_DATE('20010201','YYYYMMDD') AND
 CC.CloudCover > 0
```

717 View CloudCover_HR2

717.1 SQL Query for view CloudCover_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.DayMeasured,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 Count(T.DayMeasured) as GoodDays
from
 CloudCover_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.DayMeasured) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 Count(T.DayMeasured) >= 4*30*0.7
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 Count(T.DayMeasured) as GoodDays
from
 CloudCover_HR1 T
```

```
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.DayMeasured) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 Count(T.DayMeasured) >= 4*30*0.7
```

718 View DAILY_DIXON_OBS_VU

718.1 SQL Query for view DAILY_DIXON_OBS_VU

```
select (SELECTDAY,MeasureCode,DixonCntfrom AIR_TEMP_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PRECIP_RAIN_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PRECIP_SOLID_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom CLOUD_OBS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom GRASS_BIOMETRY_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom TREE_BIOMETRY_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom LAND_COVER_DU2)

union
select ALL(SELECTDAY,'MD' as MeasureCode,ExtCnt as DixonCntfrom LAND_COVER_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom GRASSBIOM_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom TREEBIOM_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PHENOLOGY_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PHENOLOGY_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PHENOLOGY_SITE_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom ATMOSPHERE_SITE_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOILMOISTURE_SITE_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOILCHAR_SITE_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SURFACEWATER_SITE_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SITE_PROXIMITY_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SITE_LOCATION_DU2)

union
select ALL(SELECTDAY,'LC' as MeasureCode,(QualSPCnt+QuanSPCnt+BISPCnt) as DixonCntFROM
SITE_PHOTO_DU2WHERE (QualSPCnt+QuanSPCnt+BISPCnt)>0)

union
select ALL(SELECTDAY,'MD' as MeasureCode,(ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt) as
DixonCntFROM SITE_PHOTO_DU2WHERE (ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt)>0)
```

```
union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOIL_CHAR_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom DENSITY_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PARTICLE96_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PARTICLE_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom FERTILITY_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom ACIDITY_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOIL_INFILTRATION_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom INFILTRATION_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SMGYPS_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SMGRAV_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOIL_TEMP_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom TURBIDITY_WS_DU2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SURFACE_WATER_DU2)
```

719 View DAILY_DIXON_REP_VU

719.1 SQL Query for view DAILY_DIXON_REP_VU

```
select (SELECTDAY,MeasureCode,DixonCntfrom AIR_TEMP_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PRECIP_RAIN_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PRECIP_SOLID_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom CLOUD_OBS_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom GRASS_BIOMETRY_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom TREE_BIOMETRY_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom LAND_COVER_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,'MD' as MeasureCode,ExtCnt as DixonCntfrom LAND_COVER_DR2where ExtCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom GRASSBIOM_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom TREEBIOM_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PHENOLOGY_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PHENOLOGY_WS_DR2)
```

```
union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PHENOLOGY_SITE_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom ATMOSPHERE_SITE_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOILMOISTURE_SITE_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOILCHAR_SITE_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SURFACEWATER_SITE_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SITE_PROXIMITY_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SITE_LOCATION_DR2where dixonCnt>0 and SchoolId NOT LIKE
'ZZ%')

union
select ALL(SELECTDAY,'LC' as MeasureCode,(QualSPCnt+QuanSPCnt+BISPCnt) as DixonCntFROM
SITE_PHOTO_DR2WHERE (QualSPCnt+QuanSPCnt+BISPCnt)>0)

union
select ALL(SELECTDAY,'MD' as MeasureCode,(ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt) as
DixonCntFROM SITE_PHOTO_DR2WHERE (ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt)>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOIL_CHAR_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom DENSITY_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PARTICLE96_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom PARTICLE_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom FERTILITY_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom ACIDITY_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOIL_INFILTRATION_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom INFILTRATION_WS_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SMGYPS_WS_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SMGRAV_WS_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SOIL_TEMP_DR2)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom TURBIDITY_WS_DR2where dixonCnt>0)

union
select ALL(SELECTDAY,MeasureCode,DixonCntfrom SURFACE_WATER_DR2)
```

720 View DENSITY_WS_DR1

721 View DENSITY_WS_DR2

721.1 SQL Query for view DENSITY_WS_DR2

```
select DENSITY_WS_DR1.*, (6*BDenSampCnt+2*BDAugCnt) as DixonCnt
from DENSITY_WS_DR1
```

722 View DENSITY_WS_DU1

723 View DENSITY_WS_DU2

723.1 SQL Query for view DENSITY_WS_DU2

```
select DENSITY_WS_DU1.*, (6*BDenSampCnt+2*BDAugCnt) as DixonCnt
from DENSITY_WS_DU1
```

724 View DENSITY_WS_MC1

724.1 SQL Query for view DENSITY_WS_MC1

```
select 'SC' as MeasureCode, COUNT(W.BulkDensity) as BDenSampCnt,
COUNT(AugerTopDepth+AugerBottomDepth+AugerDiameter) as BDAugCnt
from GLOBE.DENSITY_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.H  or  izonNbr = S.H or izonNbr
```

725 View DENSITY_WS_MC2

725.1 SQL Query for view DENSITY_WS_MC2

```
select DENSITY_WS_MC1.*, (6*BDenSampCnt+2*BDAugCnt) as DixonCnt
from DENSITY_WS_MC1
```

726 View DENSITY_WS_RPRT_VU

726.1 SQL Query for view DENSITY_WS_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, Min(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from DENSITY_WS
group by SchoolID
```

727 View DENSITY_WS_VU1

727.1 SQL Query for view DENSITY_WS_VU1

```
select W.SchoolId as SchoolId, W.SiteId as SiteId,
 TRUNC(S.UTtimeMeasured,'MM') AS Month, 'SC' AS MeasureCode,
 COUNT(*) AS RowCnt, COUNT(W.BulkDensity) as BDenSampCnt,
 COUNT(AugerTopDepth+AugerBottomDepth+AugerDiameter) AS BDAugCnt,
 MIN(S.UTtimeMeasured) AS BegDate,
 MAX(S.UTtimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(W.UTtimeReported-S.UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.DENSITY_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.HorizonNbr = S.HorizonNbr and
W.UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by W.SchoolId, W.SiteId, TRUNC(S.UTtimeMeasured,'MM')
```

728 View DENSITY_WS_VU2

728.1 SQL Query for view DENSITY_WS_VU2

```
select DENSITY_WS_VU1.*, (6*BDenSampCnt+2*BDAugCnt) as DixonCnt
from DENSITY_WS_VU1
```

729 View DISSOLVED_OXYGEN_VU

729.1 SQL Query for view DISSOLVED_OXYGEN_VU

```
select
 SchoolId,
 SiteId,
 DayMeasured,
 WaterTemp,
 HydSalinity,
 TitrSalinity,
 DissolvedOxy,

ROUND(EXP((1.575701e05+(-6.642308e07+(1.243800e10-8.621949e11/(WaterTemp+273.15))/(WaterTemp+273.15))/
(WaterTemp+273.15))/(WaterTemp+273.15)-139.3441-NVL(HydSalinity,TitrSalinity)*(1.7674e-02+(-1.0754e01+
2.1407e03/(WaterTemp+273.15))/(WaterTemp+273.15))),2) as SaturatedDO
from GLOBE.SURFACE_WATER
where
 WaterTemp is not NULL AND
 DissolvedOxy is not NULL
```

730 View DIURNAL_LOG_HR0

730.1 SQL Query for view DIURNAL_LOG_HR0

```
select
  TL.SchoolId,
  TL.SiteId,
  TRUNC(TL.UTtimeMeasured) as SolarDay,
  Count(TL.UTtimeMeasured) as DiurnalCnt
from
  GLOBE.TEMPERATURE_LOGGER TL,
  GLOBE.SITE_LOCATION SL
where
  TL.SchoolId = SL.SchoolId AND
  TL.SiteId = SL.SiteId AND
  SL.DataSource in (1,2) AND
  TL.UTtimeMeasured >= TO_DATE('20000601','YYYYMMDD') AND
  TL.SoilTemp5cm is not null
group by TL.SchoolId, TL.SiteId, TRUNC(TL.UTtimeMeasured)
having Count(TL.UTtimeMeasured) >= 5
```

731 View DIURNAL_LOG_HR1

731.1 SQL Query for view DIURNAL_LOG_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(*) as DiurnalDays
from (select
        D1.SchoolId,
        D1.SiteId,
        D1.SolarDay
      from
        DIURNAL_LOG_HR0 D1,
        DIURNAL_LOG_HR0 D2
      where
        D1.SchoolId = D2.SchoolId AND
        D1.SolarDay + 1 = D2.SolarDay )
group by SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

732 View DIURNAL_LOG_HR2

732.1 SQL Query for view DIURNAL_LOG_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
```

```
 T.SchoolId,
 T.SiteId,
 SUM(T.DiurnalDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 DIURNAL_LOG_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 >= 1
 AND
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.DiurnalDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 DIURNAL_LOG_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) >= 1
 AND
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2)
```

733 View DIURNAL_TEMP_HR0

733.1 SQL Query for view DIURNAL_TEMP_HR0

```
select
 SchoolId,
 SiteId,
 SolarDay
from (select
       ST.SchoolId,
       ST.SiteId,
       TRUNC(SolarDayMeas) as SolarDay,
       Depth
      from
        GLOBE.SOIL_TEMP ST,
        GLOBE.SITE_LOCATION SL
      where
       ST.SchoolId = SL.SchoolId AND
       ST.SiteId = SL.SiteId AND
       SL.DataSource in (1,2) AND
       SolarDayMeas >= TO_DATE('20000601','YYYYMMDD') AND
       SoilTemp is not null
      group by ST.SchoolId, ST.SiteId, TRUNC(SolarDayMeas), Depth
      having count(*) >= 5 )
group by SchoolId, SiteId, SolarDay
```

734 View DIURNAL_TEMP_HR1

734.1 SQL Query for view DIURNAL_TEMP_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(*) as DiurnalDays
from (select
        D1.SchoolId,
        D1.SiteId,
        D1.SolarDay
      from
        DIURNAL_TEMP_HR0 D1,
```

```
        DIURNAL_TEMP_HR0 D2
      where
        D1.SchoolId = D2.SchoolId AND
        D1.SolarDay + 1 = D2.SolarDay )
group by SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

735 View DIURNAL_TEMP_HR2

735.1 SQL Query for view DIURNAL_TEMP_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.DiurnalDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 DIURNAL_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 >= 1
 AND
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.DiurnalDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 DIURNAL_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) >= 1
 AND
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month, TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2)
```

736 View DIXON_SUMMARY_VU1

736.1 SQL Query for view DIXON_SUMMARY_VU1

```
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  ACIDITY_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  AIR_TEMP_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  ATMOSPHERE_SITE_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  BUDBURST_META_CNT
where
```

```
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  CLOUD_OBS_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  DENSITY_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  FERTILITY_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  GRASS_BIOMETRY_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  GRASSBIOM_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  GREENDOWN_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  GREENING_METADATA_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  GREENUP_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  HAZE_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  HAZE_WS_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  HUMIDITY_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
```

```
  INFILTRATION_WS_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  LAND_COVER_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  LILAC_METADATA_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  LILAC_PHENOPHASE_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  OZONE_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  PARTICLE_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  PARTICLE96_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegDate AS BegRDate, EndDate AS EndRDate
from
  PHENOLOGY_SITE_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegDate AS BegRDate, EndDate AS EndRDate
from
  PHENOLOGY_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  PRECIP_RAIN_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  PRECIP_SOLID_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SITE_LOCATION_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, 'LC' as MeasureCode,
  (QualSPCnt+QuanSPCnt+BISPCnt) as DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SITE_PHOTO_CNT
```

```
where
  (QualSPCnt+QuanSPCnt+BISPCnt) > 0
UNION ALL
select SchoolId, SiteId, Month, 'MD' as MeasureCode,
  (ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt) as DixonCnt,
  BegDate, EndDate, BegRDate, EndRDate
from
  SITE_PHOTO_CNT
where
  (ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt) > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SITE_PROXIMITY_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SKY_OBSCURED_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SMGRAV_WS_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SMGYPS_WS_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SOIL_CHAR_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SOIL_INFILTRATION_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SOIL_TEMP_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SOILCHAR_SITE_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SOILMOISTURE_SITE_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SURFACE_WATER_CNT
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  SURFACEWATER_SITE_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
```

```
from
  TEMPERATURE_LOGGER_CNT
where
  DixonCnt >0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  TREE_BIOMETRY_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  TREEBIOM_WS_CNT
where
  DixonCnt > 0
UNION ALL
select SchoolId, SiteId, Month, MeasureCode, DixonCnt, BegDate, EndDate,
  BegRDate, EndRDate
from
  TURBIDITY_WS_CNT
where
  DixonCnt > 0
```

737 View DIXON_SUMMARY_VU2

737.1 SQL Query for view DIXON_SUMMARY_VU2

```
select
 SchoolId,
 SiteId,
 Month,
 MeasureCode,
 SUM(DixonCnt) AS DixonCnt,
 MIN(BegDate) as BegDate,
 MAX(EndDate) as EndDate,
 MIN(BegRDate) as BegRDate,
 MAX(EndRDate) as EndRDate
from
 DIXON_SUMMARY_VU1
group by
 SchoolId, SiteId, Month, MeasureCode
```

738 View DIXON_SUMMARY_VU3

738.1 SQL Query for view DIXON_SUMMARY_VU3

```
select
 SchoolId,
 Month,
 DECODE(MeasureCode,'AT','Atmo','PR','Atmo','PS','Atmo','CO','Atmo',
        'AZ','Atmo','RH','Atmo','OZ','Atmo',
        'SW','Hydr','MD','Meta','GP','Meta','PX','Meta','LC','LcBi',
        'TB','LcBi', 'GB','LcBi','SM','Soil','SC','Soil','SF','Soil',
        'ST','Soil','LB','Spec','PH','Spec','BB','Spec',
        'GN','Spec', '????') as GroupCode,
 SUM(DixonCnt) AS DixonCnt,
 MIN(BegDate) as BegDate,
 MAX(EndDate) as EndDate,
 MIN(BegRDate) as BegRDate,
 MAX(EndRDate) as EndRDate
from DIXON_SUMMARY_VU2
group by SchoolId, Month, DECODE(MeasureCode,'AT','Atmo','PR','Atmo',
                                 'PS','Atmo','CO','Atmo','AZ','Atmo',
                                 'RH','Atmo','OZ','Atmo','SW','Hydr',
                                 'MD','Meta','GP','Meta','PX','Meta',
                                 'LC','LcBi','TB','LcBi','GB','LcBi',
                                 'SM','Soil','SC','Soil','SF','Soil',
                                 'ST','Soil','LB','Spec','PH','Spec',
                                 'BB','Spec','GN','Spec','????')
```

739 View EARTH_SYSTEM_1_HR0

739.1 SQL Query for view EARTH_SYSTEM_1_HR0

```
select HRnbr, SchoolId, 'atmos' as Req
from ATMOS_TEMP_HR2
UNION
select HRnbr, SchoolId, 'water' as Req
from WATER_TEMP_HR2
UNION
select HRnbr, SchoolId, 'soil' as Req
from SOIL_TEMP_HR2
UNION
select HRnbr, SchoolId, 'diurnal' as Req
from DIURNAL_TEMP_HR2
UNION
select HRnbr, SchoolId, 'diurnal' as Req
from DIURNAL_LOG_HR2
```

740 View EARTH_SYSTEM_1_HR1

740.1 SQL Query for view EARTH_SYSTEM_1_HR1

```
select
 HRnbr,
 SchoolId
from
 EARTH_SYSTEM_1_HR0
group by HRnbr, SchoolId
having
 count(*) = 4 AND
 HRnbr >= 13
```

741 View EARTH_SYSTEM_2_HR0

741.1 SQL Query for view EARTH_SYSTEM_2_HR0

```
select HRnbr, SchoolId, 'precip' as Req
from GLOBE.Acidity_HR4
UNION
select HRnbr, SchoolId, 'surfwater' as Req
from PH_SURFWATER_HR3
UNION
select HRnbr, SchoolId, 'soil' as Req
from PH_SOIL_HR3
```

742 View EARTH_SYSTEM_2_HR1

742.1 SQL Query for view EARTH_SYSTEM_2_HR1

```
select
 HRnbr,
 SchoolId
from
 EARTH_SYSTEM_2_HR0
group by HRnbr, SchoolId
having count(*) = 3 AND
 HRnbr >= 13
```

743 View EARTH_SYSTEM_3_HR0

743.1 SQL Query for view EARTH_SYSTEM_3_HR0

```
select HRnbr, SchoolId, 'Nit' as Req
from Nitrate_HR2
UNION
select HRnbr, SchoolId, 'NPK' as Req
from SoilNPK_HR2
UNION
select HRnbr, SchoolId, 'GT' as Req
```

```
from GrassTrees_HR2
```

744 View EARTH_SYSTEM_3_HR1

744.1 SQL Query for view EARTH_SYSTEM_3_HR1

```
select HRnbr, SchoolId
from EARTH_SYSTEM_3_HR0
group by HRnbr, SchoolId
having count(*) = 3 AND
 HRnbr >= 13
```

745 View EARTH_SYSTEM_4_HR0

745.1 SQL Query for view EARTH_SYSTEM_4_HR0

```
select HRnbr, SchoolId, 'Turb' as Req
from Turbidity_HR2
UNION
select HRnbr, SchoolId, 'CloudCover' as Req
from CloudCover_HR2
UNION
select HRnbr, SchoolId, 'Canopy' as Req
from Canopy_HR3
```

746 View EARTH_SYSTEM_4_HR1

746.1 SQL Query for view EARTH_SYSTEM_4_HR1

```
select HRnbr, SchoolId
from EARTH_SYSTEM_4_HR0
group by HRnbr, SchoolId
having count(*) = 3 AND
 HRnbr >= 13
```

747 View EARTH_SYSTEM_5_HR0

747.1 SQL Query for view EARTH_SYSTEM_5_HR0

```
select
 BB.HRnbr,
 BB.SchoolId,
 BB.SiteId
from
 Budburst_HR2 BB,
 BBtemp_HR1 T,
 BBzPrecip_HR1 P
where
 BB.SchoolId = T.SchoolId AND
 BB.SiteId = T.SiteId AND
 BB.Day = T.Day AND
 BB.SchoolId = P.SchoolId AND
 BB.SiteId = P.SiteId AND
 BB.Day = P.Day
```

748 View EARTH_SYSTEM_5_HR1

748.1 SQL Query for view EARTH_SYSTEM_5_HR1

```
select distinct
 HRnbr,
 SchoolId
from
 EARTH_SYSTEM_5_HR0
where
 HRnbr >= 13
```

749 View EARTH_SYSTEM_6_HR1

749.1 SQL Query for view EARTH_SYSTEM_6_HR1

```
select distinct
 HRnbr,
 SchoolId
from Phenology_HR8
where HRnbr >= 13
```

750 View EARTH_SYSTEM_ALL_HR1

750.1 SQL Query for view EARTH_SYSTEM_ALL_HR1

```
select
 HRnbr,
 SchoolId,
 'ES_1' as OptionType
from
 EARTH_SYSTEM_1_HR1
UNION ALL
select
 HRnbr,
 SchoolId,
 'ES_2' as OptionType
from
 EARTH_SYSTEM_2_HR1
UNION ALL
select
 HRnbr,
 SchoolId,
 'ES_3' as OptionType
from
 EARTH_SYSTEM_3_HR1
UNION ALL
select
 HRnbr,
 SchoolId,
 'ES_4' as OptionType
from
 EARTH_SYSTEM_4_HR1
UNION ALL
select
 HRnbr,
 SchoolId,
 'ES_5' as OptionType
from
 EARTH_SYSTEM_5_HR1
UNION ALL
select
 HRnbr,
 SchoolId,
 'ES_6' as OptionType
from
 EARTH_SYSTEM_6_HR1
```

751 View EARTH_SYSTEM_ALL_HR2

751.1 SQL Query for view EARTH_SYSTEM_ALL_HR2

```
select distinct
 HRnbr,
 SchoolId,
 1 as DixonCnt,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+ 2*(HRnbr-1)*31,'MM') as BeginMonth,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM') as EndMonth
from EARTH_SYSTEM_ALL_HR1
```

752 View FERTILITY_WS_DR1

753 View FERTILITY_WS_DR2

753.1 SQL Query for view FERTILITY_WS_DR2

```
select FERTILITY_WS_DR1.*, (NSampCnt+PSampCnt+KSampCnt) as DixonCnt
from FERTILITY_WS_DR1
```

754 View FERTILITY_WS_DU1

755 View FERTILITY_WS_DU2

755.1 SQL Query for view FERTILITY_WS_DU2

```
select FERTILITY_WS_DU1.*, (NSampCnt+PSampCnt+KSampCnt) as DixonCnt
from FERTILITY_WS_DU1
```

756 View FERTILITY_WS_MC1

756.1 SQL Query for view FERTILITY_WS_MC1

```
select 'SC' as MeasureCode, COUNT(W.NitrogenEst) as NSampCnt, COUNT(W.PhosphorusEst) as PSampCnt,
COUNT(W.PotassiumEst) as KSampCnt
from GLOBE.FERTILITY_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.H  or  izonNbr = S.H or izonNbr
```

757 View FERTILITY_WS_MC2

757.1 SQL Query for view FERTILITY_WS_MC2

```
select FERTILITY_WS_MC1.*, (NSampCnt+PSampCnt+KSampCnt) as DixonCnt
from FERTILITY_WS_MC1
```

758 View FERTILITY_WS_RPRT_VU

758.1 SQL Query for view FERTILITY_WS_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from FERTILITY_WS
group by SchoolID
```

759 View FERTILITY_WS_VU1

759.1 SQL Query for view FERTILITY_WS_VU1

```
select
 W.SchoolId,
 W.SiteId,
 TRUNC(S.UTtimeMeasured,'MM') AS Month,
 'SC' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(W.NitrogenEst) AS NSampCnt,
 COUNT(W.PhosphorusEst) AS PSampCnt,
 COUNT(W.PotassiumEst) AS KSampCnt,
 MIN(S.UTtimeMeasured) AS BegDate,
 MAX(S.UTtimeMeasured) AS EndDate,
 MIN(S.UTtimeReported) AS BegRDate,
 MAX(S.UTtimeReported) AS EndRDate,
 ROUND(AVG(W.UTtimeReported-S.UTtimeMeasured),2) AS LagTimeAvg
from
 GLOBE.FERTILITY_WS W, GLOBE.SOIL_CHAR S
where
 W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.HorizonNbr = S.HorizonNbr and W.UTtimeReported
< TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 W.SchoolId, W.SiteId, TRUNC(S.UTtimeMeasured,'MM')
```

760 View FERTILITY_WS_VU2

760.1 SQL Query for view FERTILITY_WS_VU2

```
select FERTILITY_WS_VU1.*, (NSampCnt+PSampCnt+KSampCnt) as DixonCnt
from FERTILITY_WS_VU1
```

761 View FIPS_TO_COUNTY

761.1 SQL Query for view FIPS_TO_COUNTY

```
select DISTINCT
 FIPScounty,
 County
from GLOBE.ZIP_INFO
```

762 View FIRECOVER_SUMMARY

763 View FIREFUEL_SUMMARY

764 View FIREFUEL_VU1

765 View FIREFUEL_VU2

766 View FLOWERSPECIES_COUNTS_VU1

766.1 SQL Query for view FLOWERSPECIES_COUNTS_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(StartTimeUT,'MM') as Month,
  'HB' as MeasureCode,
  Count(*) as RowCnt,
  Count(AdultMaleCnt) as FAMCnt,
  Count(AdultFemaleCnt) as FAFCnt,
  Count(NoSexThroatCnt) as FNTCnt,
  Count(NoSexYoungCnt) as FNYCnt,
  Count(YoungMaleCnt) as FYMCnt,
  Count(UnusualCnt) as FUnCnt,
  MIN(StartTimeUT) as BegDate,
  MAX(StartTimeUT) as EndDate,
  MIN(UTtimereported) as BegRDate,
  MAX(UTtimereported) as EndRDate,
  ROUND(AVG(UTtimereported-StartTimeUT),2) as LagTimeAvg
from FLOWERSPECIES_COUNTS
where UTtimereported < TRUNC(SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(StartTimeUT,'MM')
```

767 View FLOWERSPECIES_COUNTS_VU2

767.1 SQL Query for view FLOWERSPECIES_COUNTS_VU2

```
select
  FLOWERSPECIES_COUNTS_VU1.*,
  (famcnt+fafcnt+fntcnt+fnycnt+fymcnt+funcnt) as dixoncnt
from FLOWERSPECIES_COUNTS_VU1
```

768 View GDS_MOISTCHG_30DAYS_VU

768.1 SQL Query for view GDS_MOISTCHG_30DAYS_VU

```
select Precip.Lat, Precip.Lon, Precip.RecentDay, Precip.GDS_30day, Precip.PET_30day,
Precip.Rain_30Day, Precip.Snow_30day, Precip.RainEqv_30day, Snowmelt.Snowmelt as Snowmelt_30day,
Precip.Rain_30Day+NVL(Precip.RainEqv_30day,0) +NVL (Snowmelt.Snowmelt,0) -Precip.PET_30day as
WaterDiff_30day, (Precip.CompletePct) *DECODE (Snowmelt.WEcntPast,0,0.9,1) *DECODE
(Snowmelt.WEcntPres,0,0.9,1) as CompletePct
```

```
from GDS_PRECIP_30DAYS_VU Precip, GDS_SNOWMELT_30DAYS_VU Snowmelt
where Precip.Lat = Snowmelt.Lat  (
+
)  AND  Precip.Lon  =  Snowmelt.Lon  (
+
)
```

769 View GDS_PRECIP_30DAYS_VU

769.1 SQL Query for view GDS_PRECIP_30DAYS_VU

```
select GPD.Lat, GPD.Lon, Max(GPD.Day) AS RecentDay, Sum(GPD.GDS) AS GDS_30day, Sum(GPD.PET) AS
PET_30day, Sum(GPD.Rain) AS Rain_30day, Sum(GPD.Snow) AS Snow_30day, Sum(GPD.RainEqv) AS
RainEqv_30day, Round((Count(GPD.GDS)+Count(GPD.Rain)+Count(GPD.Snow)+Count(GPD.RainEqv))/120,1) AS
CompletePct
FROM GDS_PRECIP_DAILY_VU GPD
WHERE GPD.Day>=SYSDATE-30
GROUP BY Lat, Lon
HAVING Count (*) > 15 AND Max (GPD.Day) >=SYSDATE-4
```

770 View GDS_PRECIP_DAILY_VU

770.1 SQL Query for view GDS_PRECIP_DAILY_VU

```
select Round(Latitude,1) AS Lat, Round(Longitude,1) AS Lon, SolarDay AS Day, Round(Avg(GDSvalue),1) AS
GDS, Round(Avg(PETvalue),1) AS PET, Round(Avg(NewRain),1) AS Rain, Round(Avg(NewSnow),1) AS Snow,
Round(Avg(RainEquiv),1) AS RainEqv, Round(Avg(Snowpack),1) AS Snowpack, Round(Avg(WaterEquiv),1) AS
WaterEqv, MIN(DECODE(Joinfour.F5,'E','E',null)) AS F5
FROM GLOBE.SITE_LOCATION Loc, GLOBE.PHENOLOGY_WORKUP JoinFour
WHERE  Loc.SchoolId=JoinFour.SchoolId  AND
Loc.SiteId=JoinFour.SiteId
GROUP BY Round(Latitude,1), Round(Longitude,1), SolarDay
```

771 View GDS_SNOWMELT_30DAYS_VU

771.1 SQL Query for view GDS_SNOWMELT_30DAYS_VU

```
select GDS2.Lat, GDS2.Lon, Max(GDS2.Day) as RecentDay, Max(GDS1.Day) as MonthAgo,
Round(Avg(GDS1.WaterEqv)-Avg(GDS2.WaterEqv),1) as Snowmelt,
Count(GDS1.WaterEqv*DECODE(GDS1.F5,'E',0,1)) as WEcntPast,
Count(GDS2.WaterEqv*DECODE(GDS1.F5,'E',0,1)) AS WEcntPres
FROM GDS_PRECIP_DAILY_VU GDS1, GDS_PRECIP_DAILY_VU GDS2
where GDS1.Lon = GDS2.Lon AND GDS1.Lat = GDS2.Lat AND GDS2.Day >= SYSDATE-4 AND GDS2.Day - GDS1.Day >=
29 AND GDS2.Day - GDS1.Day <= 35- (SYSDATE-GDS2.Day)
GROUP BY GDS2.Lat, GDS2.Lon
```

772 View GLOBE_CAMPAIGN_LIST

772.1 SQL Query for view GLOBE_CAMPAIGN_LIST

```
select GLOBE_SCHOOL.SCHOOLID as CampaignId, GLOBE_SCHOOL.SCHOOLNAME as CampaignName,
GLOBE_SCHOOL.SCHOOLTYPE as SchoolType, GLOBE_SCHOOL.ADDRESSLINE1 as AddressLine1,
GLOBE_SCHOOL.ADDRESSLINE2 as AddressLine2, GLOBE_SCHOOL.CITY as City, GLOBE_SCHOOL.STATEPROVCODE as
StateProvCode, GLOBE_SCHOOL.POSTALCODE as PostalCode, GLOBE_SCHOOL.COUNTRYCODE as Countrycode,
GLOBE_SCHOOL.DIRECTORFULLNAME as CampaignLeader, GLOBE_SCHOOL.PARENTORGANIZATION as ParentOrg,
GLOBE_SCHOOL.SCHOOLTELEPHONE as Telephone, GLOBE_SCHOOL.SCHOOLFAX as Fax, GLOBE_SCHOOL.SCHOOLEMAIL as
Email, GLOBE_SCHOOL.SCHOOLHOMEPAGE as HomePage, GLOBE_SCHOOL.LIFECYCLECODE as LifeCycleCode,
GLOBE_SCHOOL.CYCLECHANGEDATE as CycleChangeDate, GLOBE_SCHOOL.COMMENTS as CampaignDescription,
GLOBE_CAMPAIGN.BEGINDATE as BeginCampaignDate, GLOBE_CAMPAIGN.ENDDATE as EndCampaignDate
from GLOBE.GLOBE_SCHOOL, GLOBE.GLOBE_CAMPAIGN
where as CampaignId = GLOBE_CAMPAIGN.CAMPAIGNID and as CampaignId = GLOBE_CAMPAIGN.PARTICIPANTSCHOOL
```

773 View GLOBE_PI_SCIENTISTS

773.1 SQL Query for view GLOBE_PI_SCIENTISTS

```
SELECT
```

```
  GLOBE_TEACHER.TEACHERID AS "TeacherId",
  GLOBE_TEACHER.TEACHERPREFIX AS "Pre",
  GLOBE_TEACHER.TEACHEROTHERNAMES AS "First",
  GLOBE_TEACHER.TEACHERSURNAME AS "Last",
  GLOBE_TEACHER.WORKTELEPHONE AS "WorkTel",
  GLOBE_TEACHER.TEACHEREMAIL AS "Email",
  GLOBE_SCHOOL.SCHOOLID AS "SchoolId",
  GLOBE_SCHOOL.SCHOOLNAME AS "SchoolName",
  GLOBE_SCHOOL.CITY AS "City",
  GLOBE_SCHOOL.STATEPROVCODE AS "State",
  GLOBE_SCHOOL.SCHOOLTELEPHONE AS "SchTel",
  GLOBE_SCHOOL.SCHOOLFAX AS "SchFax",
  TEACHER_TO_SCHOOL.TEACHERROLE AS "Role"
FROM
  GLOBE.GLOBE_TEACHER, GLOBE.TEACHER_TO_SCHOOL, GLOBE.GLOBE_SCHOOL
WHERE
  GLOBE_SCHOOL.SCHOOLID = TEACHER_TO_SCHOOL.SCHOOLID AND
  GLOBE_TEACHER.TEACHERID = TEACHER_TO_SCHOOL.TEACHERID AND
  TEACHER_TO_SCHOOL.TEACHERROLE = 'P.I.' AND TEACHER_TO_SCHOOL.DATEEND Is Null
```

774 View GRASSBIOM_HR1

774.1 SQL Query for view GRASSBIOM_HR1

```
select
 (select A.SchoolId, a.SiteId, TRUNC(a.Daymeasured) AS Day
from globe.grass_biometry A, globe.grassbiom_ws B, globe.land_cover c
WHERE  A.SchoolId = B.SchoolId AND
A.SiteId = B.SiteId AND
A.DayMeasured = B.DayMeasured AND
A.SchoolId = C.SchoolId AND
A.SiteId = C.SiteId AND
A.DayMeasured >= TO_DATE('1998 06 01','YYYY MM DD') AND
C.MUCCode LIKE '4%'
GROUP By A.Schoolid, A.SiteId, TRUNC(a.Daymeasured)
HAVING COUNT(GreenBiomass) >= 3 AND COUNT(BrownBiomass) >= 3)
```

775 View GRASSBIOM_HR1_LocP

775.1 SQL Query for view GRASSBIOM_HR1_LocP

```
select A.SchoolId, A.SiteId, A.Day,
DECODE(B.DataSource,0,'Unchecked',1,'GPS',2,'Other') as LocMethod,
Count(C.PhotoId) as Photos
from GRASSBIOM_HR1 A, GLOBE.SITE_LOCATION B, GLOBE.SITE_PHOTO C
where
 A.SchoolId = B.SchoolId AND A.SchoolId = C.SchoolId AND
 A.SiteId = B.SiteId AND A.SiteId = C.SiteId
group by A.SchoolId, A.SiteId, A.Day,
DECODE(B.DataSource,0,'Unchecked',1,'GPS',2,'Other')
```

776 View GRASSBIOM_HR2

776.1 SQL Query for view GRASSBIOM_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From GRASSBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
```

```
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From GRASSBIOM_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From GRASSBIOM_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From GRASSBIOM_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From GRASSBIOM_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
```

```
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From GRASSBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

777 View GRASSBIOM_HR3

777.1 SQL Query for view GRASSBIOM_HR3

```
(select
 Hrnbr, SchoolId
 From grassbiom_hr2
 Group by Hrnbr, SchoolId
 Having COUNT(SchoolId) >= 2
)
```

778 View GRASSBIOM_WS_DR1

779 View GRASSBIOM_WS_DR2

779.1 SQL Query for view GRASSBIOM_WS_DR2

```
select GRASSBIOM_WS_DR1.*, 2*biomcnt as DixonCnt
from GRASSBIOM_WS_DR1
```

780 View GRASSBIOM_WS_DU1

781 View GRASSBIOM_WS_DU2

781.1 SQL Query for view GRASSBIOM_WS_DU2

```
select GRASSBIOM_WS_DU1.*, 2*biomcnt as DixonCnt
from GRASSBIOM_WS_DU1
```

782 View GRASSBIOM_WS_MC1

782.1 SQL Query for view GRASSBIOM_WS_MC1

```
select 'GB' as MeasureCode, COUNT(GreenBiomass+BrownBiomass) as BiomCnt
from GLOBE.GRASSBIOM_WS
```

783 View GRASSBIOM_WS_MC2

783.1 SQL Query for view GRASSBIOM_WS_MC2

```
select GRASSBIOM_WS_MC1.*, 2*biomcnt as DixonCnt
from GRASSBIOM_WS_MC1
```

784 View GRASSBIOM_WS_RPRT_VU

784.1 SQL Query for view GRASSBIOM_WS_RPRT_VU

```
select Distinct as SchoolID, 'GB' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from GRASSBIOM_WS
group by SchoolID
```

785 View GRASSBIOM_WS_VU1

785.1 SQL Query for view GRASSBIOM_WS_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'GB' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT(GreenBiomass+BrownBiomass) AS BiomCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from
 GLOBE.GRASSBIOM_WS
where
 UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

786 View GRASSBIOM_WS_VU2

786.1 SQL Query for view GRASSBIOM_WS_VU2

```
select GRASSBIOM_WS_VU1.*, 2*biomcnt as DixonCnt
from GRASS_BIOM_WS_VU1
```

787 View GRASS_BIOMETRY_DR1

787.1 SQL Query for view GRASS_BIOMETRY_DR1

```
select TRUNC(UTtimeReported,'DD') as Day, 'GB' as MeasureCode, COUNT(*) as RowCnt,
SUM(NVL(GreenSquares,0)) +SUM (NVL(BrownSquares,0)) +SUM (NVL(MinusSquares,0)) AS GCCntFROM
GLOBE.GRASS_BIOMETRYWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTtimeReported,'DD')
```

788 View GRASS_BIOMETRY_DR2

788.1 SQL Query for view GRASS_BIOMETRY_DR2

```
select GRASS_BIOMETRY_DR1.*, gccnt as DixonCnt
from GRASS_BIOMETRY_DR1
where gccnt > 0
```

789 View GRASS_BIOMETRY_DU1

789.1 SQL Query for view GRASS_BIOMETRY_DU1

```
select TRUNC(UTtimeMeasured,'DD') as Day, 'GB' as MeasureCode, COUNT(*) as RowCnt,
SUM(NVL(GreenSquares,0)) +SUM (NVL(BrownSquares,0)) +SUM (NVL(MinusSquares,0)) AS GCCntFROM
GLOBE.GRASS_BIOMETRYWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTtimeMeasured,'DD')
```

790 View GRASS_BIOMETRY_DU2

790.1 SQL Query for view GRASS_BIOMETRY_DU2

```
select GRASS_BIOMETRY_DU1.*, gccnt as DixonCnt
from GRASS_BIOMETRY_DU1
where gccnt > 0
```

791 View GRASS_BIOMETRY_MC1

791.1 SQL Query for view GRASS_BIOMETRY_MC1

```
select 'GB' as MeasureCode, COUNT(*) as RowCnt, SUM(NVL(GreenSquares,0)) +SUM (NVL(BrownSquares,0))
+SUM (NVL(MinusSquares,0)) AS GCCntFROM as GLOBE.GRASS_BIOMETRY
```

792 View GRASS_BIOMETRY_MC2

792.1 SQL Query for view GRASS_BIOMETRY_MC2

```
select GRASS_BIOMETRY_MC1.*, 2*gccnt as DixonCnt
from GRASS_BIOMETRY_MC1
where gccnt > 0
```

793 View GRASS_BIOMETRY_RPRT_VU

793.1 SQL Query for view GRASS_BIOMETRY_RPRT_VU

```
select Distinct as SchoolID, 'GB' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from GRASS_BIOMETRY
group by SchoolID
```

794 View GRASS_BIOMETRY_VU1

794.1 SQL Query for view GRASS_BIOMETRY_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'GB' AS MeasureCode,
 COUNT(*) AS RowCnt,
 SUM(NVL(GreenSquares,0)) +SUM (NVL(BrownSquares,0)) +SUM (NVL(MinusSquares,0)) AS GCCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from
 GLOBE.GRASS_BIOMETRYWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY
 SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

795 View GRASS_BIOMETRY_VU2

795.1 SQL Query for view GRASS_BIOMETRY_VU2

```
select GBVU1.*, gccnt AS DixonCnt
from GRASS_BIOMETRY_VU1 GBVU1
```

796 View GREENDOWN_COLOR

796.1 SQL Query for view GREENDOWN_COLOR

```
select
 SchoolId,
 SiteId,
 GreeningId,
 Year,
 GreeningCycle,
 LeafId,
 SolarDay,
 Trunc(SolarDay-TO_DATE(Year,'YYYY'))+1 as Day,
 DyingState,
 LeafColorCode
from
 GLOBE.GREENDOWN_WS
```

797 View GREENDOWN_FALLENLEAF

797.1 SQL Query for view GREENDOWN_FALLENLEAF

```
select distinct
 X.SchoolId,
 X.SiteId,
 X.GreeningId,
 X.Year,
 X.GreeningCycle,
 X.LeafId
from
 (select * from GLOBE.GREENDOWN_WS where DyingState='F') X,
 (select * from GLOBE.GREENDOWN_WS where DyingState='E') Y
where
 X.SchoolId=Y.SchoolId(+) and
 X.SiteId=Y.SiteId(+) and
 X.Year=Y.Year(+) and
 X.GreeningCycle=Y.GreeningCycle(+) and
 X.GreeningId=Y.GreeningId(+) and
 X.LeafId=Y.LeafId(+) and
 (Y.SchoolId is null or X.SolarDay<Y.SolarDay)
```

798 View GREENDOWN_STATES

798.1 SQL Query for view GREENDOWN_STATES

```
select
 X.SchoolId,
 X.SiteId,
 X.GreeningId,
 X.Year,
 X.GreeningCycle,
 X.LeafId,
 Trunc(MAX(U.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as LastUnchgDay,
 Trunc(MIN(C.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as FirstChgDay,
 Trunc(MAX(C.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as LastChgDay,
 Trunc(MIN(S.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as FirstSnowDay,
 Trunc(MAX(S.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as LastSnowDay,
 Trunc(MIN(E.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as EndDay,
 Trunc(MIN(F.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as FallDay,
 MAX(Decode(FL.SchoolId, null, '-', 'Y')) as EarlyFall
from
  GLOBE.GREENDOWN_WS X,
 (select * from GLOBE.GREENDOWN_WS where DyingState='U') U,
 (select * from GLOBE.GREENDOWN_WS where DyingState='C') C,
 (select * from GLOBE.GREENDOWN_WS where DyingState='S') S,
 (select * from GLOBE.GREENDOWN_WS where DyingState='E') E,
 (select * from GLOBE.GREENDOWN_WS where DyingState='F') F,
  GREENDOWN_FALLENLEAF FL
where
 X.SchoolId=U.SchoolId(+) and
 X.SiteId=U.SiteId(+) and
 X.GreeningId=U.GreeningId(+) and
 X.Year=U.Year(+) and
 X.GreeningCycle=U.GreeningCycle(+) and
 X.LeafId=U.LeafId(+) and
 X.SchoolId=C.SchoolId(+) and
 X.SiteId=C.SiteId(+) and
 X.GreeningId=C.GreeningId(+) and
 X.Year=C.Year(+) and
 X.GreeningCycle=C.GreeningCycle(+) and
```

```
 X.LeafId=C.LeafId(+) and
 X.SchoolId=S.SchoolId(+) and
 X.SiteId=S.SiteId(+) and
 X.GreeningId=S.GreeningId(+) and
 X.Year=S.Year(+) and
 X.GreeningCycle=S.GreeningCycle(+) and
 X.LeafId=S.LeafId(+) and
 X.SchoolId=E.SchoolId(+) and
 X.SiteId=E.SiteId(+) and
 X.GreeningId=E.GreeningId(+) and
 X.Year=E.Year(+) and
 X.GreeningCycle=E.GreeningCycle(+) and
 X.LeafId=E.LeafId(+) and
 X.SchoolId=F.SchoolId(+) and
 X.SiteId=F.SiteId(+) and
 X.GreeningId=F.GreeningId(+) and
 X.Year=F.Year(+) and
 X.GreeningCycle=F.GreeningCycle(+) and
 X.LeafId=F.LeafId(+) and
 X.SchoolId=FL.SchoolId(+) and
 X.SiteId=FL.SiteId(+) and
 X.GreeningId=FL.GreeningId(+) and
 X.Year=FL.Year(+) and
 X.GreeningCycle=FL.GreeningCycle(+) and
 X.LeafId=FL.LeafId(+)
group by
 X.SchoolId, X.SiteId, X.GreeningId, X.Year, X.GreeningCycle, X.LeafId
```

799 View GREENDOWN_WS_VU1

799.1 SQL Query for view GREENDOWN_WS_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(Solarday,'MM') AS Month,
  'GN' as MeasureCode,
  COUNT(*) as RowCnt,
  COUNT(Dyingstate) as DSCnt,
  COUNT(Leafcolorcode)  as LCCnt,
  MIN(Solarday) AS BegDate,
  MAX(Solarday) AS EndDate,
  MIN(UTtimereported) AS BegRDate,
  MAX(UTtimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-Solarday),2) AS LagTimeAvg
FROM
  GREENDOWN_WS
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(Solarday,'MM')
```

800 View GREENDOWN_WS_VU2

800.1 SQL Query for view GREENDOWN_WS_VU2

```
SELECT
  GREENDOWN_WS_VU1.*,
  (DSCnt+LCCnt) as DixonCnt
FROM
  GREENDOWN_WS_VU1
```

801 View GREENING_METADATA_VU1

801.1 SQL Query for view GREENING_METADATA_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(UTtimereported,'MM') AS Month,
  'GN' as MeasureCode,
  COUNT(*) as RowCnt,
  COUNT(TreeGrassShrub) as TGSCnt,
```

```
  COUNT(Genus4)         as GenCnt,
  MIN(DateDefined) AS BegDate,
  MAX(DateDefined) AS EndDate,
  MIN(UTTimereported) AS BegRDate,
  MAX(UTTimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-DateDefined),2) AS LagTimeAvg
FROM
  GREENING_METADATA
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(UTtimereported,'MM')
```

802 View GREENING_METADATA_VU2

802.1 SQL Query for view GREENING_METADATA_VU2

```
SELECT
  GREENING_METADATA_VU1.*,
  (TGSCnt+GenCnt) as DixonCnt
FROM
  GREENING_METADATA_VU1
```

803 View GREENUP_GROWTH

803.1 SQL Query for view GREENUP_GROWTH

```
select
 X.SchoolId,
 X.SiteId,
 X.GreeningId,
 X.Year,
 X.GreeningCycle,
 X.LeafId,
 X.SolarDay,
 Trunc(X.SolarDay-TO_DATE(X.Year,'YYYY'))+1 as Day,
 X.GreeningState,
 X.LeafLengthMM as Length,
 Trunc(X.LeafLengthMM/Greatest(1,Y.MaxLength)*100) as PctOfMax
from
 GLOBE.GREENUP_WS  X,
 GREENUP_MAXLENGTH  Y
where
 X.GreeningState <> 'D' and
 X.SchoolId=Y.SchoolId and
 X.SiteId=Y.SiteId and
 X.GreeningId=Y.GreeningId and
 X.Year=Y.Year and
 X.GreeningCycle=Y.GreeningCycle and
 X.LeafId=Y.LeafId
```

804 View GREENUP_LOSTLEAF

804.1 SQL Query for view GREENUP_LOSTLEAF

```
select distinct
 X.SchoolId,
 X.SiteId,
 X.GreeningId,
 X.Year,
 X.GreeningCycle,
 X.LeafId
from
 (select * from GLOBE.GREENUP_WS where GreeningState='L') X,
 (select * from GLOBE.GREENUP_WS where GreeningState='M') Y
where
 X.SchoolId=Y.SchoolId(+) and
 X.SiteId=Y.SiteId(+) and
 X.Year=Y.Year(+) and
 X.GreeningCycle=Y.GreeningCycle(+) and
 X.GreeningId=Y.GreeningId(+) and
 X.LeafId=Y.LeafId(+) and
 (Y.SchoolId is null or (X.SolarDay<Y.SolarDay))
```

805 View GREENUP_MAXLENGTH

805.1 SQL Query for view GREENUP_MAXLENGTH

```
select
 SchoolId,
 SiteId,
 GreeningId,
 Year,
 GreeningCycle,
 LeafId,
 MAX(LeafLengthMM) as MaxLength
from GLOBE.GREENUP_WS
group by SchoolId, SiteId, GreeningId, Year, GreeningCycle, LeafId
```

806 View GREENUP_STATES

806.1 SQL Query for view GREENUP_STATES

```
select
 X.SchoolId,
 X.SiteId,
 X.GreeningId,
 X.Year,
 X.GreeningCycle,
 X.LeafId,
 MAX(ML.MaxLength) as MaxLength,
 Trunc(MIN(S.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as SwellDay,
 Trunc(MIN(B.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as BudburstDay,
 Trunc(MIN(G.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as GrowDay,
 MIN(GG.Day) as Pct98Day,
 Trunc(MIN(M.SolarDay)-TO_DATE(X.Year,'YYYY'))+1 as MatureDay,
 MAX(Decode(LL.SchoolId, null, '-', 'Y')) as Lost
from
  GLOBE.GREENUP_WS  X,
 (Select * from GLOBE.GREENUP_WS where GreeningState='S') S,
 (Select * from GLOBE.GREENUP_WS where GreeningState='B') B,
 (Select * from GLOBE.GREENUP_WS where GreeningState='G') G,
 (Select * from GLOBE.GREENUP_WS where GreeningState='M') M,
 (Select * from GREENUP_GROWTH where PctOfMax>=98) GG,
 GREENUP_LOSTLEAF LL,
 GREENUP_MAXLENGTH ML
where
 X.SchoolId=S.SchoolId(+) and
 X.SiteId=S.SiteId(+) and
 X.Year=S.Year(+) and
 X.GreeningCycle=S.GreeningCycle(+) and
 X.GreeningId=S.GreeningId(+) and
 X.LeafId=S.LeafId(+) and
 X.SchoolId=B.SchoolId(+) and
 X.SiteId=B.SiteId(+) and
 X.Year=B.Year(+) and
 X.GreeningCycle=B.GreeningCycle(+) and
 X.GreeningId=B.GreeningId(+) and
 X.LeafId=B.LeafId(+) and
 X.SchoolId=G.SchoolId(+) and
 X.SiteId=G.SiteId(+) and
 X.Year=G.Year(+) and
 X.GreeningCycle=G.GreeningCycle(+) and
 X.GreeningId=G.GreeningId(+) and
 X.LeafId=G.LeafId(+) and
 X.SchoolId=M.SchoolId(+) and
 X.SiteId=M.SiteId(+) and
 X.Year=M.Year(+) and
 X.GreeningCycle=M.GreeningCycle(+) and
 X.GreeningId=M.GreeningId(+) and
 X.LeafId=M.LeafId(+) and
 X.SchoolId=GG.SchoolId(+) and
 X.SiteId=GG.SiteId(+) and
 X.Year=GG.Year(+) and
 X.GreeningCycle=GG.GreeningCycle(+) and
 X.GreeningId=GG.GreeningId(+) and
 X.LeafId=GG.LeafId(+) and
 X.SchoolId=LL.SchoolId(+) and
 X.SiteId=LL.SiteId(+) and
```

```
 X.Year=LL.Year(+) and
 X.GreeningCycle=LL.GreeningCycle(+) and
 X.GreeningId=LL.GreeningId(+) and
 X.LeafId=LL.LeafId(+) and
 X.SchoolId=ML.SchoolId and
 X.SiteId=ML.SiteId and
 X.Year=ML.Year and
 X.GreeningCycle=ML.GreeningCycle and
 X.GreeningId=ML.GreeningId and
 X.LeafId=ML.LeafId
group by
 X.SchoolId, X.SiteId, X.GreeningId, X.Year, X.GreeningCycle, X.LeafId
```

807 View GREENUP_SUMMARY

807.1 SQL Query for view GREENUP_SUMMARY

```
select
 Round(SL.Latitude) as Latitude,
 Round(SL.Elevation/500)*500 as Elevation,
 TT.Genus,
 TT.Species,
 GS.Year,
 GS.GreeningCycle,
 ROUND(AVG(GS.MaxLength)) as MaxLength,
 ROUND(AVG(GS.SwellDay)) as SwellDay,
 ROUND(AVG(GS.GrowDay)) as GrowDay,
 ROUND(AVG(GS.Pct98Day)) as Pct98Day,
 ROUND(AVG(GS.MatureDay)) as MatureDay,
 COUNT(*) as LeafCnt
from
 GLOBE.GREENING_METADATA GM,
 GLOBE.TREE_TYPES TT,
 GREENUP_STATES GS,
 GLOBE.SITE_LOCATION SL
where
 GS.Lost='-' and
 GS.SchoolId=GM.SchoolId and
 GS.SiteId=GM.SiteId and
 GS.GreeningId=GM.GreeningId and
 GM.TreeGrassShrub IN ('T','S') and
 GM.Genus4=TT.Genus4 and
 GM.Species4=TT.Species4 and
 GM.SchoolId=SL.SchoolId and
 GM.SiteId=SL.SiteId
group by
 Round(SL.Latitude), Round(SL.Elevation/500)*500,
 TT.Genus, TT.Species, GS.Year, GS.GreeningCycle
```

808 View GREENUP_WS_VU1

808.1 SQL Query for view GREENUP_WS_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(Solarday,'MM') AS Month,
  'GN' as MeasureCode,
  COUNT(*) as RowCnt,
  COUNT(Greeningstate) as GSCnt,
  COUNT(Leaflengthmm)  as LLCnt,
  MIN(Solarday) AS BegDate,
  MAX(Solarday) AS EndDate,
  MIN(UTtimereported) AS BegRDate,
  MAX(UTtimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-Solarday),2) AS LagTimeAvg
FROM
  GREENUP_WS
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(Solarday,'MM')
```

809 View GREENUP_WS_VU2

809.1 SQL Query for view GREENUP_WS_VU2

```
SELECT
  GREENUP_WS_VU1.*,
  (GSCnt+LLCnt) as DixonCnt
FROM
  GREENUP_WS_VU1
```

810 View GrassTrees_HR1

810.1 SQL Query for view GrassTrees_HR1

```
select
 SchoolId,
 SiteId,
 Day,
 TorG
from
 Grass_HR0
UNION
select
 SchoolId,
 SiteId,
 Day,
 TorG
from
 Trees_HR0
```

811 View GrassTrees_HR2

811.1 SQL Query for view GrassTrees_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from GrassTrees_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 Count(*) >= 2 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from GrassTrees_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 Count(*) >= 2 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from GrassTrees_HR1 A
group by A.SchoolId,
```

```
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 Count(*) >= 2 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from GrassTrees_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 Count(*) >= 2 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from GrassTrees_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 Count(*) >= 2 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, Min(Day) as BeginDay, Max(Day) as EndDay, Count(*) as NbrObs
from GrassTrees_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 Count(*) >= 2 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

812 View Grass_HR0

812.1 SQL Query for view Grass_HR0

```
select
 GB.SchoolId,
 GB.SiteId,
 DayMeasured as Day,
 'G' as TorG
from
 GLOBE.GRASS_BIOMETRY GB,
 GLOBE.SITE_LOCATION SL
where
 GB.SchoolId = SL.SchoolId AND
 GB.SiteId = SL.SiteId AND
```

```
 SL.DataSource in (1,2) AND
 GB.DayMeasured >= TO_DATE('20000601','YYYYMMDD') AND
 GB.GreenBiomassAvg is not null AND
 GB.BrownBiomassAvg is not null
```

813 View GreenDown_HR1

813.1 SQL Query for view GreenDown_HR1

```
select
 SchoolId,
 SiteId,
 GreeningId,
 Year,
 GreeningCycle,
 Min(BeginDay) as BeginDay,
 Max(EndDay) as EndDay
from
 (select
   P.SchoolId,
   P.SiteId,
   P.GreeningId,
   P.Year,
   P.GreeningCycle,
   P.DyingState,
   Count(*) as NbrObs,
   Min(P.SolarDay) as BeginDay,
   Max(P.SolarDay) as EndDay
  from
   GLOBE.GREENDOWN_WS P,
   GLOBE.SITE_LOCATION SL
  where
   P.SchoolId = SL.SchoolId AND
   P.SiteId = SL.SiteId AND
   SL.DataSource in (1,2) AND
   P.DyingState in ('C','F','S')
  group by P.SchoolId, P.SiteId,
           P.GreeningId, P.Year,
           P.GreeningCycle, P.DyingState ) X
group by SchoolId, SiteId, GreeningId, Year, GreeningCycle
having
 count(distinct DyingState) >= 1 AND
 Sum(NbrObs) >= 7
```

814 View GreenUp_HR1

814.1 SQL Query for view GreenUp_HR1

```
select
 SchoolId,
 SiteId,
 GreeningId,
 Year,
 GreeningCycle,
 Min(BeginDay) as BeginDay,
 Max(EndDay) as EndDay
from
 (select
   P.SchoolId,
   P.SiteId,
   P.GreeningId,
   P.Year,
   P.GreeningCycle,
   P.GreeningState,
   Count(*) as NbrObs,
   Min(P.SolarDay) as BeginDay,
   Max(P.SolarDay) as EndDay
  from
   GLOBE.GREENUP_WS P,
   GLOBE.SITE_LOCATION SL
  where
   P.SchoolId = SL.SchoolId AND
   P.SiteId = SL.SiteId AND
   SL.DataSource in (1,2) AND
   P.GreeningState in ('S','B','G','M')
```

```
  group by P.SchoolId, P.SiteId,
           P.GreeningId, P.Year,
           P.GreeningCycle, P.GreeningState ) X
group by SchoolId, SiteId, GreeningId, Year, GreeningCycle
having
 count(distinct GreeningState) >= 3 AND
 Sum(NbrObs) >= 7
```

815 View HAZE_VISUALIZATION

815.1 SQL Query for view HAZE_VISUALIZATION

```
Select
 H.SchoolId,
 H.SiteId,
 H.SolarDay,
 H.SolarHour,
 H.SolarMin,
 H.PhotometerId,
 H.AirTemp,
 H.BarometricMb,
 H.SkyColorEst,
 H.HazeEst,
 Red.AOTAvg as AOTAvgRed,
 Red.AOTMin as AOTMinRed,
 Red.AOTMax as AOTMaxRed,
 Red.TransPctAvg as TransPctAvgRed,
 Red.TransPctMin as TransPctMinRed,
 Red.TransPctMax as TransPctMaxRed,
 Grn.AOTAvg as AOTAvgGrn,
 Grn.AOTMin as AOTMinGrn,
 Grn.AOTMax as AOTMaxGrn,
 Grn.TransPctAvg as TransPctAvgGrn,
 Grn.TransPctMin as TransPctMinGrn,
 Grn.TransPctMax as TransPctMaxGrn,
 H.Comments,
 H.VisualDay,
 H.UTtimeMeasured,
 H.UTtimeReported,
 H.DBmasterSite
FROM
 GLOBE.HAZE H,
 GLOBE.HAZE_WS_SUMMARY Red,
 GLOBE.HAZE_WS_SUMMARY Grn
WHERE
 H.SchoolId = Red.SchoolId(+) AND
 H.SiteId = Red.SiteId(+) AND
 H.SolarDay = Red.SolarDay(+) AND
 H.SolarMin = Red.SolarMin(+) AND
 H.PhotometerId = Red.PhotometerId(+) AND
 H.SchoolId = Grn.SchoolId(+) AND
 H.SiteId = Grn.SiteId(+) AND
 H.SolarDay = Grn.SolarDay(+) AND
 H.SolarMin = Grn.SolarMin(+) AND
 H.PhotometerId = Grn.PhotometerId(+) AND
 Red.ChannelId = 'R' AND
 Grn.ChannelId = 'G'
```

816 View HAZE_VU1

816.1 SQL Query for view HAZE_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(UTTimeMeasured,'MM') as Month,
  'AZ' as MeasureCode,
  Count(*) as RowCnt,
  COUNT(AirTemp) as ATmpCnt,
  COUNT(BarometricMb) as BarPCnt,
  SUM(DECODE(SkyColorEst,0,0,1)) as SkyCCnt,
```

```
  SUM(DECODE(NVL(HazeEst,0),0,0,1)) as HazeCnt,
  MIN(TransPctMinGrn) as MinTPctG,
  MIN(TransPctMinRed) as MinTPctR,
  MAX(TransPctMaxGrn) as MaxTPctG,
  Max(TransPctMaxRed) as MaxTPctR,
  ROUND(AVG(TransPctAvgGrn),1) as AvgTPctG,
  ROUND(AVG(TransPctAvgRed),1) as AvgTPctR,
  MIN(AOTMinGrn) as MinAOTG,
  MIN(AOTMinRed) as MinAOTR,
  MAX(AOTMaxGrn) as MaxAOTG,
  MAX(AOTMaxRed) as MaxAOTR,
  ROUND(AVG(AOTAvgGrn),3) as AvgAOTG,
  ROUND(AVG(AOTAvgRed),3) as AvgAOTR,
  MIN(UTTimeMeasured) AS BegDate,
  MAX(UTTimeMeasured) AS EndDate,
  MIN(UTtimereported) AS BegRDate,
  MAX(UTtimereported) AS EndRDate,
  ROUND(AVG(UTtimeReported-UTTimeMeasured),2) AS LagTimeAvg
from HAZE_VISUALIZATION
where UTtimeReported < TRUNC(SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(UTTimeMeasured,'MM')
```

817 View HAZE_VU2

817.1 SQL Query for view HAZE_VU2

```
select
  HAZE_VU1.*,
  ATmpCnt+BarPCnt+SkyCCnt+HazeCnt as dixoncnt
from HAZE_VU1
```

818 View HAZE_WS_CALC

818.1 SQL Query for view HAZE_WS_CALC

```
SELECT
  HWS.SCHOOLID,
  HWS.SITEID,
  HWS.SOLARDAY,
  HWS.SOLARHOUR,
  HWS.SOLARMIN,
  HWS.PHOTOMETERID,
  HWS.CHANNELID,
  HWS.SAMPLENBR,
  HWS.DARKVOLTAGE,
  HWS.SUNLIGHTVOLTAGE,
  CalcAOT(ETConst, Wavelength, HWS.DarkVoltage, SunlightVoltage, BarometricMB,
          HWS.UTTimeMeasured, Latitude, Longitude) AS AOT,
  CalcAZTransPct(AOT) AS TransmissionPct,
  HWS.UTTimeMeasured,
  HWS.UTTimeReported
FROM HAZE H, HAZE_WS HWS, PHOTOMETER_CALIB PC, SITE_LOCATION SL
WHERE H.Schoolid = HWS.Schoolid AND H.Siteid = HWS.Siteid AND
  H.SolarDay = HWS.SolarDay AND H.SolarHour = HWS.SolarHour AND
  H.SolarMin = HWS.SolarMin AND H.Photometerid = HWS.PhotometerID AND
  H.Photometerid = PC.Photometerid AND HWS.Channelid = PC.Channelid AND
  H.Schoolid = SL.Schoolid and H.Siteid = SL.Siteid
```

819 View HAZE_WS_SUMMARY

819.1 SQL Query for view HAZE_WS_SUMMARY

```
select
 HWC.SCHOOLID,
 HWC.SITEID,
 HWC.SOLARDAY,
 HWC.SOLARHOUR,
 HWC.SOLARMIN,
 HWC.PHOTOMETERID,
 HWC.CHANNELID,
 Round(Avg(HWC.AOT),4) AS AOTAvg,
 Round(Min(HWC.AOT),4) AS AOTMin,
 Round(Max(HWC.AOT),4) AS AOTMax,
```

```
 Round(Avg(HWC.TRANSMISSIONPCT),1) AS TransPctAvg,
 Round(Min(HWC.TRANSMISSIONPCT),1) AS TransPctMin,
 Round(Max(HWC.TRANSMISSIONPCT),1) AS TransPctMax
from GLOBE.HAZE_WS_CALC HWC
GROUP BY
 HWC.SCHOOLID, HWC.SITEID, HWC.SOLARDAY,
 HWC.SOLARHOUR, HWC.SOLARMIN, HWC.PHOTOMETERID,
 HWC.CHANNELID
```

820 View HAZE_WS_VU1

820.1 SQL Query for view HAZE_WS_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(UTTimeMeasured,'MM') AS Month,
  'AZ' as MeasureCode,
  COUNT(*) as RowCnt,
  COUNT(DarkVoltage*SunlightVoltage) as VoltCnt,
  MIN(UTTimeMeasured) AS BegDate,
  MAX(UTTimeMeasured) AS EndDate,
  MIN(UTTimereported) AS BegRDate,
  MAX(UTTimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-UTTimeMeasured),2) AS LagTimeAvg
FROM
  HAZE_WS
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(UTTimeMeasured,'MM')
```

821 View HAZE_WS_VU2

821.1 SQL Query for view HAZE_WS_VU2

```
SELECT
  HAZE_WS_VU1.*,
  2*VoltCnt AS DixonCnt
FROM
  HAZE_WS_VU1
```

822 View HONOR_ROLL_OVERALL

822.1 SQL Query for view HONOR_ROLL_OVERALL

```
select
 SchoolId,
 SUM(DixonCnt) as TotalObs,
 MIN(BegDate) as FirstObs,
 MAX(EndDate) as LastObs
from
 GLOBE.DIXON_SUMMARY_GROUPED
group by
 SchoolId
```

823 View HONOR_ROLL_SUMMARY_VU1

823.1 SQL Query for view HONOR_ROLL_SUMMARY_VU1

```
select
 "SchoolId" as SchoolId,
 TRUNC(("HRnbr"-1)/6) +1999 as BeginYear,
 TRUNC(("HRnbr"-1)/6) +2000 as EndYear,
 "Honor Roll" as HonorRoll,
 COUNT("HRnbr") as Citations
from GLOBE.HR_CATEGORIES_ALL
group by
 "SchoolId",
 TRUNC(("HRnbr"-1)/6)+1999,
```

```
 TRUNC(("HRnbr"-1)/6)+2000,
 "Honor Roll"
```

824 View HONOR_ROLL_SUMMARY_VU2

824.1 SQL Query for view HONOR_ROLL_SUMMARY_VU2

```
select
 GC.CountryCode as CountryCode,
 GC.GlobeName as Country,
 SP.RegionType as RegionType,
 SP.ShortName as RegionName,
 GS.SchoolName as SchoolName,
 HRS.SchoolId,
 HRS.BeginYear,
 HRS.EndYear,
 HRS.HonorRoll,
 HRS.Citations
from
 Honor_Roll_Summary_VU1 HRS,
 GLOBE.GLOBE_COUNTRY GC,
 GLOBE.GLOBE_SCHOOL GS,
 GLOBE.STATE_PROV SP
where
 HRS.SchoolId = GS.SchoolId AND
 GS.CountryCode = GC.CountryCode AND
 GS.CountryCode = SP.CountryCode ( + ) AND
 GS.StateProvCode = SP.StateProvCode ( + )
```

825 View HONOR_ROLL_SUMMARY_VU3

825.1 SQL Query for view HONOR_ROLL_SUMMARY_VU3

```
select CountryCode, Country, RegionType, RegionName, BeginYear, EndYear, HonorRoll, SUM(Citations) as
Citations
from HONOR_ROLL_SUMMARY_VU2
group by CountryCode, Country, RegionType, RegionName, BeginYear, EndYear, HonorRoll
```

826 View HONOR_ROLL_SUMMARY_VU4

826.1 SQL Query for view HONOR_ROLL_SUMMARY_VU4

```
select CountryCode, Country, DECODE(RegionType || ' of ' || RegionName, ' of ', null, RegionType || '
of ' || RegionName) as StateProv, TO_CHAR(BeginYear) || ' - ' || TO_CHAR (EndYear) as Term, HonorRoll,
Citations
from HONOR_ROLL_SUMMARY_VU3
```

827 View HR_ALLCNT_SUMS

827.1 SQL Query for view HR_ALLCNT_SUMS

```
select
 X."SchoolId",
 X."AllCnt",
 NVL(Y."AllNewCnt",0) as "AllNewCnt",
 NVL(Z."AllYearCnt",0) as "AllYearCnt",
 NVL(W."AllPrevCnt",0) as "AllPrevCnt"
from
    (select
      "SchoolId",
      COUNT("HRnbr") as "AllCnt"
     from GLOBE.HR_CATEGORIES_ALL
     group by "SchoolId")  X,
    (select
      "SchoolId",
      COUNT("HRnbr") as "AllNewCnt"
     from GLOBE.HR_CATEGORIES_NEW
     group by "SchoolId")  Y,
    (select
      "SchoolId",
```

```
      COUNT("HRnbr") as "AllYearCnt"
     from GLOBE.HR_CATEGORIES_ALL
     where "HRnbr" >= TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2)-5
     group by "SchoolId")  Z,
    (select
      "SchoolId",
      COUNT("HRnbr") as "AllPrevCnt"
     from GLOBE.HR_CATEGORIES_NEW
     where TRUNC(("HRnbr"+5)/6) < TRUNC((TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('19990401','YYYYMMDD')))/2)+5)/6)
     group by "SchoolId")  W
where
 X."SchoolId" = Y."SchoolId"(+) AND
 X."SchoolId" = Z."SchoolId"(+) AND
 X."SchoolId" = W."SchoolId"(+)
```

828 View HR_CATEGORIES_ALL

828.1 SQL Query for view HR_CATEGORIES_ALL

```
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Atmosphere' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_ATMOS_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Clouds' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_CLOUDS_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Hydrology' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_HYDRO_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Soil Moisture and Temperature' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_SOILMT_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Soil Characterization' as "Honor Roll",
 '12-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_SOILCHAR_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Land Cover/Biology' as "Honor Roll",
 '12-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
```

```
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_LCBIO_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Climate' as "Honor Roll",
 '12-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_CLIMATE_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Advanced Atmosphere' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_ATMOSADV_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Earth System Science' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_EARTHSYS_CURRENT
UNION ALL
select
 SchoolId as "SchoolId",
 HRnbr as "HRnbr",
 DixonCnt as "DixonCnt",
 'Pan GLOBE' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(BeginMonth,'Month')) || ' ' || TO_CHAR(BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(EndMonth,'Month')) || ' ' || TO_CHAR(EndMonth,'YYYY') as "End Month"
from GLOBE.HR_PANGLOBE_CURRENT
```

829 View HR_CATEGORIES_ALL_CNT

829.1 SQL Query for view HR_CATEGORIES_ALL_CNT

```
select
 CA."SchoolId" as SchoolId,
 CD.HonorRoll AS HonorRoll,
 Count(CD.HonorRoll) AS HRcnt
from
 HR_CATEGORIES_ALL CA,
 HR_CATEGORIES_DISTINCT CD
where
 CA."Honor Roll" = CD.HonorRoll
group by CA."SchoolId", CD.HonorRoll
```

830 View HR_CATEGORIES_CERT_MONTHS

830.1 SQL Query for view HR_CATEGORIES_CERT_MONTHS

```
select
 X.SchoolId,
 X.CertNbr,
 X.HonorRoll,
 X.HRcnt,
 ListMonthsCert(X.SchoolId, X.CertNbr, X.HonorRoll) AS EndMonths
from
 ( select
    CN."SchoolId" as SchoolId,
    TRUNC((CN."HRnbr"+5)/6) as CertNbr,
    CD.HonorRoll AS HonorRoll,
    Count(CD.HonorRoll) AS HRcnt
   from
```

```
    GLOBE.HR_CATEGORIES_NEW CN,
    GLOBE.HR_CATEGORIES_DISTINCT CD
   where
    CN."Honor Roll" = CD.HonorRoll
   group by CN."SchoolId", TRUNC((CN."HRnbr"+5)/6), CD.HonorRoll) X
```

831 View HR_CATEGORIES_DISTINCT

831.1 SQL Query for view HR_CATEGORIES_DISTINCT

```
select DISTINCT "Honor Roll" as HonorRoll
from HR_CATEGORIES_ALL
```

832 View HR_CATEGORIES_NEW

832.1 SQL Query for view HR_CATEGORIES_NEW

```
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Atmosphere' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_ATMOS_CURRENT A, GLOBE.HR_ATMOS_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Clouds' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_CLOUDS_CURRENT A, GLOBE.HR_CLOUDS_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Hydrology' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_HYDRO_CURRENT A, GLOBE.HR_HYDRO_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Soil Moisture and Temperature' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_SOILMT_CURRENT A, GLOBE.HR_SOILMT_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Soil Characterization' as "Honor Roll",
 '12-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
```

```
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_SOILCHAR_CURRENT A, GLOBE.HR_SOILCHAR_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Land Cover/Biology' as "Honor Roll",
 '12-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_LCBIO_CURRENT A, GLOBE.HR_LCBIO_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Climate' as "Honor Roll",
 '12-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_CLIMATE_CURRENT A, GLOBE.HR_CLIMATE_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Advanced Atmosphere' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_ATMOSADV_CURRENT A, GLOBE.HR_ATMOSADV_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Earth System Science' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_EARTHSYS_CURRENT A, GLOBE.HR_EARTHSYS_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
UNION ALL
select
 A.SchoolId as "SchoolId",
 A.HRnbr as "HRnbr",
 A.DixonCnt as "DixonCnt",
 'Pan GLOBE' as "Honor Roll",
 '4-month' as "Time Span",
 RTRIM(TO_CHAR(A.BeginMonth,'Month')) || ' ' || TO_CHAR(A.BeginMonth,'YYYY') as "Begin Month",
 RTRIM(TO_CHAR(A.EndMonth,'Month')) || ' ' || TO_CHAR(A.EndMonth,'YYYY') as "End Month"
 from GLOBE.HR_PANGLOBE_CURRENT A, GLOBE.HR_PANGLOBE_PREVIOUS B
 where A.SchoolId = B.SchoolId (+)
 and A.HRnbr = B.HRnbr (+)
 and B.SchoolId is null
```

833 View HR_CATEGORIES_NEW_MONTHS

833.1 SQL Query for view HR_CATEGORIES_NEW_MONTHS

```
select
```

```
 X.SchoolId,
 X.HonorRoll,
 X.HRcnt,
 ListMonthsNew(X.SchoolId, X.HonorRoll) AS EndMonths
from
 (select
  CN."SchoolId" as SchoolId,
  CD.HonorRoll as HonorRoll,
  Count(CD.HonorRoll) as HRcnt
 from
  GLOBE.HR_CATEGORIES_NEW CN,
  GLOBE.HR_CATEGORIES_DISTINCT CD
 where
 CN."Honor Roll" = CD.HonorRoll
 group by CN."SchoolId", CD.HonorRoll) X
/
```

834 View HR_CATEGORIES_PREV_MONTHS

834.1 SQL Query for view HR_CATEGORIES_PREV_MONTHS

```
select
 X.SchoolId,
 X.HonorRoll,
 X.HRcnt,
 ListMonthsPrev(X.SchoolId, X.HonorRoll) AS EndMonths
from
 (select
  CN."SchoolId" as SchoolId,
  CD.HonorRoll as HonorRoll,
  Count(CD.HonorRoll) as HRcnt
 from
  GLOBE.HR_CATEGORIES_NEW CN,
  GLOBE.HR_CATEGORIES_DISTINCT CD
 where
 CN."Honor Roll" = CD.HonorRoll
  AND
    CN."HRnbr" < TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
    TO_DATE('19990401','YYYYMMDD')))/2) - 5
 group by CN."SchoolId", CD.HonorRoll) X
/
```

835 View HR_CATEGORIES_YEAR_CNT

835.1 SQL Query for view HR_CATEGORIES_YEAR_CNT

```
select
 CA."SchoolId" as SchoolId,
 CD.HonorRoll AS HonorRoll,
 Count(CD.HonorRoll) AS HRcnt
from
 HR_CATEGORIES_ALL CA,
 HR_CATEGORIES_DISTINCT CD
where
 CA."Honor Roll" = CD.HonorRoll AND
 CA."HRnbr" >= TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2) - 5
group by CA."SchoolId", CD.HonorRoll
```

836 View HR_CATEGORIES_YEAR_MONTHS

836.1 SQL Query for view HR_CATEGORIES_YEAR_MONTHS

```
select
 X.SchoolId,
 X.HonorRoll,
 X.HRcnt,
 ListMonthsYear(X.SchoolId, X.HonorRoll) AS EndMonths
from
 (select
  CA."SchoolId" as SchoolId,
  CD.HonorRoll AS HonorRoll,
  Count(CD.HonorRoll) AS HRcnt
 from
  GLOBE.HR_CATEGORIES_ALL CA,
  GLOBE.HR_CATEGORIES_DISTINCT CD
```

```
 where
    CA."Honor Roll" = CD.HonorRoll
  AND
    CA."HRnbr" >= TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
    TO_DATE('19990401','YYYYMMDD')))/2) - 5
 group by CA."SchoolId", CD.HonorRoll) X
```

837 View HR_CERTIFICATES

837.1 SQL Query for view HR_CERTIFICATES

```
select DISTINCT "SchoolId" as SchoolId
from HR_CATEGORIES_ALL
where "HRnbr" > =  TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'), TO_DATE('19990401','YYYYMMDD')))/2) - 5
UNION
select DISTINCT "SchoolId" as SchoolId
from HR_CATEGORIES_NEW
```

838 View HR_CERTIFICATES_APRIL

838.1 SQL Query for view HR_CERTIFICATES_APRIL

```
select
 X.SchoolId as "SchoolId",
 S.SchoolName as "SchoolName",
 X.CertNbr as "CertNbr",
 X.HonorRoll as "HonorRoll",
 X.HRcnt as "NbrNew",
 ListMonthsCert(X.SchoolId, X.CertNbr, X.HonorRoll) as "AllCertMonths"
from
 GLOBE.GLOBE_SCHOOL S,
 ( select
    CN."SchoolId" as SchoolId,
    TRUNC((CN."HRnbr"+5)/6) as CertNbr,
    CD.HonorRoll AS HonorRoll,
    Count(CD.HonorRoll) AS HRcnt
   from
    GLOBE.HR_CATEGORIES_ALL CN,
    GLOBE.HR_CATEGORIES_DISTINCT CD
   where
    CN."Honor Roll" = CD.HonorRoll AND
    CN."HRnbr"+5 >= (select MAX("HRnbr") from GLOBE.HR_CATEGORIES_ALL)
   group by CN."SchoolId", TRUNC((CN."HRnbr"+5)/6), CD.HonorRoll) X
where
 S.SchoolId = X.SchoolId
```

839 View HR_CERTIFICATES_NEW

839.1 SQL Query for view HR_CERTIFICATES_NEW

```
select
 M.SchoolId as "SchoolId",
 D."SchoolName" as "SchoolName",
 M.CertNbr as "CertNbr",
 M.HonorRoll as "HonorRoll",
 M.HRcnt as "NbrNew",
 M.EndMonths as "AllCertMonths"
from
 HR_CATEGORIES_CERT_MONTHS M,
 LETTERMERGE_DIRECTOR D
where
 M.SchoolId = D."SchoolId" AND
 M.CertNbr < (select Max(CertNbr)
              from GLOBE.HR_CATEGORIES_CERT_MONTHS)
```

840 View HR_CERTIFICATES_PREV

840.1 SQL Query for view HR_CERTIFICATES_PREV

```
select
 M.SchoolId as "SchoolId",
```

```
 D."SchoolName" as "SchoolName",
 M.CertNbr as "CertNbr",
 M.HonorRoll as "HonorRoll",
 M.HRcnt as "NbrNew",
 M.EndMonths as "AllCertMonths"
from
 HR_CATEGORIES_CERT_MONTHS M,
 CERTMERGE_DIRECTOR D
where
 M.SchoolId = D."SchoolId" AND
 M.CertNbr < (select Max(CertNbr)
              from GLOBE.HR_CATEGORIES_CERT_MONTHS)
```

841 View HR_CERTIFICATE_INFO_VU1

841.1 SQL Query for view HR_CERTIFICATE_INFO_VU1

```
select
 CS.SchoolId as "Vu1SchoolId",
 Atmos.HRcnt as "AtmosCntYear",
 Atmos.EndMonths as "AtmosMonthsYear",
 Clouds.HRcnt as "CloudsCntYear",
 Clouds.EndMonths as "CloudsMonthsYear",
 Hydro.HRcnt as "HydroCntYear",
 Hydro.EndMonths as "HydroMonthsYear",
 Climate.HRcnt as "ClimateCntYear",
 Climate.EndMonths as "ClimateMonthsYear",
 SoilMT.HRcnt as "SoilMTCntYear",
 SoilMT.EndMonths as "SoilMTMonthsYear",
 LcBio.HRcnt as "LcBioCntYear",
 LcBio.EndMonths as "LcBioMonthsYear",
 SoilChar.HRcnt as "SoilCharCntYear",
 SoilChar.EndMonths as "SoilCharMonthsYear",
 AdvAtmos.HRcnt as "AdvAtmosCntYear",
 AdvAtmos.EndMonths as "AdvAtmosMonthsYear",
 EarthSys.HRcnt as "EarthSysCntYear",
 EarthSys.EndMonths as "EarthSysMonthsYear",
 PanGlobe.HRcnt as "PanGlobeCntYear",
 PanGlobe.EndMonths as "PanGlobeMonthsYear"
from
 HR_CERTIFICATES CS,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Atmosphere') Atmos,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Clouds') Clouds,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Hydrology') Hydro,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Climate') Climate,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Soil Moisture and Temperature') SoilMT,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Land Cover/Biology') LcBio,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Soil Characterization') SoilChar,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Advanced Atmosphere') AdvAtmos,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Earth System Science') EarthSys,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_YEAR_MONTHS
  where HonorRoll = 'Pan GLOBE') PanGlobe
where
    CS.SchoolId = Atmos.SchoolId(+)
  AND
    CS.SchoolId = Clouds.SchoolId(+)
  AND
    CS.SchoolId = Hydro.SchoolId(+)
```

```
  AND
    CS.SchoolId = Climate.SchoolId(+)
  AND
    CS.SchoolId = SoilMT.SchoolId(+)
  AND
    CS.SchoolId = LcBio.SchoolId(+)
  AND
    CS.SchoolId = SoilChar.SchoolId(+)
  AND
    CS.SchoolId = AdvAtmos.SchoolId(+)
  AND
    CS.SchoolId = EarthSys.SchoolId(+)
  AND
    CS.SchoolId = PanGlobe.SchoolId(+)
```

842 View HR_CERTIFICATE_INFO_VU2

842.1 SQL Query for view HR_CERTIFICATE_INFO_VU2

```
select
 GS.SchoolId as "Vu2SchoolId",
 GS.SchoolName as "SchoolName",
 GS.AddressLine1 as "AddressLine1",
 GS.AddressLine2 as "AddressLine2",
 GS.City as "City",
 GS.StateProvCode as "StateProvCode",
 GS.PostalCode as "PostalCode",
 GS.CountryCode as "CountryCode",
 GC.GlobeName as "CountryName",
 GS.DirectorFullName as "DirectorFullName",
 GS.DirectorSalutation as "DirectorSalutation",
 GS.LifeCycleCode as "LifeCycleCode",
 GS.CycleChangeDate as "CycleChangeDate"
from
 HR_CERTIFICATES C,
 GLOBE.GLOBE_SCHOOL GS,
 GLOBE.GLOBE_COUNTRY GC
where
 C.SchoolId = GS.SchoolId AND
 GS.CountryCode = GC.CountryCode
```

843 View HR_CERTIFICATE_INFO_VU3

843.1 SQL Query for view HR_CERTIFICATE_INFO_VU3

```
select
 CS.SchoolId as "Vu3SchoolId",
 Atmos.HRcnt as "AtmosCntPrev",
 Atmos.EndMonths as "AtmosMonthsPrev",
 Clouds.HRcnt as "CloudsCntPrev",
 Clouds.EndMonths as "CloudsMonthsPrev",
 Hydro.HRcnt as "HydroCntPrev",
 Hydro.EndMonths as "HydroMonthsPrev",
 Climate.HRcnt as "ClimateCntPrev",
 Climate.EndMonths as "ClimateMonthsPrev",
 SoilMT.HRcnt as "SoilMTCntPrev",
 SoilMT.EndMonths as "SoilMTMonthsPrev",
 LcBio.HRcnt as "LcBioCntPrev",
 LcBio.EndMonths as "LcBioMonthsPrev",
 SoilChar.HRcnt as "SoilCharCntPrev",
 SoilChar.EndMonths as "SoilCharMonthsPrev",
 AdvAtmos.HRcnt as "AdvAtmosCntPrev",
 AdvAtmos.EndMonths as "AdvAtmosMonthsPrev",
 EarthSys.HRcnt as "EarthSysCntPrev",
 EarthSys.EndMonths as "EarthSysMonthsPrev",
 PanGlobe.HRcnt as "PanGlobeCntPrev",
 PanGlobe.EndMonths as "PanGlobeMonthsPrev"
from
 HR_CERTIFICATES CS,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Atmosphere') Atmos,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
```

```
  where HonorRoll = 'Clouds') Clouds,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Hydrology') Hydro,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Climate') Climate,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Soil Moisture and Temperature') SoilMT,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Land Cover/Biology') LcBio,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Soil Characterization') SoilChar,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Advanced Atmosphere') AdvAtmos,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Earth System Science') EarthSys,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_PREV_MONTHS
  where HonorRoll = 'Pan GLOBE') PanGlobe
where
    CS.SchoolId = Atmos.SchoolId(+)
  AND
    CS.SchoolId = Clouds.SchoolId(+)
  AND
    CS.SchoolId = Hydro.SchoolId(+)
  AND
    CS.SchoolId = Climate.SchoolId(+)
  AND
    CS.SchoolId = SoilMT.SchoolId(+)
  AND
    CS.SchoolId = LcBio.SchoolId(+)
  AND
    CS.SchoolId = SoilChar.SchoolId(+)
  AND
    CS.SchoolId = AdvAtmos.SchoolId(+)
  AND
    CS.SchoolId = EarthSys.SchoolId(+)
  AND
    CS.SchoolId = PanGlobe.SchoolId(+)
```

844 View HR_CERTIFICATE_TEACHERS

844.1 SQL Query for view HR_CERTIFICATE_TEACHERS

```
select
 CS.SchoolId,
 COUNT(GT.TeacherId) as TchrCnt,
 GLOBE.ListTeachers(CS.SchoolId) as TeacherList
from
 GLOBE.HR_CERTIFICATES CS,
 (select SchoolId, TeacherId
  from GLOBE.TEACHER_TO_SCHOOL
  where DateEnd is null) TS,
 (select TeacherId
  from GLOBE.GLOBE_TEACHER
  where LifeCycleCode <> 'I' OR LifeCycleCode IS NULL) GT
where
 CS.SchoolId = TS.SchoolId(+) AND
 TS.TeacherId = GT.TeacherId(+)
group by CS.SchoolId
```

845 View HR_LETTER_INFO_VU1

845.1 SQL Query for view HR_LETTER_INFO_VU1

```
select
 LS.SchoolId as "Vu1SchoolId",
 Atmos.HRcnt as "AtmosCntNew",
 Atmos.EndMonths as "AtmosMonthsNew",
```

```
 Clouds.HRcnt as "CloudsCntNew",
 Clouds.EndMonths as "CloudsMonthsNew",
 Hydro.HRcnt as "HydroCntNew",
 Hydro.EndMonths as "HydroMonthsNew",
 Climate.HRcnt as "ClimateCntNew",
 Climate.EndMonths as "ClimateMonthsNew",
 SoilMT.HRcnt as "SoilMTCntNew",
 SoilMT.EndMonths as "SoilMTMonthsNew",
 LcBio.HRcnt as "LcBioCntNew",
 LcBio.EndMonths as "LcBioMonthsNew",
 SoilChar.HRcnt as "SoilCharCntNew",
 SoilChar.EndMonths as "SoilCharMonthsNew",
 AdvAtmos.HRcnt as "AdvAtmosCntNew",
 AdvAtmos.EndMonths as "AdvAtmosMonthsNew",
 EarthSys.HRcnt as "EarthSysCntNew",
 EarthSys.EndMonths as "EarthSysMonthsNew",
 PanGlobe.HRcnt as "PanGlobeCntNew",
 PanGlobe.EndMonths as "PanGlobeMonthsNew"
from
 HR_LTRSCHOOLS LS,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Atmosphere') Atmos,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Clouds') Clouds,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Hydrology') Hydro,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Climate') Climate,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Soil Moisture and Temperature') SoilMT,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Land Cover/Biology') LcBio,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Soil Characterization') SoilChar,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Advanced Atmosphere') AdvAtmos,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Earth System Science') EarthSys,
 (select SchoolId, HRcnt, EndMonths
  from GLOBE.HR_CATEGORIES_NEW_MONTHS
  where HonorRoll = 'Pan GLOBE') PanGlobe
where
    LS.SchoolId = Atmos.SchoolId(+)
  AND
    LS.SchoolId = Clouds.SchoolId(+)
  AND
    LS.SchoolId = Hydro.SchoolId(+)
  AND
    LS.SchoolId = Climate.SchoolId(+)
  AND
    LS.SchoolId = SoilMT.SchoolId(+)
  AND
    LS.SchoolId = LcBio.SchoolId(+)
  AND
    LS.SchoolId = SoilChar.SchoolId(+)
  AND
    LS.SchoolId = AdvAtmos.SchoolId(+)
  AND
    LS.SchoolId = EarthSys.SchoolId(+)
  AND
    LS.SchoolId = PanGlobe.SchoolId(+)
```

846 View HR_LETTER_INFO_VU2

846.1 SQL Query for view HR_LETTER_INFO_VU2

```
select
 GS.SchoolId as "Vu2SchoolId",
```

```
 GS.SchoolName as "SchoolName",
 GS.AddressLine1 as "AddressLine1",
 GS.AddressLine2 as "AddressLine2",
 GS.City as "City",
 GS.StateProvCode as "StateProvCode",
 GS.PostalCode as "PostalCode",
 GS.CountryCode as "CountryCode",
 GC.GlobeName as "CountryName",
 GS.DirectorFullName as "DirectorFullName",
 GS.DirectorSalutation as "DirectorSalutation",
 GS.LifeCycleCode as "LifeCycleCode",
 GS.CycleChangeDate as "CycleChangeDate"
from
 HR_LTRSCHOOLS LS,
 GLOBE.GLOBE_SCHOOL GS,
 GLOBE.GLOBE_COUNTRY GC
where
 LS.SchoolId = GS.SchoolId AND
 GS.CountryCode = GC.CountryCode
```

847 View HR_LETTER_INFO_VU3

847.1 SQL Query for view HR_LETTER_INFO_VU3

```
select
 LS.SchoolId as "Vu3SchoolId",
 Atmos.HRcnt as "AtmosCntYear",
 Clouds.HRcnt as "CloudsCntYear",
 Hydro.HRcnt as "HydroCntYear",
 Climate.HRcnt as "ClimateCntYear",
 SoilMT.HRcnt as "SoilMTCntYear",
 LcBio.HRcnt as "LcBioCntYear",
 SoilChar.HRcnt as "SoilCharCntYear",
 AdvAtmos.HRcnt as "AdvAtmosCntYear",
 EarthSys.HRcnt as "EarthSysCntYear",
 PanGlobe.HRcnt as "PanGlobeCntYear"
from
 HR_LTRSCHOOLS LS,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Atmosphere' ) Atmos,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Clouds' ) Clouds,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Hydrology' ) Hydro,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Climate' ) Climate,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Soil Moisture and
Temperature' ) SoilMT,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Land Cover/Biology' ) LcBio,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Soil Characterization' )
SoilChar,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Advanced Atmosphere' )
AdvAtmos,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Earth System Science' )
EarthSys,
 ( select SchoolId, HRcnt from HR_CATEGORIES_YEAR_CNT where HonorRoll = 'Pan GLOBE' ) PanGlobe
where
 LS.SchoolId = Atmos.SchoolId (+) AND
 LS.SchoolId = Clouds.SchoolId (+) AND
 LS.SchoolId = Hydro.SchoolId (+) AND
 LS.SchoolId = Climate.SchoolId (+) AND
 LS.SchoolId = SoilMT.SchoolId (+) AND
 LS.SchoolId = LcBio.SchoolId (+) AND
 LS.SchoolId = SoilChar.SchoolId (+) AND
 LS.SchoolId = AdvAtmos.SchoolId (+) AND
 LS.SchoolId = EarthSys.SchoolId (+) AND
 LS.SchoolId = PanGlobe.SchoolId (+)
```

848 View HR_LETTER_INFO_VU4

848.1 SQL Query for view HR_LETTER_INFO_VU4

```
select
 LS.SchoolId as "Vu4SchoolId",
 Atmos.HRcnt as "AtmosCntAll",
 Clouds.HRcnt as "CloudsCntAll",
 Hydro.HRcnt as "HydroCntAll",
```

```
 Climate.HRcnt as "ClimateCntAll",
 SoilMT.HRcnt as "SoilMTCntAll",
 LcBio.HRcnt as "LcBioCntAll",
 SoilChar.HRcnt as "SoilCharCntAll",
 AdvAtmos.HRcnt as "AdvAtmosCntAll",
 EarthSys.HRcnt as "EarthSysCntAll",
 PanGlobe.HRcnt as "PanGlobeCntAll"
from
 HR_LTRSCHOOLS LS,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Atmosphere' ) Atmos,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Clouds' ) Clouds,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Hydrology' ) Hydro,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Climate' ) Climate,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Soil Moisture and Temperature'
) SoilMT,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Land Cover/Biology' ) LcBio,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Soil Characterization' )
SoilChar,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Advanced Atmosphere' )
AdvAtmos,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Earth System Science' )
EarthSys,
 ( select SchoolId, HRcnt from HR_CATEGORIES_ALL_CNT where HonorRoll = 'Pan GLOBE' ) PanGlobe
where
 LS.SchoolId = Atmos.SchoolId (+) AND
 LS.SchoolId = Clouds.SchoolId (+) AND
 LS.SchoolId = Hydro.SchoolId (+) AND
 LS.SchoolId = Climate.SchoolId (+) AND
 LS.SchoolId = SoilMT.SchoolId (+) AND
 LS.SchoolId = LcBio.SchoolId (+) AND
 LS.SchoolId = SoilChar.SchoolId (+) AND
 LS.SchoolId = AdvAtmos.SchoolId (+) AND
 LS.SchoolId = EarthSys.SchoolId (+) AND
 LS.SchoolId = PanGlobe.SchoolId (+)
```

849 View HR_LTRSCHOOLS

849.1 SQL Query for view HR_LTRSCHOOLS

```
select DISTINCT "SchoolId" as SchoolId
from HR_CATEGORIES_NEW
```

850 View HR_LTRSCHOOL_TEACHERS

850.1 SQL Query for view HR_LTRSCHOOL_TEACHERS

```
select
 LS.SchoolId,
 COUNT(GT.TeacherId) as TchrCnt,
 GLOBE.ListTeachers(LS.SchoolId) as TeacherList
from
 GLOBE.HR_LTRSCHOOLS LS,
 (select TeacherId, SchoolId
  from GLOBE.TEACHER_TO_SCHOOL
  where DateEnd is null) TS,
 (select TeacherId
  from GLOBE.GLOBE_TEACHER
  where LifeCycleCode <> 'I' OR LifeCycleCode is null) GT
where
 LS.SchoolId = TS.SchoolId (+) AND
 TS.TeacherId = GT.TeacherId (+)
group by LS.SchoolId
```

851 View HR_OVERALL_VU

851.1 SQL Query for view HR_OVERALL_VU

```
select
 SchoolId,
 TotalObs,
 FirstObs,
```

```
 LastObs,
 500*Floor(TotalObs/500) as Rung
from GLOBE.HONOR_ROLL_OVERALL
where
 TotalObs >= 500 AND TotalObs < 2500
UNION ALL
select
 SchoolId,
 TotalObs,
 FirstObs,
 LastObs,
 2500 as Rung
from GLOBE.HONOR_ROLL_OVERALL
where
 TotalObs >= 2500 AND TotalObs < 5000
UNION ALL
select
 SchoolId,
 TotalObs,
 FirstObs,
 LastObs,
 5000*Floor(TotalObs/5000) as Rung
from GLOBE.HONOR_ROLL_OVERALL
where
 TotalObs >= 5000
```

852 View HR_RUNGCHG

852.1 SQL Query for view HR_RUNGCHG

```
select A.SchoolId, A.TotalObs, A.Rung as NewRung, B.Rung as OldRung, A.FirstObs, A.LastObs
from GLOBE.HR_OVERALL_CURRENT A, GLOBE.HR_OVERALL_PREVIOUS B
where A.SchoolId = B.SchoolId ( + ) AND ( A.Rung > B.Rung OR B.SchoolId Is Null )
```

853 View HR_RUNGONLY_CERTIFICATE

853.1 SQL Query for view HR_RUNGONLY_CERTIFICATE

```
select A.SchoolId, A.TotalObs, A.NewRung, A.OldRung, A.FirstObs, A.LastObs
from HR_RUNGCHG A, GLOBE.HR_CERTIFICATES B
where A.SchoolId = B.SchoolId ( + ) AND B.SchoolId Is Null
```

854 View HR_RUNGONLY_LETTER

854.1 SQL Query for view HR_RUNGONLY_LETTER

```
select A.SchoolId, A.TotalObs, A.NewRung, A.OldRung, A.FirstObs, A.LastObs
from HR_RUNGCHG A, GLOBE.HR_LTRSCHOOLS B
where A.SchoolId = B.SchoolId ( + ) AND B.SchoolId Is Null
```

855 View HUMIDITY_VU1

855.1 SQL Query for view HUMIDITY_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(UTTimeMeasured,'MM') AS Month,
  'RH' as MeasureCode,
  Count(*) as RowCnt,
  COUNT(DryBulbTemp*WetBulbTemp*DECODE(HumidityMthd,1,1,null)) as SPsycCnt,
  COUNT(DryBulbTemp*WetBulbTemp*DECODE(HumidityMthd,2,1,null)) as HydromCnt,
  ROUND(AVG(HumidityPct),1) as HumPctAvg,
  ROUND(AVG(DewPoint),1) as DewPtAvg,
  MIN(UTTimeMeasured) AS BegDate,
  MAX(UTTimeMeasured) AS EndDate,
  MIN(UTtimereported) AS BegRDate,
  MAX(UTtimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-UTTimeMeasured),2) AS LagTimeAvg
FROM
```

```
  HUMIDITY
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(UTTimeMeasured,'MM')
```

856 View HUMIDITY_VU2

856.1 SQL Query for view HUMIDITY_VU2

```
SELECT
  HUMIDITY_VU1.*,
  (2*SPsycCnt+2*HydromCnt) as DixonCnt
FROM
  HUMIDITY_VU1
```

857 View HUMMINGBIRD_COUNTS_VU1

857.1 SQL Query for view HUMMINGBIRD_COUNTS_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(StartTimeUT,'MM') as Month,
  'HB' as MeasureCode,
  Count(*) as RowCnt,
  Count(AdultMaleCnt) as HAMCnt,
  Count(AdultFemaleCnt) as HAFCnt,
  Count(NoSexThroatCnt) as HNTCnt,
  Count(NoSexYoungCnt) as HNYCnt,
  Count(YoungMaleCnt) as HYMCnt,
  Count(UnusualCnt) as HUnCnt,
  MIN(StartTimeUT) as BegDate,
  MAX(StartTimeUT) as EndDate,
  MIN(UTtimereported) as BegRDate,
  MAX(UTtimereported) as EndRDate,
  ROUND(AVG(UTtimereported-StartTimeUT),2) as LagTimeAvg
from HUMMINGBIRD_COUNTS
where UTtimereported < TRUNC(SYSDATE,'DD')+22/24
group by SchoolId, SiteId, TRUNC(StartTimeUT,'MM')
```

858 View HUMMINGBIRD_COUNTS_VU2

858.1 SQL Query for view HUMMINGBIRD_COUNTS_VU2

```
select
  HUMMINGBIRD_COUNTS_VU1.*,
  (hamcnt+hafcnt+hntcnt+hnycnt+hymcnt+huncnt) as dixoncnt
from HUMMINGBIRD_COUNTS_VU1
```

859 View HUMMINGBIRD_FIRST_LAST

860 View HUMMINGBIRD_FREQUENCIES

860.1 SQL Query for view HUMMINGBIRD_FREQUENCIES

```
SELECT "Lat", "Long", "AvgFreq/Hr", "To be determined" From GLOBE.HUMMINGBIRD_COUNTS, SITE_LOCATION
```

861 View HUMMINGBIRD_SITE_VU1

861.1 SQL Query for view HUMMINGBIRD_SITE_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(VersionDate,'MM') as Month,
  'MD' as MeasureCode,
  Count(*) as RowCnt,
```

```
  COUNT(AtmosLink) as ALnkCnt,
  COUNT(MoistLink) as MLnkCnt,
  COUNT(HasFlowerbed) as HFbCnt,
  COUNT(HasFeeder) as HFdCnt,
  COUNT(HasNest) as HNstCnt,
  COUNT(NestFoundDate) as NFDtCnt,
  MIN(VersionDate) AS BegDate,
  MAX(VersionDate) AS EndDate,
  MIN(UTtimestamp) AS BegRDate,
  MAX(UTtimestamp) AS EndRDate,
  ROUND(AVG(UTtimestamp-VersionDate),2) AS LagTimeAvg
from HUMMINGBIRD_SITE
where UTtimestamp < TRUNC(SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(VersionDate,'MM')
```

862 View HUMMINGBIRD_SITE_VU2

862.1 SQL Query for view HUMMINGBIRD_SITE_VU2

```
select
  HUMMINGBIRD_SITE_VU1.*,
  (alnkcnt+mlnkcnt+hfbcnt+hfdcnt+hnstcnt+nfdtcnt) as dixoncnt
from HUMMINGBIRD_SITE_VU1
```

863 View Humidity_HR1

863.1 SQL Query for view Humidity_HR1

```
select
 SchoolId,
 SiteId,
 SolarDay
from
 GLOBE.HUMIDITY
where
 SolarDay >= TO_DATE('20010201','YYYYMMDD') AND
 HumidityPct is not null
group by SchoolId, SiteId, SolarDay
```

864 View Humidity_HR2

864.1 SQL Query for view Humidity_HR2

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(*) as GoodDays
from
 Humidity_HR1
group by
 SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

865 View Humidity_HR3

865.1 SQL Query for view Humidity_HR3

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Humidity_HR2 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
```

```
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.GoodDays) >= 4*30*(.70)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Humidity_HR2 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.GoodDays) >= 4*30*(.70)
```

866 View Humidity_HR4

866.1 SQL Query for view Humidity_HR4

```
select distinct
 HRnbr,
 SL.SchoolId,
 'Humidity' as Measurement,
 BeginMonth,
 EndMonth
from
 Humidity_HR3 HR3, GLOBE.SITE_LOCATION SL
where
 HRnbr >= 13 AND
 HR3.SchoolId = SL.SCHOOLID AND
 HR3.SiteId = SL.SITEID AND
 SL.DataSource IN (1,2)
```

867 View INFILTRATION_WS_DR1

868 View INFILTRATION_WS_DR2

868.1 SQL Query for view INFILTRATION_WS_DR2

```
select INFILTRATION_WS_DR1.*, 2*flocnt as DixonCnt
from INFILTRATION_WS_DR1
```

869 View INFILTRATION_WS_DU1

870 View INFILTRATION_WS_DU2

870.1 SQL Query for view INFILTRATION_WS_DU2

```
select INFILTRATION_WS_DU1.*, 2*flocnt as DixonCnt
from INFILTRATION_WS_DU1
```

871 View INFILTRATION_WS_MC1

871.1 SQL Query for view INFILTRATION_WS_MC1

```
select 'SF' as MeasureCode, COUNT(*) as RowCnt, COUNT(FlowRate) as FloCnt
from GLOBE.INFILTRATION_WS
```

872 View INFILTRATION_WS_MC2

872.1 SQL Query for view INFILTRATION_WS_MC2

```
select INFILTRATION_WS_MC1.*, 2*flocnt as DixonCnt
from INFILTRATION_WS_MC1
```

873 View INFILTRATION_WS_RPRT_VU

873.1 SQL Query for view INFILTRATION_WS_RPRT_VU

```
select Distinct as SchoolID, 'SF' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from INFILTRATION_WS
group by SchoolID
```

874 View INFILTRATION_WS_VU1

874.1 SQL Query for view INFILTRATION_WS_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'SF' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(FlowRate) AS FloCnt,
 MIN(DayMeasured) AS BegDate,
 MAX(DayMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-DayMeasured),2) AS LagTimeAvg
from GLOBE.INFILTRATION_WS
where UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

875 View INFILTRATION_WS_VU2

875.1 SQL Query for view INFILTRATION_WS_VU2

```
select INFILTRATION_WS_VU1.*, 2*flocnt as DixonCnt
from INFILTRATION_WS_VU1
```

876 View LAND_COVER_DR1

876.1 SQL Query for view LAND_COVER_DR1

```
select TRUNC(UTtimeReported,'DD') as Day, 'LC' as MeasureCode,
COUNT(DECODE(SiteCode,'LCS',1,'LCL',1,'LCN',1,Null)) as LCMUCCnt, COUNT(DECODE(SiteCode,'LCL',1,Null))
as QualMUCCnt, COUNT(DECODE(SiteCode,'LCN',1,Null)) as QuanMUCCnt,
COUNT(DECODE(SiteCode,'BIO',1,Null)) as BIMUCCnt, COUNT(DECODE(SiteCode,'ATM',1,Null)) as ATMUCCnt,
COUNT(DECODE(SiteCode,'SMS',1,Null)) as SMMUCCnt, COUNT(DECODE(SiteCode,'SCS',1,Null)) as SCMUCCnt,
COUNT(DECODE(SiteCode,'SWS',1,Null)) as SWMUCCnt, COUNT(DECODE(MUCsource,'G',1,Null)) as GndCnt,
COUNT(DECODE(MUCsource,'M',1,Null)) as MapCnt, COUNT(DECODE(MUCsource,'X',1,Null)) as ExtCnt,
COUNT(DECODE(MUCsource,'A',1,Null)) AS AndCntFROM GLOBE.LAND_COVER as LC, GLOBE.STUDY_SITE
SSWHERELC.SchoolId = SS.SchoolIdAND LC.SiteId = SS.SiteIdAND UTtimeReported < TRUNC (SYSDATE,'DD') +
22/24GROUP BY TRUNC as (UTtimeReported,'DD')
```

877 View LAND_COVER_DR2

877.1 SQL Query for view LAND_COVER_DR2

```
select LAND_COVER_DR1.*, (GndCnt + MapCnt) as DixonCnt
from LAND_COVER_DR1
```

878 View LAND_COVER_DU1

878.1 SQL Query for view LAND_COVER_DU1

```
select TRUNC(DayObserved,'DD') as Day, 'LC' as MeasureCode,
COUNT(DECODE(SiteCode,'LCS',1,'LCL',1,'LCN',1,Null)) as LCMUCCnt, COUNT(DECODE(SiteCode,'LCL',1,Null))
as QualMUCCnt, COUNT(DECODE(SiteCode,'LCN',1,Null)) as QuanMUCCnt,
COUNT(DECODE(SiteCode,'BIO',1,Null)) as BIMUCCnt, COUNT(DECODE(SiteCode,'ATM',1,Null)) as ATMUCCnt,
COUNT(DECODE(SiteCode,'SMS',1,Null)) as SMMUCCnt, COUNT(DECODE(SiteCode,'SCS',1,Null)) as SCMUCCnt,
COUNT(DECODE(SiteCode,'SWS',1,Null)) as SWMUCCnt, COUNT(DECODE(MUCsource,'G',1,Null)) as GndCnt,
COUNT(DECODE(MUCsource,'M',1,Null)) as MapCnt, COUNT(DECODE(MUCsource,'X',1,Null)) as ExtCnt,
COUNT(DECODE(MUCsource,'A',1,Null)) AS AndCntFROM GLOBE.LAND_COVER as LC, GLOBE.STUDY_SITE
SSWHERELC.SchoolId = SS.SchoolIdAND LC.SiteId = SS.SiteIdAND UTtimeReported < TRUNC (SYSDATE,'DD') +
22/24GROUP BY TRUNC as (DayObserved,'DD')
```

879 View LAND_COVER_DU2

879.1 SQL Query for view LAND_COVER_DU2

```
select LAND_COVER_DU1.*, (GndCnt + MapCnt) as DixonCnt
from LAND_COVER_DU1
```

880 View LAND_COVER_MC1

880.1 SQL Query for view LAND_COVER_MC1

```
select 'LC' as MeasureCode, COUNT(DECODE(SiteCode,'LCS',1,'LCL',1,'LCN',1,Null)) as LCMUCCnt,
COUNT(DECODE(SiteCode,'LCL',1,Null)) as QualMUCCnt, COUNT(DECODE(SiteCode,'LCN',1,Null)) as
QuanMUCCnt, COUNT(DECODE(SiteCode,'BIO',1,Null)) as BIMUCCnt, COUNT(DECODE(SiteCode,'ATM',1,Null)) as
ATMUCCnt, COUNT(DECODE(SiteCode,'SMS',1,Null)) as SMMUCCnt, COUNT(DECODE(SiteCode,'SCS',1,Null)) as
SCMUCCnt, COUNT(DECODE(SiteCode,'SWS',1,Null)) as SWMUCCnt, COUNT(DECODE(MUCsource,'G',1,Null)) as
GndCnt, COUNT(DECODE(MUCsource,'M',1,Null)) as MapCnt, COUNT(DECODE(MUCsource,'X',1,Null)) as ExtCnt,
COUNT(DECODE(MUCsource,'A',1,Null)) AS AndCntFROM GLOBE.LAND_COVER as LC,  GLOBE.STUDY_SITE
SSWHERELC.SchoolId = SS.SchoolIdAND LC.SiteId = SS.SiteId
```

881 View LAND_COVER_MC2

881.1 SQL Query for view LAND_COVER_MC2

```
select LAND_COVER_MC1.*, (GndCnt + MapCnt) as DixonCnt
from LAND_COVER_MC1
```

882 View LAND_COVER_RPRT_VU

882.1 SQL Query for view LAND_COVER_RPRT_VU

```
select Distinct as SchoolID, 'LC' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from LAND_COVER
group by SchoolID
```

883 View LAND_COVER_VU1

883.1 SQL Query for view LAND_COVER_VU1

```
select
 LC.SchoolId,
 LC.SiteId,
 TRUNC(DayObserved,'MM') AS Month,
 'LC' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT(DECODE(SiteCode,'LCS',1,'LCL',1,'LCN',1,Null)) AS LCMUCCnt,
 COUNT(DECODE(SiteCode,'LCL',1,Null)) AS QualMUCCnt,
 COUNT(DECODE(SiteCode,'LCN',1,Null)) AS QuanMUCCnt,
 COUNT(DECODE(SiteCode,'BIO',1,Null)) AS BIMUCCnt,
 COUNT(DECODE(SiteCode,'ATM',1,Null)) AS ATMUCCnt,
 COUNT(DECODE(SiteCode,'SMS',1,Null)) AS SMMUCCnt,
 COUNT(DECODE(SiteCode,'SCS',1,Null)) AS SCMUCCnt,
 COUNT(DECODE(SiteCode,'SWS',1,Null)) AS SWMUCCnt,
 COUNT(DECODE(MUCsource,'G',1,Null)) AS GndCnt,
```

```
 COUNT(DECODE(MUCsource,'M',1,Null)) AS MapCnt,
 COUNT(DECODE(MUCsource,'X',1,Null)) AS ExtCnt,
 COUNT(DECODE(MUCsource,'A',1,Null)) AS AndCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
FROM GLOBE.LAND_COVER LC, GLOBE.STUDY_SITE SS
WHERE LC.SchoolId = SS.SchoolId AND LC.SiteId = SS.SiteId AND
 UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(DayObserved,'MM')
```

884 View LAND_COVER_VU2

884.1 SQL Query for view LAND_COVER_VU2

```
select LAND_COVER_VU1.*, (GndCnt+MapCnt) as DixonCnt
from LAND_COVER_VU1
```

885 View LCBIO_HRDX

886 View LCBIO_OVERALL_HR1

886.1 SQL Query for view LCBIO_OVERALL_HR1

```
select hrnbr, schoolid, 'Multiple Tree Sites' as Method
from treebiom_hr3
union
select hrnbr, schoolid, 'Multiple Grass Sites' as Method
from grassbiom_hr3
union
select hrnbr, schoolid, 'Tree and Grass Sites' as Method
from treegrassbiom_hr4
union
select hrnbr, schoolid, 'Ten plus LCL Sites' as Method
from lcl_10_hr3
union
select a.hrnbr, a.schoolid, 'Five plus LCL Sites and Trees' as Method
from lcl_5_hr3 a, treebiom_hr2 b
where a.HrNbr = b.HrNbr AND a.SchoolId = b.SchoolId
union
select a.hrnbr, a.schoolid, 'Five plus LCL Sites and Grass'
from lcl_5_hr3 a, grassbiom_hr2 b
where a.HrNbr = b.HrNbr AND a.SchoolId = b.SchoolId
```

887 View LCBIO_OVERALL_HR2

887.1 SQL Query for view LCBIO_OVERALL_HR2

```
select HRnbr, SchoolId, TRUNC(TO_DATE('19980601','YYYYMMDD')+2*(HRnbr-1)*31,'MM') as BeginMonth,
TRUNC(TO_DATE('19980601','YYYYMMDD')+(2*(HRnbr+5)-1)*31,'MM') as EndMonth, Min(Method) as Method1,
Max(Method) as Method2
from LCBIO_OVERALL_HR1
group by HRnbr, SchoolId, TRUNC(TO_DATE('19980601','YYYYMMDD')+2*(HRnbr-1)*31,'MM'),
TRUNC(TO_DATE('19980601','YYYYMMDD')+(2*(HRnbr+5)-1)*31,'MM')
```

888 View LCBIO_OVERALL_HR3

888.1 SQL Query for view LCBIO_OVERALL_HR3

```
select A.*, SUM(B.DixonCnt) as DixonCnt
from LCBIO_OVERALL_HR2 A, GLOBE.DIXON_SUMMARY_GROUPED B
where A.SchoolId = B.SchoolId AND B.GroupCode = 'LcBi' AND B.Month BETWEEN A.BeginMonth AND A.EndMonth
group by A.HRnbr, A.SchoolId, A.BeginMonth, A.EndMonth, A.Method1, A.Method2
```

889 View LCL_10_HR3

889.1 SQL Query for view LCL_10_HR3

```
(select
 Hrnbr, SchoolId
 From lcl_hr2
 Group by Hrnbr, SchoolId
 Having COUNT(SchoolId) >= 10
)
```

890 View LCL_5_HR3

890.1 SQL Query for view LCL_5_HR3

```
(select
 Hrnbr, SchoolId
 From lcl_hr2
 Group by Hrnbr, SchoolId
 Having COUNT(SchoolId) >= 5 AND COUNT(SchoolId) < 10
)
```

891 View LCL_HR1

891.1 SQL Query for view LCL_HR1

```
(select a.schoolid, a.siteid, TRUNC(a.dayobserved) AS Day
 from globe.land_cover a, globe.study_site b
 WHERE A.SchoolId = B.SchoolId AND
 A.SiteId = B.SiteId AND B.SiteCode = 'LCL' AND
 A.DayObserved >=  TO_DATE('1998 06 01','YYYY MM DD')
 GROUP BY A.Schoolid, A.SiteId, TRUNC(a.dayobserved))
```

892 View LCL_HR1_LocP

892.1 SQL Query for view LCL_HR1_LocP

```
select A.SchoolId, A.SiteId, A.Day,
DECODE(B.DataSource,0,'Unchecked',1,'GPS',2,'Other') as LocMethod,
Count(C.PhotoId) as Photos
from LCL_HR1 A, GLOBE.SITE_LOCATION B, GLOBE.SITE_PHOTO C
where
 A.SchoolId = B.SchoolId AND A.SchoolId = C.SchoolId AND
 A.SiteId = B.SiteId AND A.SiteId = C.SiteId
group by A.SchoolId, A.SiteId, A.Day,
DECODE(B.DataSource,0,'Unchecked',1,'GPS',2,'Other')
```

893 View LCL_HR2

893.1 SQL Query for view LCL_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From LCL_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
```

```
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From LCL_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From LCL_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From LCL_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From LCL_HR1_LocP A
Where A.LocMethod In ('GPS','Other')
AND A.Photos >= 4
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
```

```
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From LCL_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

894 View LETTERMERGE_DIRECTOR_VU

894.1 SQL Query for view LETTERMERGE_DIRECTOR_VU

```
select
 LS.SchoolId as "SchoolId",
 LS.TchrCnt as "TchrCnt",
 LS.TeacherList as "TeacherList",
 HR_LETTER_INFO_VU2.*,
 HR_LETTER_INFO_VU1.*,
 HR_LETTER_INFO_VU3.*,
 HR_LETTER_INFO_VU4.*,
 RC.NewRung as "NewRung",
 RC.OldRung as "OldRung",
 RM.BegDate as "FirstObs",
 RM.EndDate as "LastObs",
 RM.DixonCnt as "TotalObs",
 ROUND(MONTHS_BETWEEN(RM.EndDate, RM.BegDate)) as "ObsMonths",
 ROUND(RM.DixonCnt/MONTHS_BETWEEN(RM.EndDate, RM.BegDate)) as "ObsRate",
 AC."AllCnt",
 AC."AllYearCnt",
 AC."AllNewCnt",
 AC."AllPrevCnt"
from
 GLOBE.HR_LTRSCHOOL_TEACHERS LS,
 GLOBE.HR_LETTER_INFO_VU1,
 GLOBE.HR_LETTER_INFO_VU2,
 GLOBE.HR_LETTER_INFO_VU3,
 GLOBE.HR_LETTER_INFO_VU4,
 HR_ALLCNT_SUMS AC,
 GLOBE.HR_RUNGCHG RC,
 GLOBE.DIXON_SUMMARY_CNT RM
where
 LS.SchoolId = HR_LETTER_INFO_VU1."Vu1SchoolId" AND
 LS.SchoolId = HR_LETTER_INFO_VU2."Vu2SchoolId" AND
 LS.SchoolId = HR_LETTER_INFO_VU3."Vu3SchoolId" AND
 LS.SchoolId = HR_LETTER_INFO_VU4."Vu4SchoolId" AND
 LS.SchoolId = AC."SchoolId" AND
 LS.SchoolId = RC.SchoolId (+) AND
 LS.SchoolId = RM.SchoolId
```

895 View LETTERMERGE_RUNGONLY_VU

895.1 SQL Query for view LETTERMERGE_RUNGONLY_VU

```
select
 RL.SchoolId as "SchoolId",
 RL.TotalObs as "TotalObs",
 RL.NewRung as "NewRung",
 RL.OldRung as "OldRung",
 RL.FirstObs as "FirstObs",
 RL.LastObs as "LastObs",
 ROUND(MONTHS_BETWEEN(RL.LastObs, RL.FirstObs)) as "ObsMonths",
 ROUND(RL.TotalObs/MONTHS_BETWEEN(RL.LastObs, RL.FirstObs)) as "ObsRate",
 NVL(AC."AllCnt",0) as "AllCnt",
```

```
 NVL(AC."AllYearCnt",0) as "AllYearCnt",
 NVL(AC."AllNewCnt",0) as "AllNewCnt",
 NVL(AC."AllPrevCnt",0) as "AllPrevCnt",
 GS.SchoolName as "SchoolName",
 GS.AddressLine1 as "AddressLine1",
 GS.AddressLine2 as "AddressLine2",
 GS.City as "City",
 GS.StateProvCode as "StateProvCode",
 GS.PostalCode as "PostalCode",
 GS.CountryCode as "CountryCode",
 GC.GlobeName as "CountryName",
 GS.DirectorFullName as "DirectorFullName",
 GS.DirectorSalutation as "DirectorSalutation",
 GS.LifeCycleCode as "LifeCycleCode",
 GS.CycleChangeDate as "CycleChangeDate"
from
 GLOBE.HR_RUNGONLY_LETTER RL,
 GLOBE.GLOBE_SCHOOL GS,
 HR_ALLCNT_SUMS AC,
 GLOBE.GLOBE_COUNTRY GC
where
 RL.SchoolId = GS.SchoolId  AND
 RL.SchoolId = AC."SchoolId"(+) AND
 GS.Countrycode  =  GC.CountryCode
```

896 View LETTERMERGE_TEACHER_VU

896.1 SQL Query for view LETTERMERGE_TEACHER_VU

```
select
 LS.SchoolId as "SchoolId",
 GT.TeacherId as "TeacherId",
 GT.TeacherPrefix as "Pre",
 GT.TeacherOtherNames as "FirstName",
 GT.TeacherSurname as "LastName",
 GT.TeacherSuffix as "Suf",
 GT.InternalAddress as "InternalAddress",
 GT.LifeCycleCode as "TeachLCcode",
 GT.CycleChangeDate as "TeachLCDate",
 HR_LETTER_INFO_VU2.*,
 HR_LETTER_INFO_VU1.*,
 HR_LETTER_INFO_VU3.*,
 HR_LETTER_INFO_VU4.*,
 RC.NewRung as "NewRung",
 RC.OldRung as "OldRung",
 RM.BegDate as "FirstObs",
 RM.EndDate as "LastObs",
 RM.DixonCnt as "TotalObs",
 ROUND(MONTHS_BETWEEN(RM.EndDate, RM.BegDate)) as "ObsMonths",
 ROUND(RM.DixonCnt/MONTHS_BETWEEN(RM.EndDate, RM.BegDate)) as "ObsRate",
 AC."AllCnt",
 AC."AllYearCnt",
 AC."AllNewCnt",
 AC."AllPrevCnt"
from
 GLOBE.HR_LTRSCHOOLS LS,
 GLOBE.TEACHER_TO_SCHOOL TS,
 GLOBE.GLOBE_TEACHER GT,
 GLOBE.HR_LETTER_INFO_VU1,
 GLOBE.HR_LETTER_INFO_VU2,
 GLOBE.HR_LETTER_INFO_VU3,
 GLOBE.HR_LETTER_INFO_VU4,
 HR_ALLCNT_SUMS AC,
 GLOBE.HR_RUNGCHG RC,
 GLOBE.DIXON_SUMMARY_CNT RM
where
 LS.SchoolId = TS.SchoolId  AND
 TS.TeacherId  =  GT.TeacherId  AND
 TS.DateEnd  IS  NULL  AND
 GT.LifeCycleCode  <>  'I'  AND
 LS.SchoolId  =  HR_LETTER_INFO_VU1."Vu1SchoolId" AND
 LS.SchoolId  =  HR_LETTER_INFO_VU2."Vu2SchoolId" AND
 LS.SchoolId  =  HR_LETTER_INFO_VU3."Vu3SchoolId" AND
 LS.SchoolId  =  HR_LETTER_INFO_VU4."Vu4SchoolId" AND
 LS.SchoolId  =  AC."SchoolId" AND
 LS.SchoolId  =  RC.SchoolId (+) AND
 LS.SchoolId  =  RM.SchoolId
```

897 View LILAC_METADATA_VU1

897.1 SQL Query for view LILAC_METADATA_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(UTTimeReported, 'MM') AS Month,
  'LB' AS Measurecode,
  COUNT(*) AS RowCnt,
  COUNT(DatePlanted) AS DPcnt,
  COUNT(HeightAtPlanting) AS HtCnt,
  MIN(UTTimeReported) AS BegDate,
  MAX(UTTimeReported) AS EndDate,
  MIN(UTTimeReported) AS BegRDate,
  MAX(UTTimeReported) AS EndRDate,
  ROUND(NVL(AVG(UTTimeReported-DatePlanted),9999),2) AS LagTimeAvg
FROM
  LILAC_METADATA
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(UTTimeReported,'MM')
```

898 View LILAC_METADATA_VU2

898.1 SQL Query for view LILAC_METADATA_VU2

```
SELECT
  LILAC_METADATA_VU1.*,
  (DPCnt+HtCnt) AS DixonCnt
FROM
  LILAC_METADATA_VU1
```

899 View LILAC_PHENOPHASE_VU

899.1 SQL Query for view LILAC_PHENOPHASE_VU

```
select SchoolId, SiteId, LilacId, Year, FirstLeafDate-TRUNC(FirstLeafDate,'YYYY') as FirstLeafDay,
FullLeafDate-TRUNC(FullLeafDate,'YYYY') as FullLeafDay, FirstBloomDate-TRUNC(FirstBloomDate,'YYYY') as
FirstBloomDay, FullBloomDate-TRUNC(FullBloomDate,'YYYY') as FullBloomDay,
EndBloomDate-TRUNC(EndBloomDate,'YYYY') as EndBloomDay, Comments
from GLOBE.LILAC_PHENOPHASE
```

900 View LILAC_PHENOPHASE_VU1

900.1 SQL Query for view LILAC_PHENOPHASE_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(NVL(EndBloomDate,UTTimeReported),'MM') AS Month,
  'LB' AS MeasureCode,
  COUNT(*) AS RowCnt,
  COUNT(FirstLeafDate)     AS L1BCnt,
  COUNT(FirstLeafLastObs)  AS L1PCnt,
  COUNT(FullLeafDate)      AS LFBCnt,
  COUNT(FullLeafLastObs)   AS LFPCnt,
  COUNT(FirstBloomDate)    AS B1BCnt,
  COUNT(FirstBloomLastObs) AS B1PCnt,
  COUNT(FullBloomDate)     AS BFBCnt,
  COUNT(FullBloomLastObs)  AS BFPCnt,
  COUNT(EndBloomDate)      AS BEBCnt,
  COUNT(EndBloomLastObs)   AS BEPCnt,
  MIN(NVL(FirstLeafDate,UTTimeReported)) AS BegDate,
  MAX(NVL(EndBloomDate, UTTimeReported)) AS EndDate,
  MIN(UTTimeReported) AS BegRDate,
  MAX(UTTimeReported) AS EndRDate,
```

```
  ROUND(AVG(UTTimeReported-NVL(EndBloomDate,UTTimeReported)),2)
     AS LAGTIMEAVG
FROM
  LILAC_PHENOPHASE
WHERE
   UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(NVL(EndBloomDate,UTTimeReported), 'MM')
```

901 View LILAC_PHENOPHASE_VU2

901.1 SQL Query for view LILAC_PHENOPHASE_VU2

```
SELECT
  LILAC_PHENOPHASE_VU1.*,
  (L1BCnt+L1PCnt+LFBCnt+LFPCnt+B1BCnt+B1PCnt+BFBCnt+BFPCnt+BEBCnt+BEPCnt)
       AS DixonCnt
FROM
  LILAC_PHENOPHASE_VU1
```

902 View LILAC_SUMMARY

902.1 SQL Query for view LILAC_SUMMARY

```
select
 LP.Year,
 Round(SL.Latitude) as Lat,
 LM.LilacTypeCode as Type,
 Round(AVG(FirstLeafDate-TRUNC(FirstLeafDate,'YYYY'))) as FirstLeafAvg,
 Round(AVG(FullLeafDate-TRUNC(FirstLeafDate,'YYYY'))) as FullLeafAvg,
 Round(AVG(FirstBloomDate-TRUNC(FirstLeafDate,'YYYY'))) as FirstBloomAvg,
 Round(AVG(FullBloomDate-TRUNC(FirstLeafDate,'YYYY'))) as FullBloomAvg,
 Round(AVG(EndBloomDate-TRUNC(FirstLeafDate,'YYYY'))) as EndBloomAvg
from
 GLOBE.LILAC_METADATA LM,
 GLOBE.SITE_LOCATION SL,
 GLOBE.LILAC_PHENOPHASE LP
where
 LM.SchoolId = SL.SchoolIdand and
 LM.SiteId = SL.SiteIdand and
 LM.SchoolId = LP.SchoolId and
 LM.siteId = LP.SiteId
GROUP BY LP.Year, Round(SL.Latitude), LM.LilacTypeCode
```

903 View LOGGER_HR0

903.1 SQL Query for view LOGGER_HR0

```
select
  TL.SchoolId,
  TL.SiteId,
  TRUNC(TL.UTtimeMeasured) as SolarDay
from
  GLOBE.TEMPERATURE_LOGGER TL,
  GLOBE.SITE_LOCATION SL
where
  TL.SchoolId = SL.SchoolId AND
  TL.SiteId = SL.SiteId AND
  SL.DataSource in (1,2) AND
  TL.UTtimeMeasured >= TO_DATE('20000601','YYYYMMDD') AND
  TL.SoilTemp10cm is not null AND
  (TL.SoilTemp5cm is not null OR SoilTemp50cm is not null)
```

904 View Lilac_HR1

904.1 SQL Query for view Lilac_HR1

```
select
 P.SchoolId,
 P.SiteId,
```

```
 P.Year,
 Min(P.FirstLeafDate) as BeginDay,
 Max(P.FullBloomDate) as EndDay
from
 GLOBE.LILAC_PHENOPHASE P,
 GLOBE.SITE_LOCATION SL
where
 P.SchoolId = SL.SchoolId AND
 P.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 P.FirstLeafDate is not null AND
 P.FirstBloomDate is not null AND
 P.FullBloomDate is not null
group by P.SchoolId, P.SiteId, P.Year
having Count(*) >= 2
```

905 View MCODE_COUNTS_VU1

905.1 SQL Query for view MCODE_COUNTS_VU1

```
(SELECTMeasureCode,DixonCntfrom AIR_TEMP_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PRECIP_RAIN_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PRECIP_SOLID_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom CLOUD_OBS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom GRASS_BIOMETRY_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom TREE_BIOMETRY_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom LAND_COVER_MC2)
union ALL(SELECT'MD' as MeasureCode,ExtCnt as DixonCntfrom LAND_COVER_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom GRASSBIOM_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom TREEBIOM_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PHENOLOGY_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PHENOLOGY_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PHENOLOGY_SITE_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom ATMOSPHERE_SITE_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SOILMOISTURE_SITE_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SOILCHAR_SITE_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SURFACEWATER_SITE_MC2)
union ALL(SELECT'MD' as MeasureCode,DixonCntfrom SITE_PROXIMITY_MC2)
union ALL(SELECT'MD' as MeasureCode,DixonCntfrom SITE_LOCATION_MC2)
union ALL(SELECT'LC' as MeasureCode,(QualSPCnt+QuanSPCnt+BISPCnt) as DixonCntFROM SITE_PHOTO_MC2WHERE
(QualSPCnt+QuanSPCnt+BISPCnt)>0)
union ALL(SELECT'MD' as MeasureCode,(ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt) as DixonCntFROM
SITE_PHOTO_MC2WHERE (ATSPcnt+SMSPcnt+SCSPcnt+SWSPcnt+PHSPcnt+SCHSPcnt)>0)
union ALL(SELECTMeasureCode,DixonCntfrom SOIL_CHAR_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom DENSITY_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PARTICLE96_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom PARTICLE_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom FERTILITY_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom ACIDITY_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SOIL_INFILTRATION_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom INFILTRATION_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SMGYPS_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SMGRAV_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SOIL_TEMP_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom TURBIDITY_WS_MC2)
union ALL(SELECTMeasureCode,DixonCntfrom SURFACE_WATER_MC2)
```

906 View MCODE_COUNTS_VU2

906.1 SQL Query for view MCODE_COUNTS_VU2

```
select MeasureCode, SUM(DixonCnt) as DixonCnt
from MCODE_COUNTS_VU1
group by MeasureCode
```

907 View Nitrate_HR1

907.1 SQL Query for view Nitrate_HR1

```
select
 SW.SchoolId,
 SW.SiteId,
```

```
 SW.DayMeasured,
 SW.NitriteNitrate,
 SW.NitriteOnly
from
 GLOBE.SURFACE_WATER SW, GLOBE.SITE_LOCATION SL
where
 SW.SchoolId = SL.SchoolId AND
 SW.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 SW.DayMeasured >= TO_DATE('20010201','YYYYMMDD') AND
 SW.NitriteNitrate is not null
```

908 View Nitrate_HR2

908.1 SQL Query for view Nitrate_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.DayMeasured,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 Count(T.DayMeasured) as GoodDays
from
 Nitrate_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.DayMeasured) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 Count(T.DayMeasured) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 Count(T.DayMeasured) as GoodDays
from
 Nitrate_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.DayMeasured) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 Count(T.DayMeasured) >= 8
```

909 View OZONE_VU1

909.1 SQL Query for view OZONE_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(UTStartTime,'MM') AS Month,
  'OZ' AS MeasureCode,
  COUNT(*) AS RowCnt,
  COUNT(AirTempStart) AS ATbCnt,
  COUNT(AirTempStop) AS ATeCnt,
  COUNT(WindDirStart) AS WDbCnt,
  COUNT(WindDirStop) AS WDeCnt,
  COUNT(OzonePPB) AS OzPPBCnt,
  SUM(DECODE(OzoneTMAflag,'M',1,0)) OzMflgCnt,
  ROUND(AVG(OzonePPB)) AS OzPPBAvg,
  MIN(UTStartTime) AS BegDate,
  MAX(UTStartTime) AS EndDate,
  MIN(UTtimereported) AS BegRDate,
  MAX(UTtimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-UTStartTime),2) AS LagTimeAvg
FROM
  OZONE
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(UTStartTime,'MM')
```

910 View OZONE_VU2

910.1 SQL Query for view OZONE_VU2

```
SELECT
  OZONE_VU1.*,
  (ATbCnt+ATeCnt+WDbCnt+WDeCnt+OzPPBCnt+OzMflgCnt) AS DixonCnt
FROM
  OZONE_VU1
```

911 View Ozone_HR1

911.1 SQL Query for view Ozone_HR1

```
select
 SchoolId,
 SiteId,
 SolarDay
from
 GLOBE.OZONE
where
 SolarDay >= TO_DATE('20010201','YYYYMMDD') AND
 OzonePPB is not null AND
 WindDirStart is not null AND
 WindDirStop is not null
group by SchoolId, SiteId, SolarDay
```

912 View Ozone_HR2

912.1 SQL Query for view Ozone_HR2

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(*) as GoodDays
from
 Ozone_HR1
group by
 SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

913 View Ozone_HR3

913.1 SQL Query for view Ozone_HR3

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Ozone_HR2 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.GoodDays) >= 4*30*(.70)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.GoodDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 Ozone_HR2 T
```

```
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.GoodDays) >= 4*30*(.70)
```

914 View Ozone_HR4

914.1 SQL Query for view Ozone_HR4

```
select distinct
 HRnbr,
 SL.SchoolId,
 'Ozone' as Measurement,
 BeginMonth,
 EndMonth
from
 Ozone_HR3 HR3, GLOBE.SITE_LOCATION SL
where
 HRnbr >= 13 AND
 HR3.SchoolId = SL.SCHOOLID AND
 HR3.SiteId = SL.SITEID AND
 SL.DataSource IN (1,2)
```

915 View PAN_GLOBE_HR1

915.1 SQL Query for view PAN_GLOBE_HR1

```
select HRnbr, SchoolId, Measurement from PG_ATMOS_HR2
UNION ALL
select HRnbr, SchoolId, Measurement from PG_HYDRO_HR2
UNION ALL
select HRnbr, SchoolId, Measurement from PG_SOIL_HR2
UNION ALL
select HRnbr, SchoolId, Measurement from PG_LAND_HR2
```

916 View PAN_GLOBE_HR2

916.1 SQL Query for view PAN_GLOBE_HR2

```
select
 HRnbr,
 SchoolId,
 1 as DixonCnt,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+ 2*(HRnbr-1)*31,'MM') as BeginMonth,
 TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM') as EndMonth
from
 PAN_GLOBE_HR1
group by HRnbr, SchoolId
having Count(*) = 4
```

917 View PARTICLE96_WS_DR1

918 View PARTICLE96_WS_DR2

918.1 SQL Query for view PARTICLE96_WS_DR2

```
select PARTICLE96_WS_DR1.*, (3*PS96SampCnt) as DixonCnt
from PARTICLE96_WS_DR1
```

919 View PARTICLE96_WS_DU1

920 View PARTICLE96_WS_DU2

920.1 SQL Query for view PARTICLE96_WS_DU2

```
select PARTICLE96_WS_DU1.*, (3*PS96SampCnt) as DixonCnt
from PARTICLE96_WS_DU1
```

921 View PARTICLE96_WS_MC1

921.1 SQL Query for view PARTICLE96_WS_MC1

```
select 'SC' as MeasureCode, COUNT(W.SandPct96+W.SiltPct96+W.ClayPct96) as PS96SampCnt
from GLOBE.PARTICLE96_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.H  or  izonNbr = S.H or izonNbr
```

922 View PARTICLE96_WS_MC2

922.1 SQL Query for view PARTICLE96_WS_MC2

```
select PARTICLE96_WS_MC1.*, (3*PS96SampCnt) as DixonCnt
from PARTICLE96_WS_MC1
```

923 View PARTICLE96_WS_RPRT_VU

923.1 SQL Query for view PARTICLE96_WS_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from PARTICLE96_WS
group by SchoolID
```

924 View PARTICLE96_WS_VU1

924.1 SQL Query for view PARTICLE96_WS_VU1

```
select
 W.SchoolId,
 W.SiteId,
 TRUNC(S.UTtimeMeasured,'MM') AS Month,
 'SC' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT(W.SandPct96+W.SiltPct96+W.ClayPct96) AS PS96SampCnt,
 MIN(S.UTtimeMeasured) AS BegDate,
 MAX(S.UTtimeMeasured) AS EndDate,
 MIN(S.UTtimeReported) AS BegRDate,
 MAX(S.UTtimeReported) AS EndRDate,
 ROUND(AVG(W.UTtimeReported-S.UTtimeMeasured),2) AS LagTimeAvg
from
 GLOBE.PARTICLE96_WS W, GLOBE.SOIL_CHAR S
where
 W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and
 W.HorizonNbr = S.HorizonNbr and
 W.UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 W.SchoolId, W.SiteId, TRUNC(S.UTtimeMeasured,'MM')
```

925 View PARTICLE96_WS_VU2

925.1 SQL Query for view PARTICLE96_WS_VU2

```
select PARTICLE96_WS_VU1.*, (3*PS96SampCnt) as DixonCnt
from PARTICLE96_WS_VU1
```

926 View PARTICLE_WS_DR1

927 View PARTICLE_WS_DR2

927.1 SQL Query for view PARTICLE_WS_DR2

```
select PARTICLE_WS_DR1.*, (3*PSUSDASampCnt+3*PSISSSSampCnt) as DixonCnt
from PARTICLE_WS_DR1
```

928 View PARTICLE_WS_DU1

929 View PARTICLE_WS_DU2

929.1 SQL Query for view PARTICLE_WS_DU2

```
select PARTICLE_WS_DU1.*, (3*PSUSDASampCnt+3*PSISSSSampCnt) as DixonCnt
from PARTICLE_WS_DU1
```

930 View PARTICLE_WS_MC1

930.1 SQL Query for view PARTICLE_WS_MC1

```
select 'SC' as MeasureCode, COUNT(W.SandPctUSDA+W.SiltPctUSDA+W.ClayPct) as PSUSDASampCnt,
COUNT(W.SandPctISSS+W.SiltPctISSS+W.ClayPct) as PSISSSSampCnt
from GLOBE.PARTICLE_WS W, GLOBE.SOIL_CHAR S
where W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and W.H  or  izonNbr = S.H or izonNbr
```

931 View PARTICLE_WS_MC2

931.1 SQL Query for view PARTICLE_WS_MC2

```
select PARTICLE_WS_MC1.*, (3*PSUSDASampCnt+3*PSISSSSampCnt) as DixonCnt
from PARTICLE_WS_MC1
```

932 View PARTICLE_WS_RPRT_VU

932.1 SQL Query for view PARTICLE_WS_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from PARTICLE_WS
group by SchoolID
```

933 View PARTICLE_WS_VU1

933.1 SQL Query for view PARTICLE_WS_VU1

```
select
 W.SchoolId,
 W.SiteId,
 TRUNC(S.UTtimeMeasured,'MM') AS Month,
 'SC' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(W.SandPctUSDA+W.SiltPctUSDA+W.ClayPct) AS PSUSDASampCnt,
 COUNT(W.SandPctISSS+W.SiltPctISSS+W.ClayPct) AS PSISSSSampCnt,
 MIN(S.UTtimeMeasured) AS BegDate,
 MAX(S.UTtimeMeasured) AS EndDate,
 MIN(S.UTtimeReported) AS BegRDate,
 MAX(S.UTtimeReported) AS EndRDate,
 ROUND(AVG(W.UTtimeReported-S.UTtimeMeasured),2) AS LagTimeAvg
from
 GLOBE.PARTICLE_WS W, GLOBE.SOIL_CHAR S
where
 W.SchoolId = S.SchoolId and W.SiteId = S.SiteId and
 W.HorizonNbr = S.HorizonNbr and
 W.UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 W.SchoolId, W.SiteId, TRUNC(S.UTtimeMeasured,'MM')
```

934 View PARTICLE_WS_VU2

934.1 SQL Query for view PARTICLE_WS_VU2

```
select PARTICLE_WS_VU1.*, (3*PSUSDASampCnt+3*PSISSSSampCnt) as DixonCnt
from PARTICLE_WS_VU1
```

935 View PG_ATMOS_HR1

935.1 SQL Query for view PG_ATMOS_HR1

```
select HRnbr, SchoolId, 'Precip' as Measurement
from PG_Precip_HR2
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'AirTemp' as Measurement
from ATMOS_TEMP_HR2
where HRnbr >=7
UNION
select HRnbr, SchoolId, 'Clouds' as Measurement
from GLOBE.HR_CLOUDS_CURRENT
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'Acidity' as Measurement
from GLOBE.Acidity_HR4
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'Humidity' as Measurement
from GLOBE.Humidity_HR4
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'Ozone' as Measurement
from GLOBE.Ozone_HR4
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'Aerosol' as Measurement
from GLOBE.Aerosol_HR4
where HRnbr >= 7
```

936 View PG_ATMOS_HR2

936.1 SQL Query for view PG_ATMOS_HR2

```
select
 HRnbr,
 SchoolId,
 'Atmos' as Measurement
from PG_ATMOS_HR1
where HRnbr >= 7
group by HRnbr, SchoolId
having Count(*) >= 3
```

937 View PG_DissolvedOxy_HR1

937.1 SQL Query for view PG_DissolvedOxy_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(DissolvedOxy) AS OxyCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR DissolvedOxy is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(DissolvedOxy) > 0
```

938 View PG_DissolvedOxy_HR2

938.1 SQL Query for view PG_DissolvedOxy_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_DissolvedOxy_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_DissolvedOxy_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

939 View PG_Grass_HR1

939.1 SQL Query for view PG_Grass_HR1

```
select distinct
 A.SchoolId,
 A.SiteId,
 A.Daymeasured AS Day
from
 globe.grass_biometry A,
 globe.site_location B,
 globe.land_cover C
where
 A.SchoolId = B.SchoolId AND
 A.SiteId = B.SiteId AND
 A.GreenBiomassAvg is not null AND
 A.BrownBiomassAvg is not null AND
 A.GreenSquares is not null AND
 A.BrownSquares is not null AND
 B.DataSource in (1,2) AND
 A.SchoolId = C.SchoolId AND
 A.SiteId = C.SiteId AND
 A.DayMeasured >= TO_DATE('1998 06 01','YYYY MM DD') AND
 C.MUCCode LIKE '4%'
```

940 View PG_Grass_HR2

940.1 SQL Query for view PG_Grass_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId
from PG_Grass_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
```

```
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId
from PG_Grass_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId
from PG_Grass_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId
from PG_Grass_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId
from PG_Grass_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId
from PG_Grass_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
```

```
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

941 View PG_GravMoist12_HR0

941.1 SQL Query for view PG_GravMoist12_HR0

```
select
 GS.SchoolId,
 GS.SiteId,
 GS.DayMeasured as Day,
 GS.Depth
from
 GLOBE.SMT_GravStarInt_HR1 GS,
 GLOBE.SITE_LOCATION SL
where
 GS.SchoolId = SL.SchoolId AND
 GS.SiteId = SL.SiteId AND
 SL.DataSource in (1,2)
UNION
select
 GH.SchoolId,
 GH.SiteId,
 GH.DayMeasured as Day,
 GH.Depth
from
 GLOBE.SMT_GravHole_HR1 GH,
 GLOBE.SITE_LOCATION SL
where
 GH.SchoolId = SL.SchoolId AND
 GH.SiteId = SL.SiteId AND
 SL.DataSource in (1,2)
```

942 View PG_GravMoist12_HR1

942.1 SQL Query for view PG_GravMoist12_HR1

```
select
 SchoolId,
 SiteId,
 Day,
 Count(Depth) as Levels
from
 PG_GravMoist12_HR0
group by SchoolId, SiteId, Day
```

943 View PG_GravMoist12_HR2

943.1 SQL Query for view PG_GravMoist12_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_GravMoist12_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 SUM(Levels) >= 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_GravMoist12_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
```

```
having
 SUM(Levels) >= 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_GravMoist12_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 SUM(Levels) >= 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_GravMoist12_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 SUM(Levels) >= 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_GravMoist12_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 SUM(Levels) >= 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_GravMoist12_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 SUM(Levels) >= 12 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

944 View PG_HYDRO_HR1

944.1 SQL Query for view PG_HYDRO_HR1

```
select HRnbr, SchoolId, 'WaterTemp' as Measurement
from PG_temp_HR2
UNION
select HRnbr, SchoolId, 'Acidity' as Measurement
from PG_acidity_HR2
UNION
select HRnbr, SchoolId, 'Turbidity' as Measurement
from PG_turbidity_HR2
UNION
select HRnbr, SchoolId, 'Nitrate' as Measurement
from PG_Nitrate_HR2
UNION
select HRnbr, SchoolId, 'Alkalinity' as Measurement
from PG_alkalinity_HR2
UNION
select HRnbr, SchoolId, 'DissolvedOxy' as Measurement
from PG_DissolvedOxy_HR2
UNION
select HRnbr, SchoolId, 'Salinity' as Measurement
from PG_salinity_HR2
UNION
select HRnbr, SchoolId, 'Conductivity' as Measurement
from PG_conductivity_HR2
```

945 View PG_HYDRO_HR2

945.1 SQL Query for view PG_HYDRO_HR2

```
select
 HRnbr,
 SchoolId,
 'Hydro' as Measurement
from
 PG_HYDRO_HR1
where HRnbr >= 7
group by HRnbr, SchoolId
having Count(*) >= 2
```

946 View PG_Infiltration_HR1

946.1 SQL Query for view PG_Infiltration_HR1

```
select
 I.SchoolId,
 I.SiteId,
 I.DayMeasured as Day,
 Count(SampleNbr) as Samples
from
 GLOBE.SOIL_INFILTRATION I,
 GLOBE.SITE_LOCATION SL
where
 I.SchoolId = SL.SchoolId AND
 I.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 I.InnerRingDiam is not null AND
 I.OuterRingDiam is not null AND
 I.SatWaterContent is not null
group by I.SchoolId, I.SiteId, I.DayMeasured
```

947 View PG_Infiltration_HR2

947.1 SQL Query for view PG_Infiltration_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId
from PG_Infiltration_HR1 A
```

```
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 Count(*) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId
from PG_Infiltration_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 Count(*) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId
from PG_Infiltration_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 Count(*) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId
from PG_Infiltration_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 Count(*) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId
from PG_Infiltration_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 Count(*) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
```

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId
from PG_Infiltration_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 Count(*) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

948 View PG_LAND_HR1

948.1 SQL Query for view PG_LAND_HR1

```
select HRnbr, SchoolId, 'Grass' as Measurement
from PG_GRASS_HR2
where HRnbr >= 7
UNION ALL
select HRnbr, SchoolId, 'Tree' as Measurement
from GLOBE.TREEBIOM_HR2
where HRnbr >= 7
UNION ALL
select HRnbr, SchoolId, 'Qual5' as Measurement
from (select HRnbr, SchoolId
      from GLOBE.LCL_HR2
      where HRnbr >= 7
      group by HRnbr, SchoolId
      having COUNT(SchoolId) >= 5 )
```

949 View PG_LAND_HR2

949.1 SQL Query for view PG_LAND_HR2

```
select
 HRnbr,
 SchoolId,
 'Land' as Measurement
from
 PG_LAND_HR1
group by HRnbr, SchoolId
```

950 View PG_NearSurf_HR1

950.1 SQL Query for view PG_NearSurf_HR1

```
select
 SC.SchoolId,
 SC.SiteId,
 Count(SC.HorizonNbr) as Levels,
 TRUNC(SC.UTtimeMeasured) as Day
from
 GLOBE.SOIL_CHAR SC,
 GLOBE.SITE_LOCATION SL
where
 SC.SchoolId = SL.SchoolId AND
 SC.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 SC.HorizonNbr in (1,2) AND
 SC.MainColorCode is not null AND
 SC.RockEstimate > 0 AND
 SC.RootEstimate > 0 AND
 SC.Carbonates > 0 AND
 (SC.TextureField >0 OR SC.TextureUSDA >0) AND
 SC.NitrogenEst > 0 AND
 SC.PhosphorusEst > 0 AND
 SC.PotassiumEst > 0 AND
```

```
 SC.BulkDensity is not null AND
 SC.Acidity is not null AND
 (SC.SandPct96 is not null OR
  SC.SandPctUSDA is not null OR
  SC.SandPctISSS is not null)
group by SC.SchoolId, SC.SiteId, TRUNC(SC.UTtimeMeasured)
```

951 View PG_NearSurf_HR2

951.1 SQL Query for view PG_NearSurf_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_NearSurf_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 SUM(Levels) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_NearSurf_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 SUM(Levels) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_NearSurf_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 SUM(Levels) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_NearSurf_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 SUM(Levels) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_NearSurf_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 SUM(Levels) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, SUM(Levels) as Levels
from PG_NearSurf_HR1 A
group by A.SchoolId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 SUM(Levels) >= 4 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

952 View PG_Nitrate_HR1

952.1 SQL Query for view PG_Nitrate_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(NitriteNitrate) AS NitrateCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR NitriteNitrate is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(NitriteNitrate) > 0
```

953 View PG_Precip_HR1

953.1 SQL Query for view PG_Precip_HR1

```
select
 P.SchoolId,
 P.SiteId,
 P.Month,
 SumRDays,
 SumSDays,
 SumREDays
from
 GLOBE.ATMOS_PRECIP_HR2 P,
 GLOBE.SITE_LOCATION SL
where
 P.SchoolId = SL.SchoolId AND
```

```
 P.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 P.Month >= TO_DATE('20000201','YYYYMMDD')
```

954 View PG_Precip_HR2

954.1 SQL Query for view PG_Precip_HR2

```
select
2*TRUNC(Round(MONTHS_BETWEEN(P.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 as HRnbr,
P.SchoolId,
P.SiteId,
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) as PrecipDays
from PG_Precip_HR1 P
group by P.SchoolId, P.SiteId,
2*TRUNC(Round(MONTHS_BETWEEN(P.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
MIN(P.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
AND
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) >= 4*30*(.70)
UNION ALL
select
2*TRUNC(Round((MONTHS_BETWEEN(P.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
P.SchoolId,
P.SiteId,
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) as PrecipDays
from PG_Precip_HR1 P
group by P.SchoolId, P.SiteId,
2*TRUNC(Round((MONTHS_BETWEEN(P.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
MIN(P.Month) > TO_DATE('1999 02 01','YYYY MM DD')
AND
SUM(GREATEST(SumRDays, SumSDays, SumREDays)) >= 4*30*(.70)
```

955 View PG_SOIL_HR1

955.1 SQL Query for view PG_SOIL_HR1

```
select HRnbr, SchoolId, 'SoilChar_1' as Measurement
from GLOBE.SOILCHAR_HR4
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'NearSurf_2' as Measurement
from PG_NearSurf_HR2
where HRnbr >= 7
UNION
select HRnbr, SchoolId, 'GravMoist_3' as Measurement
from GLOBE.SMT_MOISTURE_HR2
where
 LocMethod IN ('GPS','Other') AND
 HRnbr >= 7
UNION
select HRnbr, SchoolId, 'GravMoist_4' as Measurement
from PG_GravMoist12_HR2
where HRnbr >= 7
UNION
select B.HRnbr, B.SchoolId, 'BlockMoist_5' as Measurement
from
 GLOBE.SMT_BLOCK_HR2 B,
 GLOBE.SITE_LOCATION SL
where
 B.SchoolId = SL.SchoolId AND
 B.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 B.HRnbr >= 7
UNION
select ST.HRnbr, ST.SchoolId, 'SoilTemp_6' as Measurement
from
 GLOBE.SMT_DIURNAL_HR3  D,
 GLOBE.SMT_TEMP_HR2 ST
where
 D.HRnbr = ST.HRnbr AND
 D.HRnbr >= 7 AND
 D.SchoolId = ST.SchoolId AND
 D.LocMethod IN ('GPS','Other')
```

```
UNION
select HRnbr, SchoolId, 'Infiltration_7' as Measurement
from PG_Infiltration_HR2
where HRnbr >= 7
```

956 View PG_SOIL_HR2

956.1 SQL Query for view PG_SOIL_HR2

```
select
 HRnbr,
 SchoolId,
 'Soil' as Measurement
from
 PG_SOIL_HR1
where HRnbr >= 7
group by HRnbr, SchoolId
```

957 View PG_acidity_HR1

957.1 SQL Query for view PG_acidity_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(Acidity) AS AcidityCnt
from
SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR Acidity is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(Acidity) > 0
```

958 View PG_acidity_HR2

958.1 SQL Query for view PG_acidity_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_acidity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_acidity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

959 View PG_alkalinity_HR1

959.1 SQL Query for view PG_alkalinity_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(Alkalinity) AS AlkalCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR Alkalinity is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(Alkalinity) > 0
```

960 View PG_alkalinity_HR2

960.1 SQL Query for view PG_alkalinity_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_alkalinity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_alkalinity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

961 View PG_conductivity_HR1

961.1 SQL Query for view PG_conductivity_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(Conductivity) AS ConductivityCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR Conductivity is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(Conductivity) > 0
```

962 View PG_conductivity_HR2

962.1 SQL Query for view PG_conductivity_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_conductivity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_conductivity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

963 View PG_nitrate_HR2

963.1 SQL Query for view PG_nitrate_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_nitrate_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_nitrate_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

964 View PG_salinity_HR1

964.1 SQL Query for view PG_salinity_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(HydSalinity)+COUNT(TitrSalinity) AS SalinityCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR
```

```
  HydSalinity is not null OR
  TitrSalinity is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(HydSalinity)+COUNT(TitrSalinity) > 0
```

965 View PG_salinity_HR2

965.1 SQL Query for view PG_salinity_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_salinity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_salinity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

966 View PG_temp_HR1

966.1 SQL Query for view PG_temp_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(WaterTemp) AS TempCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR WaterTemp is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(WaterTemp) > 0
```

967 View PG_temp_HR2

967.1 SQL Query for view PG_temp_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_temp_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
```

```
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_temp_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

968 View PG_turbidity_HR1

968.1 SQL Query for view PG_turbidity_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'WW') as Week,
 DECODE(MIN(SourceState),0,Null,1) AS BadSource,
 COUNT(*) AS RowCnt,
 COUNT(TurbiditySecchi)+COUNT(TurbidityTube) AS TurbCnt
from
 SURFACE_WATER
where
 DayMeasured >= TO_DATE('20000201','YYYYMMDD') AND
 (SourceState = 0 OR
  TurbiditySecchi is not null OR
  TurbidityTube is not null)
group by Schoolid, SiteId, TRUNC(DayMeasured,'WW')
having
 DECODE(MIN(SourceState),0,Null,1)=1 OR
 COUNT(TurbiditySecchi)+COUNT(TurbidityTube) > 0
```

969 View PG_turbidity_HR2

969.1 SQL Query for view PG_turbidity_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_turbidity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))/4,1))+1
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId
from
 PG_turbidity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Week,TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
having
 MIN(T.Week) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND
 Count(*) >= 8
```

970 View PHENOLOGY

970.1 SQL Query for view PHENOLOGY

```
select
 WS.SchoolId,
 WS.SiteId,
 WS.Year,
 ROUND(MIN(WS.BudburstDay)+(MAX(WS.BudburstDay)-MIN(WS.BudburstDay))/2) as BudburstDay,
 ROUND(AVG(WS.BudburstDay-TRUNC(WS.BudburstDay,'YYYY'))) +1 as BBdayOfYear,
 MAX(BM.CommonName) as LocalName,
 BM.Genus4 as Genus,
 BM.Species4 as Species,
 MAX(TT.Genus) as GenusName,
 MAX(TT.Species) as SpeciesName
From GLOBE.PHENOLOGY_WS WS, GLOBE.BUDBURST_METADATA BM, GLOBE.TREE_TYPES TT
Where
 WS.SchoolId = BM.SchoolId AND
 WS.SiteId = BM.SiteId AND
 BM.Genus4 = TT.Genus4 (+) AND
 BM.Species4 = TT.Species4 (+)
Group By WS.SchoolId, WS.SiteId, WS.Year, BM.Genus4, BM.Species4
```

971 View PHENOLOGY_DR1

972 View PHENOLOGY_DR2

972.1 SQL Query for view PHENOLOGY_DR2

```
select PHENOLOGY_DR1.*, (PHENOLOGY_DR1.RowCnt+BB.HtCnt+BB.CmCnt) as DixonCnt
from PHENOLOGY_DR1, BUDBURST_META_DR1 BB
where PHENOLOGY_DR1.SchoolId = BB.SchoolId  AND  PHENOLOGY_DR1.SiteId  =  BB.SiteId  AND
PHENOLOGY_DR1.Month  =  BB.Month
```

973 View PHENOLOGY_DU1

974 View PHENOLOGY_DU2

974.1 SQL Query for view PHENOLOGY_DU2

```
select PHENOLOGY_DU1.*, (PHENOLOGY_DU1.RowCnt+BB.HtCnt+BB.CmCnt) as DixonCnt
from PHENOLOGY_DU1, BUDBURST_META_DU1 BB
where PHENOLOGY_DU1.SchoolId = BB.SchoolId  AND  PHENOLOGY_DU1.SiteId  =  BB.SiteId  AND
PHENOLOGY_DU1.Month  =  BB.Month
```

975 View PHENOLOGY_MC1

975.1 SQL Query for view PHENOLOGY_MC1

```
select 'PH' as MeasureCode, COUNT(*) as RowCnt, COUNT(BudburstDay) as BBAvgCnt, COUNT(GDS_30day) as
GDS30Cnt, COUNT(PET_30day) as PET30Cnt, COUNT(RainEqv_30day) as PR30Cnt, COUNT(Snowmelt_30day) as
SM30Cnt, COUNT(WaterDiff_30day) as WD30Cnt
from GLOBE.PHENOLOGY
```

976 View PHENOLOGY_MC2

976.1 SQL Query for view PHENOLOGY_MC2

```
select PHENOLOGY_MC1.*, ROWCNT as DixonCnt
from PHENOLOGY_MC1
```

977 View PHENOLOGY_RPRT_VU

977.1 SQL Query for view PHENOLOGY_RPRT_VU

```
select Distinct as SchoolID, 'PH' as MeasureCode, MIN(UTTimeReported) as FirstReport,
```

```
MAX(UTTimeReported) as LastReport
group by SchoolID
```

978 View PHENOLOGY_SITE_DR1

979 View PHENOLOGY_SITE_DR2

979.1 SQL Query for view PHENOLOGY_SITE_DR2

```
select PHENOLOGY_SITE_DR1.*, (ATLnkCnt+STypeCnt) as DixonCnt
from PHENOLOGY_SITE_DR1
```

980 View PHENOLOGY_SITE_DU1

981 View PHENOLOGY_SITE_DU2

981.1 SQL Query for view PHENOLOGY_SITE_DU2

```
select PHENOLOGY_SITE_DU1.*, (ATLnkCnt+STypeCnt) as DixonCnt
from PHENOLOGY_SITE_DU1
```

982 View PHENOLOGY_SITE_MC1

982.1 SQL Query for view PHENOLOGY_SITE_MC1

```
select 'MD' as MeasureCode, COUNT(AtmosLink) as ATLnkCnt, COUNT(SiteType) as STypeCnt
from GLOBE.PHENOLOGY_SITE
```

983 View PHENOLOGY_SITE_MC2

983.1 SQL Query for view PHENOLOGY_SITE_MC2

```
select PHENOLOGY_SITE_MC1.*, (ATLnkCnt+STypeCnt) as DixonCnt
from PHENOLOGY_SITE_MC1
```

984 View PHENOLOGY_SITE_RPRT_VU

984.1 SQL Query for view PHENOLOGY_SITE_RPRT_VU

```
select Distinct as SchoolID, 'MD' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from PHENOLOGY_SITE
group by SchoolID
```

985 View PHENOLOGY_SITE_VU1

985.1 SQL Query for view PHENOLOGY_SITE_VU1

```
select
 Schoolid,
 Siteid,
 TRUNC(VersionDate,'MM') AS Month,
 'MD' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(AtmosLink) AS ATLnkCnt,
 COUNT(SiteType) AS STypeCnt,
 MIN(VersionDate) AS BegDate,
 MAX(VersionDate) AS EndDate,
 MIN(UTtimestamp) AS BegRDate,
 MAX(UTtimestamp) AS EndRDate,
 ROUND(AVG(UTtimeStamp-VersionDate),2) AS LagTimeAvg
from
 GLOBE.PHENOLOGY_SITE
where
```

```
 UTtimeStamp < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 SchoolId, SiteId, TRUNC(VersionDate,'MM')
```

986 View PHENOLOGY_SITE_VU2

986.1 SQL Query for view PHENOLOGY_SITE_VU2

```
select PHENOLOGY_SITE_VU1.*, (ATLnkCnt+STypeCnt) as DixonCnt
from PHENOLOGY_SITE_VU1
```

987 View PHENOLOGY_VU1

987.1 SQL Query for view PHENOLOGY_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(BudBurstDay,'MM') AS MONTH,
 'PH' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(BudburstDay) as BBAvgCnt,
 MIN(BudBurstDay) AS BegDate,
 MAX(BudBurstDay) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-BudburstDay),2) AS LagTimeAvg
from
 GLOBE.PHENOLOGY
where
 UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 SchoolId, SiteId, TRUNC(BudBurstDay,'MM')
```

988 View PHENOLOGY_VU2

988.1 SQL Query for view PHENOLOGY_VU2

```
select PHENOLOGY_VU1.*, (PHENOLOGY_VU1.RowCnt+BB.HtCnt+BB.CmCnt) as DixonCnt
from PHENOLOGY_VU1, BUDBURST_META_VU1 BB
where PHENOLOGY_VU1.SchoolId = BB.SchoolId  AND  PHENOLOGY_VU1.SiteId  =  BB.SiteId  AND
PHENOLOGY_VU1.Month  =  BB.Month
```

989 View PHENOLOGY_WS_DR1

990 View PHENOLOGY_WS_DR2

990.1 SQL Query for view PHENOLOGY_WS_DR2

```
select PHENOLOGY_WS_DR1.*, (BBCnt+POCnt) as DixonCnt
from PHENOLOGY_WS_DR1
```

991 View PHENOLOGY_WS_DU1

992 View PHENOLOGY_WS_DU2

992.1 SQL Query for view PHENOLOGY_WS_DU2

```
select PHENOLOGY_WS_DU1.*, (BBCnt+POCnt) as DixonCnt
from PHENOLOGY_WS_DU1
```

993 View PHENOLOGY_WS_MC1

993.1 SQL Query for view PHENOLOGY_WS_MC1

```
select 'PH' as MeasureCode, COUNT(BudburstDay) as BBCnt, COUNT(PrevObsDay) as POCnt,
SUM(DECODE(BranchId,'1',1,0)*DECODE(TreeHeight,NULL,0,1)) as HtCnt,
SUM(DECODE(BranchId,'1',1,0)*DECODE(TreeCircum,NULL,0,1)) as CmCnt
from GLOBE.PHENOLOGY_WS
```

994 View PHENOLOGY_WS_MC2

994.1 SQL Query for view PHENOLOGY_WS_MC2

```
select PHENOLOGY_WS_MC1.*, (BBCnt+POCnt+HtCnt+CmCnt) as DixonCnt
from PHENOLOGY_WS_MC1
```

995 View PHENOLOGY_WS_RPRT_VU

995.1 SQL Query for view PHENOLOGY_WS_RPRT_VU

```
select Distinct as SchoolID, 'PH' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from PHENOLOGY_WS
group by SchoolID
```

996 View PHENOLOGY_WS_VU1

996.1 SQL Query for view PHENOLOGY_WS_VU1

```
select
 Schoolid,
 Siteid,
 TRUNC(BudBurstDay,'MM') AS MONTH,
 'PH' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(BudburstDay) as BBCnt,
 COUNT(PrevObsDay) AS POCnt,
 MIN(BudBurstDay) AS BegDate,
 MAX(BudBurstDay) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-BudburstDay),2) AS LagTimeAvg
from
 GLOBE.PHENOLOGY_WS
where
 UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by
 SchoolId, SiteId, TRUNC(BudBurstDay,'MM')
```

997 View PHENOLOGY_WS_VU2

997.1 SQL Query for view PHENOLOGY_WS_VU2

```
select PHENOLOGY_WS_VU1.*, (BBCnt+POCnt) as DixonCnt
from PHENOLOGY_WS_VU1
```

998 View PHN_JoinFour_VU

998.1 SQL Query for view PHN_JoinFour_VU

```
select PHN_JoinThree_VU.*, PET_CHART.PETVALUE, NewRain+NVL(RainEquiv,0) -PETvalue as WaterDiff
from PHN_JoinThree_VU, GLOBE.PET_CHART
where Round  (
PHN_Jo in Three_VU.GDSvalue
,
1 )  = PET_CHART.AVGTEMP
```

999 View PHN_JoinThree_VU

999.1 SQL Query for view PHN_JoinThree_VU

```
select
 AIR_TEMP.SCHOOLID as SchoolId,
 AIR_TEMP.SITEID as SiteId,
 AIR_TEMP.SOLARDAY as SolarDay,
 Round((MaxTemp+MinTemp)/2,1) as AvgTemp,
 Greatest(0,Round((MaxTemp+MinTemp)/2,1)) as GDSvalue,
 PRECIP_RAIN.RAINAMOUNT as NewRain,
 PRECIP_RAIN.DAILYTMAFLAG as F1,
 PRECIP_SOLID.NEWAMOUNT as NewSnow,
 PRECIP_SOLID.DAILYTMAFLAG as F2,
 NVL(RainEquivalent,Decode(NewAmount,0,0,null)) as RainEquiv,
 PRECIP_SOLID.RAINEQUIVTMAFLAG as F3,
 PRECIP_SOLID.CURRENTDEPTH as Snowpack,
 PRECIP_SOLID.TOTALTMAFLAG as F4,
 NVL(WaterEquiv,Decode(CurrentDepth,0,0,1,0.1,null,null,Round(CurrentDepth/7,1))) as WaterEquiv,
 NVL(WaterEquivTMAflag,Decode(CurrentDepth,null,TotalTMAflag,0,TotalTMAflag,1,TotalTMAflag,'E')) as F5
from
 GLOBE.AIR_TEMP,
 GLOBE.PRECIP_RAIN,
 GLOBE.PRECIP_SOLID
where
 AIR_TEMP.SolarDay>=TRUNC(SYSDATE,'YEAR') AND
 AIR_TEMP.SOLARDAY = PRECIP_RAIN.SOLARDAY AND
 AIR_TEMP.SITEID = PRECIP_RAIN.SITEID AND
 AIR_TEMP.SCHOOLID = PRECIP_RAIN.SCHOOLID AND
 PRECIP_RAIN.SOLARDAY = PRECIP_SOLID.SOLARDAY AND
 PRECIP_RAIN.SITEID = PRECIP_SOLID.SITEID AND
 PRECIP_RAIN.SCHOOLID = PRECIP_SOLID.SCHOOLID AND
 ((TRUNC(AIR_TEMP.SolarDay+182,'YEAR')=TRUNC(SYSDATE,'YEAR') AND
   TRUNC(SYSDATE+182,'YEAR')=TRUNC(SYSDATE,'YEAR')) OR
  (TRUNC(AIR_TEMP.SolarDay-181,'YEAR')=TRUNC(SYSDATE,'YEAR') AND
   TRUNC(SYSDATE-181,'YEAR')=TRUNC(SYSDATE,'YEAR'))) AND
 AIR_TEMP.UTTIMEREPORTED>=TRUNC(SYSDATE,'YEAR') AND
 PRECIP_RAIN.UTTIMEREPORTED>=TRUNC(SYSDATE,'YEAR') AND
 PRECIP_SOLID.UTTIMEREPORTED>=TRUNC(SYSDATE,'YEAR')
```

1000 View PH_SOIL_HR1

1000.1 SQL Query for view PH_SOIL_HR1

```
select distinct
 SC.SchoolId,
 SC.SiteId,
 SC.HorizonNbr,
 TRUNC(SC.UTtimeMeasured) as HorizDay
from
 GLOBE.SOIL_CHAR SC,
 GLOBE.SITE_LOCATION SL
where
 SC.SchoolId = SL.SchoolId AND
 SC.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 SC.UTtimeMeasured >= TO_DATE('19990201','YYYYMMDD') AND
 SC.Acidity is not null
```

1001 View PH_SOIL_HR2

1001.1 SQL Query for view PH_SOIL_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId, A.HorizonNbr
from PH_SOIL_HR1 A
group by A.SchoolId, A.SiteId, A.HorizonNbr,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId, A.HorizonNbr
from PH_SOIL_HR1 A
group by A.SchoolId, A.SiteId, A.HorizonNbr,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId, A.HorizonNbr
from PH_SOIL_HR1 A
group by A.SchoolId, A.SiteId, A.HorizonNbr,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId, A.HorizonNbr
from PH_SOIL_HR1 A
group by A.SchoolId, A.SiteId, A.HorizonNbr,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId, A.HorizonNbr
from PH_SOIL_HR1 A
group by A.SchoolId, A.SiteId, A.HorizonNbr,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId, A.HorizonNbr
from PH_SOIL_HR1 A
group by A.SchoolId, A.SiteId, A.HorizonNbr,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(HorizDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1002 View PH_SOIL_HR3

1002.1 SQL Query for view PH_SOIL_HR3

```
select
 HRnbr,
 SchoolId
from PH_SOIL_HR2
group by HRnbr, SchoolId
having Count(*) >= 4
```

1003 View PH_SURFWATER_HR1

1003.1 SQL Query for view PH_SURFWATER_HR1

```
select
 SchoolId,
 SiteId,
 Month,
 WpHCnt as pHDays
from GLOBE.SURFACE_WATER_CNT
where
 WphCnt > 0 AND
 Month >= TO_DATE('19990201','YYYYMMDD')  AND
 Month  <  TRUNC(SYSDATE,'MM')
```

1004 View PH_SURFWATER_HR2

1004.1 SQL Query for view PH_SURFWATER_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.pHDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 PH_SURFWATER_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.pHDays) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.pHDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 PH_SURFWATER_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.pHDays) >= 8
```

1005 View PH_SURFWATER_HR3

1005.1 SQL Query for view PH_SURFWATER_HR3

```
select distinct
 SW.HRnbr,
 SW.SchoolId
from
 PH_SURFWATER_HR2 SW,
 GLOBE.SITE_LOCATION SL
where
 SW.SchoolId = SL.SchoolId AND
 SW.SiteId = SL.SiteId AND
 SL.DataSource IN (1,2)
```

1006 View POSTCHECKLIST_VIEW

1006.1 SQL Query for view POSTCHECKLIST_VIEW

```
select WORKSHOP_SESSION.WORKSHOPNAME, WORKSHOP_SESSION.LOCATIONCITY, WORKSHOP_SESSION.BUDGET,
WORKSHOP_SESSION.ENDDATE, POST_CHECKLIST.SESSIONID, POST_CHECKLIST.POSTACTION, POST_CHECKLIST.DUEDATE,
POST_CHECKLIST.RECEIVEDDATE, POST_CHECKLIST.COMMENTS
from WORKSHOP_SESSION, POST_CHECKLIST
where WORKSHOP_SESSION.SESSIONID = POST_CHECKLIST.SESSIONID and WORKSHOP_SESSION.BEGINDATE < =
CURRENT_DATE and WORKSHOP_SESSION.ENDDATE > =  CURRENT_DATE
order by WORKSHOP_SESSION.ENDDATE ASC
```

1007 View PRECHECKLIST_VIEW

1007.1 SQL Query for view PRECHECKLIST_VIEW

```
select WORKSHOP_SESSION.WORKSHOPNAME, WORKSHOP_SESSION.LOCATIONCITY, WORKSHOP_SESSION.BUDGET,
WORKSHOP_SESSION.BEGINDATE, PRE_CHECKLIST.SESSIONID, PRE_CHECKLIST.PREACTION, PRE_CHECKLIST.DUEDATE,
PRE_CHECKLIST.RECEIVEDDATE, PRE_CHECKLIST.COMMENTS
from WORKSHOP_SESSION, PRE_CHECKLIST
where WORKSHOP_SESSION.SESSIONID = PRE_CHECKLIST.SESSIONID and WORKSHOP_SESSION.BEGINDATE > =
CURRENT_DATE
order by WORKSHOP_SESSION.BEGINDATE ASC
```

1008 View PRECIP_RAIN_DR1

1008.1 SQL Query for view PRECIP_RAIN_DR1

```
select TRUNC(UTTimeReported,'DD') as Day, 'PR' as MeasureCode, COUNT(*) as RowCnt, COUNT(RainAmount)
as RainCnt, COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) as DailyRMCnt, COUNT(Acidity) AS RpHCntFROM
GLOBE.PRECIP_RAINWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTTimeReported,'DD')
```

1009 View PRECIP_RAIN_DR2

1009.1 SQL Query for view PRECIP_RAIN_DR2

```
select PRECIP_RAIN_DR1.*, (raincnt+dailyrmcnt+rpHcnt) as DixonCnt
from PRECIP_RAIN_DR1
```

1010 View PRECIP_RAIN_DU1

1010.1 SQL Query for view PRECIP_RAIN_DU1

```
select TRUNC(UTTimeMeasured,'DD') as Day, 'PR' as MeasureCode, COUNT(*) as RowCnt, COUNT(RainAmount)
as RainCnt, COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) as DailyRMCnt, COUNT(Acidity) AS RpHCntFROM
GLOBE.PRECIP_RAINWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTTimeMeasured,'DD')
```

1011 View PRECIP_RAIN_DU2

1011.1 SQL Query for view PRECIP_RAIN_DU2

```
select PRECIP_RAIN_DU1.*, (raincnt+dailyrmcnt+rpHcnt) as DixonCnt
from PRECIP_RAIN_DU1
```

1012 View PRECIP_RAIN_MC1

1012.1 SQL Query for view PRECIP_RAIN_MC1

```
select 'PR' as MeasureCode, COUNT(*) as RowCnt, COUNT(RainAmount) as RainCnt,
COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) as DailyRMCnt, COUNT(Acidity) AS RpHCntFROM as
GLOBE.PRECIP_RAIN
```

1013 View PRECIP_RAIN_MC2

1013.1 SQL Query for view PRECIP_RAIN_MC2

```
select PRECIP_RAIN_MC1.*, (raincnt+dailyrmcnt+3*rpHcnt) as DixonCnt
from PRECIP_RAIN_MC1
```

1014 View PRECIP_RAIN_RPRT_VU

1014.1 SQL Query for view PRECIP_RAIN_RPRT_VU

```
select Distinct as SchoolID, 'PR' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from PRECIP_RAIN
group by SchoolID
```

1015 View PRECIP_RAIN_VU1

1015.1 SQL Query for view PRECIP_RAIN_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay,'MM') AS Month,
 'PR' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(RainAmount) AS RainCnt,
 MIN(RainAmount) as RainMin,
 MAX(RainAmount) as RainMax,
 ROUND(AVG(RainAmount),1) as RainAvg,
 COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) AS DailyRMCnt,
 COUNT(Acidity) AS RpHCnt, SUM(RainAmount) AS RainSum,
 SUM(DaysAccumulated*DECODE(NVL(RainAmount,-9),-9,0,1)) AS SumRDays,
 ROUND(SUM(Acidity*Rainamount)     /(DECODE(SUM(RainAmount*DECODE(NVL(Acidity,-9),-9,0,1)),0,NULL,1)
*SUM(Rainamount*DECODE(NVL(Acidity,-9),-9,0,1))),1) AS RpHWAvg,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
 from GLOBE.PRECIP_RAIN
 where UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
 GROUP BY SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

1016 View PRECIP_RAIN_VU2

1016.1 SQL Query for view PRECIP_RAIN_VU2

```
select PRECIP_RAIN_VU1.*, (raincnt+dailyrmcnt+3*rpHcnt) as DixonCnt
from PRECIP_RAIN_VU1
```

1017 View PRECIP_SOLID_DR1

1017.1 SQL Query for view PRECIP_SOLID_DR1

```
select TRUNC(UTTimeReported,'DD') as Day, 'PS' as MeasureCode, COUNT(*) as RowCnt, COUNT(NewAmount) as
NewCnt, COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) as DailyMCnt,
COUNT(DECODE(RainEquivalent,0.0,NULL,0.1,NULL,1.0)) as REqvCntD, COUNT(CurrentDepth) as CDepCnt,
COUNT(DECODE(TotalTMAFlag,'M','M',NULL)) as TotMCnt, COUNT(Acidity) as SpHCnt, COUNT(RainEquivalent)
AS REqvCntFROM GLOBE.PRECIP_SOLIDWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTTimeReported,'DD')
```

1018 View PRECIP_SOLID_DR2

1018.1 SQL Query for view PRECIP_SOLID_DR2

```
select PRECIP_SOLID_DR1.*, (cdepcnt+totmcnt+newcnt+dailymcnt+reqvcntd+spHcnt) as DixonCnt
from PRECIP_SOLID_DR1
```

1019 View PRECIP_SOLID_DU1

1019.1 SQL Query for view PRECIP_SOLID_DU1

```
select TRUNC(UTTimeMeasured,'DD') as Day, 'PS' as MeasureCode, COUNT(*) as RowCnt, COUNT(NewAmount) as
NewCnt, COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) as DailyMCnt,
COUNT(DECODE(RainEquivalent,0.0,NULL,0.1,NULL,1.0)) as REqvCntD, COUNT(CurrentDepth) as CDepCnt,
COUNT(DECODE(TotalTMAFlag,'M','M',NULL)) as TotMCnt, COUNT(Acidity) as SpHCnt, COUNT(RainEquivalent)
AS REqvCntFROM GLOBE.PRECIP_SOLIDWHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTTimeMeasured,'DD')
```

1020 View PRECIP_SOLID_DU2

1020.1 SQL Query for view PRECIP_SOLID_DU2

```
select PRECIP_SOLID_DU1.*, (cdepcnt+totmcnt+newcnt+dailymcnt+reqvcntd+spHcnt) as DixonCnt
from PRECIP_SOLID_DU1
```

1021 View PRECIP_SOLID_MC1

1021.1 SQL Query for view PRECIP_SOLID_MC1

```
select 'PS' as MeasureCode, COUNT(*) as RowCnt, COUNT(NewAmount) as NewCnt,
COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) as DailyMCnt,
COUNT(DECODE(RainEquivalent,0.0,NULL,0.1,NULL,Null,Null,1.0)) as REqvCntD, COUNT(CurrentDepth) as
CDepCnt, COUNT(DECODE(TotalTMAFlag,'M','M',NULL)) as TotMCnt, COUNT(Acidity) as SpHCnt,
COUNT(DECODE(WaterEquiv,0.0,Null,0.1,Null,Null,Null,1.0)) as WEqvCntD, COUNT(RainEquivalent) AS
REqvCntFROM as GLOBE.PRECIP_SOLID
```

1022 View PRECIP_SOLID_MC2

1022.1 SQL Query for view PRECIP_SOLID_MC2

```
select PRECIP_SOLID_MC1.*, (cdepcnt+totmcnt+newcnt+dailymcnt+reqvcntd+3*spHcnt+weqvcntd) as DixonCnt
from PRECIP_SOLID_MC1
```

1023 View PRECIP_SOLID_RPRT_VU

1023.1 SQL Query for view PRECIP_SOLID_RPRT_VU

```
select Distinct as SchoolID, 'PS' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from PRECIP_SOLID
group by SchoolID
```

1024 View PRECIP_SOLID_VU1

1024.1 SQL Query for view PRECIP_SOLID_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay,'MM') AS Month,
 'PS' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(NewAmount) AS NewCnt,
 COUNT(DECODE(DailyTMAFlag,'M','M',NULL)) AS DailyMCnt,
 COUNT(DECODE(RainEquivalent,0.0,Null,0.1,Null,Null,Null,1.0)) AS REqvCntD,
 COUNT(CurrentDepth) AS CDepCnt,
 COUNT(DECODE(TotalTMAFlag,'M','M',NULL)) AS TotMCnt,
 COUNT(Acidity) AS SpHCnt,
 COUNT(AciditySnowpack) AS SPpHCnt,
 SUM(NewAmount) AS NewSum,
 MIN(NewAmount) as NewMin,
 MAX(NewAmount) as NewMax,
 ROUND(AVG(NewAmount),1) as NewAvg,
 SUM(RainEquivalent) AS REqvSum,
 COUNT(RainEquivalent) AS REqvCnt,
 MIN(RainEquivalent) as REqvMin,
 MAX(RainEquivalent) as REqvMax,
 ROUND(AVG(RainEquivalent),1) as REqvAvg,
 SUM(DaysAccumulated*DECODE(NVL(NewAmount,-9),-9,0,1)) AS SumSDays,
 SUM(DaysAccumulated*DECODE(NVL(RainEquivalent,-9),-9,0,1)) AS SumREDays,
 COUNT(DECODE(WaterEquiv,0.0,Null,0.1,Null,Null,Null,1.0)) as WEqvCntD,
 ROUND(SUM(Acidity*RainEquivalent)/
   (DECODE(SUM(RainEquivalent*DECODE(NVL(Acidity,-9),-9,0,1)),0,NULL,1)*
   SUM(RainEquivalent*DECODE(NVL(Acidity,-9),-9,0,1))),1) AS SpHWAvg,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
 FROM GLOBE.PRECIP_SOLID
 WHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
 GROUP BY SchoolId, SiteId, TRUNC(SolarDay,'MM')
```

1025 View PRECIP_SOLID_VU2

1025.1 SQL Query for view PRECIP_SOLID_VU2

```
select PRECIP_SOLID_VU1.*,
(cdepcnt+totmcnt+newcnt+dailymcnt+reqvcntd+3*SpHcnt+SPpHcnt+weqvcntd) AS DixonCnt
from PRECIP_SOLID_VU1
```

1026 View PROTOCOL_HIERARCHY

1026.1 SQL Query for view PROTOCOL_HIERARCHY

```
select
 InvstgnCode as Invstgn,
 MeasureName as Protocol,
 LevelType as LevelType,
 MeasureCode as MeasCode
from MEASURE_CODES
where
 IsCurrent = 'Y' OR IsCurrent is Null
```

1027 View Phenology_HR1

1027.1 SQL Query for view Phenology_HR1

```
select SchoolId, SiteId, BeginDay, EndDay
from Lilac_HR1
UNION
```

```
select SchoolId, SiteId, BeginDay, EndDay
from GreenUp_HR1
UNION
select SchoolId, SiteId, BeginDay, EndDay
from GreenDown_HR1
```

1028 View Phenology_HR2

1028.1 SQL Query for view Phenology_HR2

```
select
 BB.SchoolId,
 BB.SiteId,
 MAX(PS.VersionDate) as VersionDate
from
 Phenology_HR1 BB,
 GLOBE.PHENOLOGY_SITE PS
where
 BB.SchoolId = PS.SchoolId AND
 BB.SiteId = PS.SiteId AND
 BB.EndDay >= PS.VersionDate
group by BB.SchoolId, BB.SiteId
```

1029 View Phenology_HR3

1029.1 SQL Query for view Phenology_HR3

```
select
 BB.SchoolId,
 BB.SiteId,
 BB.BeginDay,
 BB.EndDay,
 PS.AtmosLink
from
 Phenology_HR1 BB,
 GLOBE.PHENOLOGY_SITE PS,
 Phenology_HR2 PSL
where
 BB.SchoolId = PS.SchoolId(+) AND
 BB.SiteId = PS.SiteId(+) AND
 PS.SchoolId = PSL.SchoolId AND
 PS.SiteId = PSL.SiteId AND
 PS.VersionDate = PSL.VersionDate
```

1030 View Phenology_HR4

1030.1 SQL Query for view Phenology_HR4

```
select
 P.SchoolId,
 P.SiteId,
 P.AtmosLink,
 P.BeginDay,
 P.EndDay
from
 Phenology_HR3 P,
 GLOBE.AIR_TEMP T
where
 P.SchoolId = T.SchoolId AND
 P.AtmosLink = T.SiteId AND
 T.SolarDay >= P.BeginDay AND
 T.SolarDay <= P.EndDay
group by P.SchoolId, P.SiteId, P.AtmosLink, P.BeginDay, P.EndDay
having count(*) >= 0.70*(P.EndDay-P.BeginDay)
```

1031 View Phenology_HR5

1031.1 SQL Query for view Phenology_HR5

```
select
 P.SchoolId,
```

```
 P.SiteId,
 P.AtmosLink,
 P.BeginDay,
 P.EndDay,
 Sum(R.DaysAccumulated) as RainDays
from
 Phenology_HR4 P,
 GLOBE.PRECIP_RAIN R
where
 P.SchoolId = R.SchoolId AND
 P.AtmosLink = R.SiteId AND
 R.SolarDay >= P.BeginDay - 30 AND
 R.SolarDay <= P.EndDay + 7
group by P.SchoolId, P.SiteId, P.AtmosLink, P.BeginDay, P.EndDay
```

1032 View Phenology_HR6

1032.1 SQL Query for view Phenology_HR6

```
select
 P.SchoolId,
 P.SiteId,
 P.AtmosLink,
 P.BeginDay,
 P.EndDay,
 P.RainDays,
 NVL(Sum(S.DaysAccumulated),-1) as SnowDays
from
 Phenology_HR5 P,
 GLOBE.PRECIP_SOLID S
where
 P.SchoolId = S.SchoolId(+) AND
 P.AtmosLink = S.SiteId(+) AND
 S.SolarDay(+) >= P.BeginDay - 30 AND
 S.SolarDay(+) <= P.EndDay + 7
group by P.SchoolId, P.SiteId, P.AtmosLink, P.BeginDay, P.EndDay, P.RainDays
```

1033 View Phenology_HR7

1033.1 SQL Query for view Phenology_HR7

```
select distinct
 SchoolId,
 SiteId,
 EndDay
from
 Phenology_HR6
where
 Greatest(RainDays,SnowDays) >= (0.95)*(EndDay-BeginDay+30)
```

1034 View Phenology_HR8

1034.1 SQL Query for view Phenology_HR8

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId
from Phenology_HR7 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
```

```
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId
from Phenology_HR7 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId
from Phenology_HR7 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId
from Phenology_HR7 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId
from Phenology_HR7 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId
from Phenology_HR7 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(EndDay,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1035 View REACH250_EarlyDate

1035.1 SQL Query for view REACH250_EarlyDate

```
select X.SchoolId, Min(X.Month) as Early250CntMonth
from
(select
 S.SchoolId,
 M.Month,
 Sum(P.DixonCnt) as UpperCumulativeCnt
from
(select SchoolId
 from GLOBE.DIXON_SUMMARY_CNT
 where DixonCnt >= 250) S,
(select distinct SchoolId, Trunc(BegRDate,'MM') as Month
 from GLOBE.DIXON_SUMMARY_GROUPED) M,
GLOBE.DIXON_SUMMARY_GROUPED P
where
 S.SchoolId = M.SchoolId AND
 S.SchoolId = P.SchoolId AND
 P.BegRDate < M.Month + 32
group by S.SchoolId, M.Month
having
 Sum(P.DixonCnt) > 0 AND
 Sum(P.DixonCnt) < 5000) X
where
 X.UpperCumulativeCnt >= 250
group by X.SchoolId
```

1036 View REACH250_LateDate

1036.1 SQL Query for view REACH250_LateDate

```
select X.SchoolId, Min(X.Month) as Late250CntMonth
from
(select
 S.SchoolId,
 M.Month,
 Sum(P.DixonCnt) as LowerCumulativeCnt
from
(select SchoolId
 from GLOBE.DIXON_SUMMARY_CNT
 where DixonCnt > 250) S,
(select distinct SchoolId, Trunc(EndRDate,'MM') as Month
 from GLOBE.DIXON_SUMMARY_GROUPED) M,
GLOBE.DIXON_SUMMARY_GROUPED P
where
 S.SchoolId = M.SchoolId AND
 S.SchoolId = P.SchoolId AND
 P.EndRDate < M.Month + 32
group by S.SchoolId, M.Month
having
 Sum(P.DixonCnt) > 0 AND
 Sum(P.DixonCnt) < 5000) X
where
 X.LowerCumulativeCnt >= 250
group by X.SchoolId
```

1037 View SCHOOL_OBS_RATE

1037.1 SQL Query for view SCHOOL_OBS_RATE

```
select
 SchoolId,
 DixonCnt as TotalObs,
 ROUND(DixonCnt/MONTHS_BETWEEN(EndDate, BegDate)) as MonthlyObsRate,
 TRUNC(BegDate) as FirstObs,
 TRUNC(EndDate) as LastObs,
 ROUND(MONTHS_BETWEEN(EndDate, BegDate)) as NbrObsMonths
from
 GLOBE.DIXON_SUMMARY_CNT
where
 MONTHS_BETWEEN(EndDate, BegDate) > 0.5
```

1038 View SCHOOL_REPORT_RATE

1038.1 SQL Query for view SCHOOL_REPORT_RATE

```
select
  SchoolId,
  DixonCnt as TotalObs,
  ROUND(DixonCnt/MONTHS_BETWEEN(EndRDate, BegRDate)) as MonthlyRate,
  TRUNC(BegRDate) as FirstReport,
  TRUNC(EndRDate) as LastReport,
  ROUND(MONTHS_BETWEEN(EndRDate, BegRDate)) as NbrMonths
from
  GLOBE.DIXON_SUMMARY_CNT
where
  MONTHS_BETWEEN(EndRDate, BegRDate) >= 1
```

1039 View SITE_LOCATION_DR1

1039.1 SQL Query for view SITE_LOCATION_DR1

```
select TRUNC(UTtimestamp,'DD') as Day, 'GP' as MeasureCode, COUNT(DECODE(DataSource,1,1,NULL)) AS
GPSCntFROM GLOBE.SITE_LOCATIONWHERE UTtimestamp < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTtimestamp,'DD')
```

1040 View SITE_LOCATION_DR2

1040.1 SQL Query for view SITE_LOCATION_DR2

```
select SITE_LOCATION_DR1.*, 3*GPSCnt as DixonCnt
from SITE_LOCATION_DR1
```

1041 View SITE_LOCATION_DU2

1041.1 SQL Query for view SITE_LOCATION_DU2

```
select *
from SITE_LOCATION_DR2
```

1042 View SITE_LOCATION_MC1

1042.1 SQL Query for view SITE_LOCATION_MC1

```
select 'GP' as MeasureCode, COUNT(DECODE(DataSource,1,1,NULL)) AS GPSCntFROM as GLOBE.SITE_LOCATION
```

1043 View SITE_LOCATION_MC2

1043.1 SQL Query for view SITE_LOCATION_MC2

```
select SITE_LOCATION_MC1.*, 3*GPSCnt as DixonCnt
from SITE_LOCATION_MC1
```

1044 View SITE_LOCATION_RPRT_VU

1044.1 SQL Query for view SITE_LOCATION_RPRT_VU

```
select Distinct as SchoolID, 'GP' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from SITE_LOCATION
group by SchoolID
```

1045 View SITE_LOCATION_VU1

1045.1 SQL Query for view SITE_LOCATION_VU1

```
SELECT
  A.SchoolId,
  SiteId,
  TRUNC(UTTimestamp,'MM') AS Month,
  'GP' AS MeasureCode,
  COUNT(*) AS RowCnt,
  COUNT(DECODE(DataSource,1,1,NULL)) AS GPSCnt,
  MIN(UTTimestamp) AS BegDate,
  MAX(UTTimestamp) AS EndDate,
  MIN(UTTimestamp) AS BegRDate,
  MAX(UTTimestamp) AS EndRDate
FROM
  SITE_LOCATION A, GLOBE_SCHOOL B
WHERE
  A.SchoolId = B.SchoolId AND B.LifeCycleCode != 'M' AND
  UTTimestamp < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  A.SchoolId, SiteId, TRUNC(UTTimestamp,'MM')
```

1046 View SITE_LOCATION_VU2

1046.1 SQL Query for view SITE_LOCATION_VU2

```
select SITE_LOCATION_VU1.*, 3*GPSCnt as DixonCnt
from SITE_LOCATION_VU1
```

1047 View SITE_PHOTO_DR1

1048 View SITE_PHOTO_DR2

1048.1 SQL Query for view SITE_PHOTO_DR2

```
select *
from SITE_PHOTO_DR1
```

1049 View SITE_PHOTO_DU1

1049.1 SQL Query for view SITE_PHOTO_DU1

```
select TRUNC(UTtimeMeasured,'DD') as Day, 'SP' as MeasureCode, COUNT(DECODE(SiteCode,'LCL',1,Null)) as
QualSPCnt, COUNT(DECODE(SiteCode,'LCN',1,Null)) as QuanSPCnt, COUNT(DECODE(SiteCode,'BIO',1,Null)) as
BISPCnt, COUNT(DECODE(SiteCode,'ATM',1,Null)) as ATSPCnt, COUNT(DECODE(SiteCode,'SMS',1,Null)) as
SMSPCnt, COUNT(DECODE(SiteCode,'SCS',1,Null)) as SCSPCnt, COUNT(DECODE(SiteCode,'SWS',1,Null)) as
SWSPCnt, COUNT(DECODE(Sitecode,'PHE',1,NULL)) as PHSPCnt, COUNT(DECODE(SiteCode,'SCH',1,Null)) AS
SCHSPCntFROMGLOBE.SITE_PHOTO as SP, GLOBE.STUDY_SITE SSWHERESP.schoolid=SS.schoolid and
SP.siteid=SS.siteidGROUP BY TRUNC as (UTtimeMeasured,'DD')
```

1050 View SITE_PHOTO_DU2

1050.1 SQL Query for view SITE_PHOTO_DU2

```
select *
from SITE_PHOTO_DU1
```

1051 View SITE_PHOTO_MC1

1051.1 SQL Query for view SITE_PHOTO_MC1

```
select 'SP' as MeasureCode, COUNT(DECODE(SiteCode,'LCL',1,Null)) as QualSPCnt,
COUNT(DECODE(SiteCode,'LCN',1,Null)) as QuanSPCnt, COUNT(DECODE(SiteCode,'BIO',1,Null)) as BISPCnt,
COUNT(DECODE(SiteCode,'ATM',1,Null)) as ATSPCnt, COUNT(DECODE(SiteCode,'SMS',1,Null)) as SMSPCnt,
COUNT(DECODE(SiteCode,'SCS',1,Null)) as SCSPCnt, COUNT(DECODE(SiteCode,'SWS',1,Null)) as SWSPCnt,
```

```
COUNT(DECODE(Sitecode,'PHE',1,NULL)) as PHSPCnt, COUNT(DECODE(SiteCode,'SCH',1,Null)) AS
SCHSPCntFROMGLOBE.SITE_PHOTO as SP, GLOBE.STUDY_SITE SSWHERESP.schoolid=SS.schoolid and as
SP.siteid=SS.siteid
```

1052 View SITE_PHOTO_MC2

1052.1 SQL Query for view SITE_PHOTO_MC2

```
select *
from SITE_PHOTO_MC1
```

1053 View SITE_PHOTO_VU1

1053.1 SQL Query for view SITE_PHOTO_VU1

```
select
  SP.SchoolId,
  SP.SiteId,
  TRUNC(UTTimeMeasured,'MM') AS Month,
  'SP' AS MeasureCode,
  COUNT(*) AS RowCnt,
  COUNT(DECODE(SiteCode,'LCL',1,NULL)) AS QualSPCnt,
  COUNT(DECODE(SiteCode,'LCN',1,NULL)) AS QuanSPCnt,
  COUNT(DECODE(SiteCode,'BIO',1,NULL)) AS BISPCnt,
  COUNT(DECODE(SiteCode,'ATM',1,NULL)) AS ATSPCnt,
  COUNT(DECODE(SiteCode,'SMS',1,NULL)) AS SMSPCnt,
  COUNT(DECODE(SiteCode,'SCS',1,NULL)) AS SCSPCnt,
  COUNT(DECODE(SiteCode,'SWS',1,NULL)) AS SWSPCnt,
  COUNT(DECODE(Sitecode,'PHE',1,NULL)) AS PHSPCnt,
  COUNT(DECODE(SiteCode,'SCH',1,NULL)) AS SCHSPCnt,
  MIN(UTTimeMeasured) AS BegDate,
  MAX(UTTimeMeasured) AS EndDate,
  MIN(DateReceived) AS BegRDate,
  MAX(DateReceived) AS EndRDate,
  ROUND(AVG(ArchiveDate-UTTimeMeasured),2) AS LagTimeAvg
from
  SITE_PHOTO SP, PHOTO_ENVELOPE PE, STUDY_SITE SS
where
  SP.SchoolId = SS.SchoolId AND SP.SiteId=SS.SiteId  AND
  SP.SchoolId = PE.SchoolId AND SP.EnvelopeId = PE.EnvelopeId
group by
  SP.SchoolId, SP.SiteId, TRUNC(UTTimeMeasured,'MM')
```

1054 View SITE_PHOTO_VU2

1054.1 SQL Query for view SITE_PHOTO_VU2

```
select
 SITE_PHOTO_VU1.*,
 (qualspcnt+quanspcnt+atspcnt+bispcnt+scspcnt+smspcnt+swspcnt+phspcnt+schspcnt ) AS PhotoCnt
from SITE_PHOTO_VU1
```

1055 View SITE_PROXIMITY_DR1

1055.1 SQL Query for view SITE_PROXIMITY_DR1

```
select TRUNC(UTtimestamp,'DD') as Day, 'PX' as Measurecode, COUNT(RelatedSiteId) as RelSiteCnt,
COUNT(Distance) as DistCnt, COUNT(Direction) as DirCnt, COUNT(DECODE(DirectionSource,1,1,NULL)) as
DirSCnt, COUNT(ElevChange) as EleCnt, COUNT(DECODE(CloseByFlag,'1',1,NULL)) AS CloseCntFROM
GLOBE.SITE_PROXIMITYWHERE UTtimestamp < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTtimestamp,'DD')
```

1056 View SITE_PROXIMITY_DR2

1056.1 SQL Query for view SITE_PROXIMITY_DR2

```
select SITE_PROXIMITY_DR1.*, (RelSiteCnt+DistCnt+DirCnt+DirSCnt+EleCnt+CloseCnt) as DixonCnt
from SITE_PROXIMITY_DR1
```

1057 View SITE_PROXIMITY_DU2

1057.1 SQL Query for view SITE_PROXIMITY_DU2

```
select *
from SITE_PROXIMITY_DR2
```

1058 View SITE_PROXIMITY_MC1

1058.1 SQL Query for view SITE_PROXIMITY_MC1

```
select 'PX' as Measurecode, COUNT(RelatedSiteId) as RelSiteCnt, COUNT(Distance) as DistCnt,
COUNT(Direction) as DirCnt, COUNT(DECODE(DirectionSource,1,1,NULL)) as DirSCnt, COUNT(ElevChange) as
EleCnt, COUNT(DECODE(CloseByFlag,'1',1,NULL)) AS CloseCntFROM as GLOBE.SITE_PROXIMITY
```

1059 View SITE_PROXIMITY_MC2

1059.1 SQL Query for view SITE_PROXIMITY_MC2

```
select SITE_PROXIMITY_MC1.*, (RelSiteCnt+DistCnt+DirCnt+DirSCnt+EleCnt+CloseCnt) as DixonCnt
from SITE_PROXIMITY_MC1
```

1060 View SITE_PROXIMITY_RPRT_VU

1060.1 SQL Query for view SITE_PROXIMITY_RPRT_VU

```
select Distinct as SchoolID, 'PX' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from SITE_PROXIMITY
group by SchoolID
```

1061 View SITE_PROXIMITY_VU1

1061.1 SQL Query for view SITE_PROXIMITY_VU1

```
select
 SchoolId,
 KnownSiteId AS SiteId,
 TRUNC(UTtimestamp,'MM') AS Month,
 'PX' AS Measurecode,
 COUNT(*) AS RowCnt,
 COUNT(RelatedSiteId) AS RelSiteCnt,
 COUNT(Distance) AS DistCnt,
 COUNT(Direction) AS DirCnt,
 COUNT(DECODE(DirectionSource,1,1,NULL)) AS DirSCnt,
 COUNT(ElevChange) AS EleCnt,
 COUNT(DECODE(CloseByFlag,'1',1,NULL)) AS CloseCnt,
 MIN(UTtimestamp) AS BegDate,
 MAX(UTtimestamp) AS EndDate,
 MIN(UTtimestamp) AS BegRDate,
 MAX(UTtimestamp) AS EndRDate,
 FROM GLOBE.SITE_PROXIMITY
 WHERE UTtimestamp < TRUNC (SYSDATE,'DD') + 22/24
 GROUP BY SchoolId, KnownSiteId, TRUNC(UTtimestamp,'MM')
```

1062 View SITE_PROXIMITY_VU2

1062.1 SQL Query for view SITE_PROXIMITY_VU2

```
select SITE_PROXIMITY_VU1.*, (RelSiteCnt+DistCnt+DirCnt+DirSCnt+EleCnt+CloseCnt) as DixonCnt
from SITE_PROXIMITY_VU1
```

1063 View SKY_OBSCURED_VU1

1063.1 SQL Query for view SKY_OBSCURED_VU1

```
SELECT
  SchoolId,
  SiteId,
  TRUNC(SolarDay,'MM') AS Month,
  'CO' as MeasureCode,
  COUNT(*) as RowCnt,
  COUNT(Fog) as SOTypCnt,
  SUM(Fog) as FogSum,
  SUM(Smoke) as SmkSum,
  SUM(Haze) as HazSum,
  SUM(VolcanicAsh) as VoASum,
  SUM(Dust) as DusSum,
  SUM(Sand) as SanSum,
  SUM(Spray) as SprSum,
  SUM(HeavyRain) as HRnSum,
  SUM(HeavySnow) as HSnSum,
  SUM(BlowingSnow) as BSnSum,
  MIN(SolarDay+Solarhour/24.+SolarMin/(24.*60.)) AS BegDate,
  MAX(SolarDay+Solarhour/24.+SolarMin/(24.*60.)) AS EndDate,
  MIN(UTtimereported) AS BegRDate,
  MAX(UTtimereported) AS EndRDate,
  ROUND(AVG(UTTimeReported-(SolarDay+SolarHour/24.+SolarMin/(24.*60.))),2)
                            AS LagTimeAvg
FROM
  SKY_OBSCURED
WHERE
  UTTimeReported < TRUNC(SYSDATE,'DD') + 22/24
GROUP BY
  SchoolId, SiteId, TRUNC(Solarday,'MM')
```

1064 View SKY_OBSCURED_VU2

1064.1 SQL Query for view SKY_OBSCURED_VU2

```
SELECT
  SKY_OBSCURED_VU1.*,
  SOTypCnt as DixonCnt
FROM
  SKY_OBSCURED_VU1
```

1065 View SMGRAV_WS_DR1

1066 View SMGRAV_WS_DR2

1066.1 SQL Query for view SMGRAV_WS_DR2

```
select SMGRAV_WS_DR1.*, 3*gwccnt as DixonCnt
from SMGRAV_WS_DR1
```

1067 View SMGRAV_WS_DU1

1068 View SMGRAV_WS_DU2

1068.1 SQL Query for view SMGRAV_WS_DU2

```
select SMGRAV_WS_DU1.*, 3*gwccnt as DixonCnt
from SMGRAV_WS_DU1
```

1069 View SMGRAV_WS_MC1

1069.1 SQL Query for view SMGRAV_WS_MC1

```
select 'SM' as MeasureCode, COUNT(WaterContent) as GWCCnt
from GLOBE.SMGRAV_WS
```

1070 View SMGRAV_WS_MC2

1070.1 SQL Query for view SMGRAV_WS_MC2

```
select SMGRAV_WS_MC1.*, 3*gwccnt as DixonCnt
from SMGRAV_WS_MC1
```

1071 View SMGRAV_WS_RPRT_VU

1071.1 SQL Query for view SMGRAV_WS_RPRT_VU

```
select Distinct as SchoolID, 'SM' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from SMGRAV_WS
group by SchoolID
```

1072 View SMGRAV_WS_VU1

1072.1 SQL Query for view SMGRAV_WS_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'SM' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(WaterContent) AS GWCCnt,
 MIN(DayMeasured) AS BegDate,
 MAX(DayMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-DayMeasured),2) AS LagTimeAvg
from GLOBE.SMGRAV_WS
where UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1073 View SMGRAV_WS_VU2

1073.1 SQL Query for view SMGRAV_WS_VU2

```
select SMGRAV_WS_VU1.*, 3*gwccnt as DixonCnt
from SMGRAV_WS_VU1
```

1074 View SMGYPS_WS_DR1

1075 View SMGYPS_WS_DR2

1075.1 SQL Query for view SMGYPS_WS_DR2

```
select SMGYPS_WS_DR1.*, bsmcnt as DixonCnt
from SMGYPS_WS_DR1
```

1076 View SMGYPS_WS_DU1

1077 View SMGYPS_WS_DU2

1077.1 SQL Query for view SMGYPS_WS_DU2

```
select SMGYPS_WS_DU1.*, bsmcnt as DixonCnt
from SMGYPS_WS_DU1
```

1078 View SMGYPS_WS_MC1

1078.1 SQL Query for view SMGYPS_WS_MC1

```
select 'SM' as MeasureCode, COUNT(WaterContent) as BSMCnt
from GLOBE.SMGYPS_WS
```

1079 View SMGYPS_WS_MC2

1079.1 SQL Query for view SMGYPS_WS_MC2

```
select SMGYPS_WS_MC1.*, bsmcnt as DixonCnt
from SMGYPS_WS_MC1
```

1080 View SMGYPS_WS_RPRT_VU

1080.1 SQL Query for view SMGYPS_WS_RPRT_VU

```
select Distinct as SchoolID, 'SM' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from SMGYPS_WS
group by SchoolID
```

1081 View SMGYPS_WS_VU1

1081.1 SQL Query for view SMGYPS_WS_VU1

```
select
 SCHOOLID,
 SITEID,
 TRUNC(DayMeasured,'MM') AS Month,
 'SM' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT(WaterContent) AS BSMCnt,
 MIN(DayMeasured) AS BegDate,
 MAX(DayMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-DayMeasured),2) AS LagTimeAvg
from SMGYPS_WS
where UTTIMEREPORTED < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by SCHOOLID, SITEID, TRUNC(DayMeasured,'MM')
```

1082 View SMGYPS_WS_VU2

1082.1 SQL Query for view SMGYPS_WS_VU2

```
select SMGYPS_WS_VU1.*, bsmcnt as DixonCnt
from SMGYPS_WS_VU1
```

1083 View SMT_Block_HR1

1083.1 SQL Query for view SMT_Block_HR1

```
select DISTINCT
 SchoolId, SiteId, DayMeasured, Depth
from GLOBE.SOIL_MOISTURE
where
 Protocol IN ( 4 , 5 , 6 ) AND
 WaterContent IS NOT NULL AND
 UTtimeReported > = TO_DATE ( '19990201' , 'YYYYMMDD' ) AND
 DayMeasured > = TO_DATE ( '19990201' , 'YYYYMMDD' )
```

1084 View SMT_Block_HR2

1084.1 SQL Query for view SMT_Block_HR2

```
select
```

```
2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.DayMeasured) as BeginDay,
MAX(T.DayMeasured) as EndDay,
COUNT(T.Depth) as DepthCnt
from SMT_Block_HR1 T
Group By T.SchoolId, T.SiteId,
2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('19990201','YYYYMMDD'))/4,1))+1
HAVING
 COUNT(T.Depth) >= 4*30*(0.70)
AND
2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('19990201','YYYYMMDD'))/4,1))+1>= 1
AND
2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('19990201','YYYYMMDD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.DayMeasured) as BeginDay,
MAX(T.DayMeasured) as EndDay,
COUNT(T.Depth) as DepthCnt
from SMT_Block_HR1 T
Group By T.SchoolId, T.SiteId,
2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('19990201','YYYYMMDD'))+2)/4,1))
HAVING
 COUNT(T.Depth) >= 4*30*(0.70)
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) >= 1
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('19990201','YYYYMMDD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('19990401','YYYYMMDD')))/2)
```

1085 View SMT_Diurnal_HR1

1085.1 SQL Query for view SMT_Diurnal_HR1

```
select SchoolId, SiteId, TRUNC(SolarDayMeas) as SolarDay, Depth, COUNT(SampleNbr) as SampleCnt
from GLOBE.SOILTEMP_DIURNAL
where SoilTemp  IS  NOT  NULL  AND  SolarDayMeas > =  TO_DATE  (
'19990201'
,
'YYYYMMDD'
)
having COUNT  (
SampleNbr )  > =  5
group by SchoolId, SiteId, SolarDayMeas, Depth
```

1086 View SMT_Diurnal_HR2

1086.1 SQL Query for view SMT_Diurnal_HR2

```
select DISTINCT as D1.SchoolId, D1.SiteId, D1.SolarDay, D2.SolarDay as NextDay,
DECODE(SL.DataSource,0,'Unchecked',1,'GPS',2,'Other',3,'Default') as LocMethod
from SMT_Diurnal_HR1 D1, SMT_Diurnal_HR1 D2
where D1.SchoolId = D2.SchoolId  AND  D1.SiteId  =  D2.SiteId  AND  D1.SolarDay  +  1  =  D2.SolarDay
AND  D1.SchoolId  =  SL.SchoolId  AND  D1.SiteId  =  SL.SiteId
```

1087 View SMT_Diurnal_HR3

1087.1 SQL Query for view SMT_Diurnal_HR3

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.SolarDay) as BeginDay,
MAX(T.SolarDay)+1 as EndDay,
COUNT(T.SolarDay)as DiurnalDays,
T.LocMethod
FROM SMT_Diurnal_HR2 T
Group By T.SchoolId, T.SiteId, T.LocMethod,
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
HAVING
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 >= 1
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.SolarDay) as BeginDay,
MAX(T.SolarDay) as EndDay,
COUNT(T.SolarDay)as DiurnalDays,
T.LocMethod
FROM SMT_Diurnal_HR2 T
Group By T.SchoolId, T.SiteId, T.LocMethod,
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
HAVING
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) >= 1
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.SolarDay,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1088 View SMT_GravHole_HR1

1088.1 SQL Query for view SMT_GravHole_HR1

```
select DISTINCT SchoolId, SiteId, DayMeasured, Depth
from GLOBE.SOIL_MOISTURE
where
 Protocol = 3 AND
 WaterContent IS NOT NULL AND
 UTtimeReported > = TO_DATE ( '19990201' , 'YYYYMMDD' ) AND
 DayMeasured > = TO_DATE ( '19990201' , 'YYYYMMDD' )
```

1089 View SMT_GravHole_HR2

1089.1 SQL Query for view SMT_GravHole_HR2

```
select
2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.DayMeasured) as BeginDay,
MAX(T.DayMeasured) as EndDay,
COUNT(T.Depth) as DepthCnt
FROM SMT_GravHole_HR1 T
Group By T.SchoolId, T.SiteId,
2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
```

```
HAVING
 COUNT(T.Depth) >= 4*4
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 >= 1
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.DayMeasured) as BeginDay,
MAX(T.DayMeasured) as EndDay,
COUNT(T.Depth) as DepthCnt
FROM SMT_GravHole_HR1 T
Group By T.SchoolId, T.SiteId,
2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
HAVING
 COUNT(T.Depth) >= 4*4
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))>= 1
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1090 View SMT_GravStarInt_HR1

1090.1 SQL Query for view SMT_GravStarInt_HR1

```
select DISTINCT
 SchoolId, SiteId, DayMeasured, Depth, SampleNbr
from GLOBE.SMGRAV_WS
where
 Protocol IN ( 1 , 2 ) AND
 WaterContent IS NOT NULL AND
 UTtimeReported > = TO_DATE ( '19990201' , 'YYYYMMDD' ) AND
 DayMeasured > = TO_DATE ( '19990201' , 'YYYYMMDD' )
```

1091 View SMT_GravStarInt_HR2

1091.1 SQL Query for view SMT_GravStarInt_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.DayMeasured) as BeginDay,
MAX(T.DayMeasured) as EndDay,
COUNT(T.Depth) as DepthCnt
FROM SMT_GravStarInt_HR1 T
Group By T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
HAVING
 COUNT(T.Depth) >= 4*2*5
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 >= 1
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
```

```
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
T.SchoolId,
T.SiteId,
MIN(T.DayMeasured) as BeginDay,
MAX(T.DayMeasured) as EndDay,
COUNT(T.Depth) as DepthCnt
FROM SMT_GravStarInt_HR1 T
Group By T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
HAVING
 COUNT(T.Depth) >= 4*2*5
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) >= 1
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.DayMeasured,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1092 View SMT_Logger_HR1

1092.1 SQL Query for view SMT_Logger_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(UTtimeMeasured,'WW') as Week,
 Count(SoilTemp5cm) as Temp5cmCnt,
 Count(SoilTemp10cm) as Temp10cmCnt,
 Count(SoilTemp50cm) as Temp50cmCnt
from GLOBE.TEMPERATURE_LOGGER L
group by SchoolId, SiteId, TRUNC(UTtimeMeasured,'WW')
```

1093 View SMT_Logger_HR2

1093.1 SQL Query for view SMT_Logger_HR2

```
select
 L.SchoolId,
 L.SiteId,
 L.Week,
 DECODE(L.Temp5cmCnt,0,0,1)+DECODE(L.Temp10cmCnt,0,0,1)+DECODE(L.Temp50cmCnt,0,0,1) as DepthCnt
from
 SMT_Logger_HR1 L,
 GLOBE.SITE_LOCATION SL
where
 L.SchoolId = SL.SchoolId AND
 L.SiteId = SL.SiteId AND
 SL.DataSource in (1,2)
```

1094 View SMT_Moisture_HR1

1094.1 SQL Query for view SMT_Moisture_HR1

```
Select
 HRnbr,
 SchoolId,
 SiteId,
 BeginDay,
 EndDay,
 'Star' as Protocol,
 DepthCnt
from SMT_GravStarInt_HR2
UNION ALL
Select
 HRnbr,
 SchoolId,
 SiteId,
 BeginDay,
 EndDay,
```

```
 'Hole' as Method,
 DepthCnt
from SMT_GravHole_HR2
UNION ALL
Select
 HRnbr,
 SchoolId,
 SiteId,
 BeginDay,
 EndDay,
 'Block' as Method,
 DepthCnt
from SMT_Block_HR2
```

1095 View SMT_Moisture_HR2

1095.1 SQL Query for view SMT_Moisture_HR2

```
SELECT
 SM.HRnbr,
 SM.SchoolId,
 SM.SiteId,
 SM.Protocol,
 MAX(SM.DepthCnt) as DepthCnt,
 TRUNC(MIN(SM.BeginDay)) as BeginMonth,
 TRUNC(MAX(SM.EndDay)) as EndMonth,
 MAX(DECODE(SL.DataSource,0,'Unchecked',1,'GPS',2,'Other')) as LocMethod
FROM
 SMT_Moisture_HR1 SM,
 GLOBE.SITE_LOCATION SL
WHERE
SM.SchoolID = SL.SchoolId
AND
SM.SiteID = SL.SiteId
GROUP BY HRnbr, SM.SchoolId, SM.SiteId, SM.Protocol
```

1096 View SMT_Temp_HR1

1096.1 SQL Query for view SMT_Temp_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDayMeas,'WW') as Week,
 COUNT(distinct Depth) as DepthCnt
from
 (select DISTINCT SchoolId, SiteId, SolarDayMeas, Depth
  from GLOBE.SOIL_TEMP
  where SoilTemp IS NOT NULL
        AND
        UTtimeReported >= TO_DATE('19990201','YYYYMMDD')
        AND
        UTtimeMeasured >= TO_DATE('19990201','YYYYMMDD'))
group by SchoolId, SiteId, TRUNC(SolarDayMeas,'WW')
UNION
select * from SMT_Logger_HR2
where DepthCnt > 0
```

1097 View SMT_Temp_HR2

1097.1 SQL Query for view SMT_Temp_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 as HRnbr,
T.SchoolId,
T.SiteId,
COUNT(T.Week) as WeekCnt,
MIN(T.Week) as BeginWeek,
MAX(T.Week) as EndWeek
FROM SMT_Temp_HR1 T
WHERE T.DepthCnt >= 2
Group By T.SchoolId, T.SiteId,
```

```
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
HAVING
 COUNT(T.Week) >= 4
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 >= 1
AND
 2*TRUNC(Round(MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1 <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
T.SchoolId,
T.SiteId,
COUNT(T.Week) as WeekCnt,
MIN(T.Week) as BeginWeek,
MAX(T.Week) as EndWeek
FROM SMT_Temp_HR1 T
WHERE T.DepthCnt >= 2
Group By T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
HAVING
 COUNT(T.Week) >= 4
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) >= 1
AND
 2*TRUNC(Round((MONTHS_BETWEEN(TRUNC(T.Week,'MM'),
 TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1098 View SMT_XAllThree_HR1

1098.1 SQL Query for view SMT_XAllThree_HR1

```
select M.HRnbr, M.SchoolId, M.Protocol, M.DepthCnt, D.DiurnalDays, T.WeekCnt, M.LocMethod as
MoistLocMthd, D.LocMethod as DiurnalLocMthd
from SMT_Moisture_HR2 M, SMT_Diurnal_HR3 D, SMT_Temp_HR2 T
where M.HRnbr = T.HRnbr  AND  M.SchoolId  =  T.SchoolId  AND  M.HRnbr  =  D.HRnbr  AND  M.SchoolId  =
D.SchoolId
```

1099 View SMT_XAllThree_HR2

1099.1 SQL Query for view SMT_XAllThree_HR2

```
select distinct HRnbr, SchoolId
from SMT_XAllThree_HR1
where WeekCnt > =  12  AND  MoistLocMthd  IN  (
'GPS'
,
'Other'
)  AND  DiurnalLocMthd  IN  (
'GPS'
,
'Other'
)
```

1100 View SMT_XAllThree_HR3

1100.1 SQL Query for view SMT_XAllThree_HR3

```
select SMT.HRnbr, SMT.SchoolId, SUM(DSV.DixonCnt) as DixonCnt, MIN(
TRUNC(TO_DATE('19990201','YYYYMMDD')+ 2*(SMT.HRnbr-1)*31,'MM')) as BeginMonth,
MAX(TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(SMT.HRnbr+1)-1)*31,'MM')) as EndMonth
from SMT_XAllThree_HR2 SMT, GLOBE.DIXON_SUMMARY_VU2 DSV
where SMT.SchoolId = DSV.SchoolId  AND  DSV.MeasureCode  IN  (
```

```
'SM','ST'
)  AND  DSV.Month  >=  TRUNC  (
TO_DATE
(
'19990201','YYYYMMDD'
)
+
2*
(
SMT.HRnbr-1
)
*31,'MM'
)  AND  DSV.Month  <=  TRUNC  (
TO_DATE
(
'19990201','YYYYMMDD'
)
+
(
2*
(
SMT.HRnbr+1
)
-1
)
*31,'MM'
)
group by SMT.HRnbr, SMT.SchoolId
```

1101 View SOILCHAR_HR1

1101.1 SQL Query for view SOILCHAR_HR1

```
Select
 SchoolId,
 SiteId,
 HorizonNbr as HzNbr,
 TopDepth as Top,
 BottomDepth as Bot,
 MainColorCode as Color,
 TextureField as TexF,
 TextureUSDA as TexU,
 RockEstimate as RockEst,
 RootEstimate as RootEst,
 Carbonates as Carb,
 BulkDensity as BlkDen,
 NitrogenEst as NitEst,
 PhosphorusEst as PhosEst,
 PotassiumEst as PotEst,
 Acidity as Ph,
 NVL(SandPct96,1)*NVL(SandPctUSDA,1)*NVL(SandPctISSS,1) as Sand,
 NVL(SiltPct96,1)*NVL(SiltPctUSDA,1)*NVL(SiltPctISSS,1) as Silt,
 NVL(ClayPct96,1)*NVL(ClayPctHyd,1) as Clay,
 TRUNC(UTtimeMeasured) as Day
From GLOBE.SOIL_CHAR
Where
 MainColorCode Is Not Null AND
 (TextureField <> 0 OR TextureUSDA <> 0) AND
 RockEstimate <> 0 AND
 RootEstimate <> 0 AND
 Carbonates <> 0 AND
 BulkDensity Is Not Null AND
 NitrogenEst <> 0 AND
 PhosphorusEst <> 0 AND
 PotassiumEst <> 0 AND
 Acidity Is Not Null AND
 NVL(SandPct96,1)*NVL(SandPctUSDA,1)*NVL(SandPctISSS,1) <> 1 AND
 NVL(SiltPct96,1)*NVL(SiltPctUSDA,1)*NVL(SiltPctISSS,1) <> 1 AND
 NVL(ClayPct96,1)*NVL(ClayPctHyd,1) <> 1
```

1102 View SOILCHAR_HR2

1102.1 SQL Query for view SOILCHAR_HR2

```
Select
```

```
 SchoolId,
 SiteId,
 COUNT(HzNbr) as FullHrzNbr,
 MIN(Top) as Top,
 MAX(Bot) as Bot,
 COUNT(BlkDen) as BlkDenNbr,
 COUNT(DECODE(Sand,1,null)) as SandNbr,
 COUNT(DECODE(Silt,1,null)) as SiltNbr,
 COUNT(DECODE(Clay,1,null)) as ClayNbr,
 Max(Day) as Day
From SOILCHAR_HR1
Group By SchoolId, SiteId
```

1103 View SOILCHAR_HR2_Loc

1103.1 SQL Query for view SOILCHAR_HR2_Loc

```
select A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot, A.Day
from SOILCHAR_HR2 A
where A.SchoolId = B.SchoolId  AND  A.SiteId  =  B.SiteId  AND  B.DataSource  IN  (
1
,
2
)
```

1104 View SOILCHAR_HR3

1104.1 SQL Query for view SOILCHAR_HR3

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILCHAR_HR2_Loc A
 Group By A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILCHAR_HR2_Loc A
 Group By A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILCHAR_HR2_Loc A
 Group By A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
```

```
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILCHAR_HR2_Loc A
 Group By A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILCHAR_HR2_Loc A
 Group By A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILCHAR_HR2_Loc A
 Group By A.SchoolId, A.SiteId, A.FullHrzNbr, A.Bot,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1105 View SOILCHAR_HR4

1105.1 SQL Query for view SOILCHAR_HR4

```
Select DISTINCT
 HRnbr,
 SchoolId,
 TRUNC(TO_DATE('19980601','YYYYMMDD')+2*(HRnbr-1)*31,'MM') as BeginMonth,
 TRUNC(TO_DATE('19980601','YYYYMMDD')+(2*(HRnbr+5)-1)*31,'MM') as EndMonth
From
 SOILCHAR_HR3
Where
 Bot > 95 OR FullHrzNbr >= 4
UNION
Select
```

```
 HRnbr,
 SchoolId,
 TRUNC(TO_DATE('19980601','YYYYMMDD')+2*(HRnbr-1)*31,'MM') as BeginMonth,
 TRUNC(TO_DATE('19980601','YYYYMMDD')+(2*(HRnbr+5)-1)*31,'MM') as EndMonth
From
 SOILCHAR_HR3
Where Bot > 38
Group By
 HRnbr,
 SchoolId,
 TRUNC(TO_DATE('19980601','YYYYMMDD')+2*(HRnbr-1)*31,'MM'),
 TRUNC(TO_DATE('19980601','YYYYMMDD')+(2*(HRnbr+5)-1)*31,'MM')
Having COUNT(SiteId) >= 2
```

1106 View SOILCHAR_HR5

1106.1 SQL Query for view SOILCHAR_HR5

```
select A.HRnbr, A.SchoolId, SUM(B.DixonCnt) as DixonCnt, A.BeginMonth, A.EndMonth
from SOILCHAR_HR4 A, GLOBE.SOIL_CHAR_CNT B
where A.SchoolId = B.SchoolId  AND  B.Month  BETWEEN  A.Beg  in  Month  AND  A.EndMonth
group by A.HRnbr, A.SchoolId, A.BeginMonth, A.EndMonth
```

1107 View SOILCHAR_SITE_DR1

1107.1 SQL Query for view SOILCHAR_SITE_DR1

```
select TRUNC(UTtimeStamp,'DD') as Day, 'MD' as MeasureCode, COUNT(SlopeOfSite) as SlopeCnt,
COUNT(DECODE(SoilSource,0,NULL,1)) as SSrceCnt, COUNT(DECODE(SiteProximity,0,NULL,1)) as SProxCnt,
COUNT(DECODE(ParentMaterial,0,NULL,1)) AS SParCntFROM GLOBE.SOILCHAR_SITEWHERE UTtimeStamp < TRUNC
(SYSDATE,'DD') + 22/24GROUP BY TRUNC as (UTtimeStamp,'DD')
```

1108 View SOILCHAR_SITE_DR2

1108.1 SQL Query for view SOILCHAR_SITE_DR2

```
select SOILCHAR_SITE_DR1.*, (slopecnt+ssrcecnt+sproxcnt+sparcnt) as DixonCnt
from SOILCHAR_SITE_DR1
```

1109 View SOILCHAR_SITE_DU1

1109.1 SQL Query for view SOILCHAR_SITE_DU1

```
select TRUNC(VersionDate,'DD') as Day, 'MD' as MeasureCode, COUNT(SlopeOfSite) as SlopeCnt,
COUNT(DECODE(SoilSource,0,NULL,1)) as SSrceCnt, COUNT(DECODE(SiteProximity,0,NULL,1)) as SProxCnt,
COUNT(DECODE(ParentMaterial,0,NULL,1)) AS SParCntFROM GLOBE.SOILCHAR_SITEWHERE UTtimeStamp < TRUNC
(SYSDATE,'DD') + 22/24GROUP BY TRUNC as (VersionDate,'DD')
```

1110 View SOILCHAR_SITE_DU2

1110.1 SQL Query for view SOILCHAR_SITE_DU2

```
select SOILCHAR_SITE_DU1.*, (slopecnt+ssrcecnt+sproxcnt+sparcnt) as DixonCnt
from SOILCHAR_SITE_DU1
```

1111 View SOILCHAR_SITE_MC1

1111.1 SQL Query for view SOILCHAR_SITE_MC1

```
select 'MD' as MeasureCode, COUNT(SlopeOfSite) as SlopeCnt, COUNT(DECODE(SoilSource,0,NULL,1)) as
SSrceCnt, COUNT(DECODE(SiteProximity,0,NULL,1)) as SProxCnt, COUNT(DECODE(ParentMaterial,0,NULL,1)) AS
SParCntFROM as GLOBE.SOILCHAR_SITE
```

1112 View SOILCHAR_SITE_MC2

1112.1 SQL Query for view SOILCHAR_SITE_MC2

```
select SOILCHAR_SITE_MC1.*, (slopecnt+ssrcecnt+sproxcnt+sparcnt) as DixonCnt
from SOILCHAR_SITE_MC1
```

1113 View SOILCHAR_SITE_RPRT_VU

1113.1 SQL Query for view SOILCHAR_SITE_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from SOILCHAR_SITE
group by SchoolID
```

1114 View SOILCHAR_SITE_VU1

1114.1 SQL Query for view SOILCHAR_SITE_VU1

```
select
 Schoolid,
 Siteid,
 TRUNC(VersionDate,'MM') AS Month,
 'MD' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(SlopeOfSite) AS SlopeCnt,
 COUNT(DECODE(SoilSource,0,NULL,1)) AS SSrceCnt,
 COUNT(DECODE(SiteProximity,0,NULL,1)) AS SProxCnt,
 COUNT(DECODE(ParentMaterial,0,NULL,1)) AS SParCnt,
 MIN(VersionDate) AS BegDate,
 MAX(VersionDate) AS EndDate,
 MIN(UTtimestamp) AS BegRDate,
 MAX(UTtimestamp) AS EndRDate,
 ROUND(AVG(UTtimestamp-VersionDate),2) AS LagTimeAvg
FROM GLOBE.SOILCHAR_SITEWHERE UTtimeStamp < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(VersionDate,'MM')
```

1115 View SOILCHAR_SITE_VU2

1115.1 SQL Query for view SOILCHAR_SITE_VU2

```
select SOILCHAR_SITE_VU1.*, (slopecnt+ssrcecnt+sproxcnt+sparcnt) as DixonCnt
from SOILCHAR_SITE_VU1
```

1116 View SOILINFIL_HR1

1116.1 SQL Query for view SOILINFIL_HR1

```
select A.SchoolId, A.SiteId, F.DayMeasured as Day, Count(F.SampleNbr) as NbrSamples,
Count(F.SatWaterContent) as NbrWaterContents
from SOILCHAR_HR2 A
group by A.SchoolId, A.SiteId, F.DayMeasured
```

1117 View SOILINFIL_HR2

1117.1 SQL Query for view SOILINFIL_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From SOILINFIL_HR1 A
 Where A.NbrSamples >= 1
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
```

```
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From SOILINFIL_HR1 A
Where A.NbrSamples >= 1
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From SOILINFIL_HR1 A
Where A.NbrSamples >= 1
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From SOILINFIL_HR1 A
Where A.NbrSamples >= 1
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From SOILINFIL_HR1 A
Where A.NbrSamples >= 1
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
```

```
TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
A.SchoolId, A.SiteId,
MIN(TRUNC(Day,'MM')) as BeginMonth,
MAX(TRUNC(Day,'MM')) as EndMonth
From SOILINFIL_HR1 A
Where A.NbrSamples >= 1
Group By A.SchoolId, A.SiteId,
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
HAVING
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
(6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1118 View SOILINFIL_HR3

1118.1 SQL Query for view SOILINFIL_HR3

```
select DISTINCT A.HRnbr, A.SchoolId
from SOILINFIL_HR2 A, SOILCHAR_HR3 B
where A.HRnbr = B.HRnbr  AND  A.SchoolId  =  B.SchoolId  AND  A.SiteId  =  B.SiteId
```

1119 View SOILMOISTURE_SITE_DR1

1119.1 SQL Query for view SOILMOISTURE_SITE_DR1

```
select TRUNC(UTtimeStamp,'DD') as Day, 'MD' as MeasureCode, SUM(DECODE(SurfaceType,0,0,1)) as
SurfTypCnt, SUM(DECODE(SurfaceCover,0,0,1)) as SurfCovCnt, SUM(DECODE(CanopyCover,0,0,1)) as
CanCovCnt, COUNT(SandPct) as SandCnt, COUNT(SiltPct) as SiltCnt, Count(ClayPct) as ClayCnt,
SUM(DECODE(RockEstimate,0,0,1)) as RockCnt, SUM(DECODE(RootEstimate,0,0,1)) AS RootCntFROM
GLOBE.SOILMOISTURE_SITEWHERE UTtimeStamp < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(UTtimeStamp,'DD')
```

1120 View SOILMOISTURE_SITE_DR2

1120.1 SQL Query for view SOILMOISTURE_SITE_DR2

```
select SOILMOISTURE_SITE_DR1.*,
(surftypcnt+surfcovcnt+cancovcnt+sandcnt+siltcnt+claycnt+rockcnt+rootcnt) as DixonCnt
from SOILMOISTURE_SITE_DR1
```

1121 View SOILMOISTURE_SITE_DU1

1121.1 SQL Query for view SOILMOISTURE_SITE_DU1

```
select TRUNC(VersionDate,'DD') as Day, 'MD' as MeasureCode, SUM(DECODE(SurfaceType,0,0,1)) as
SurfTypCnt, SUM(DECODE(SurfaceCover,0,0,1)) as SurfCovCnt, SUM(DECODE(CanopyCover,0,0,1)) as
CanCovCnt, COUNT(SandPct) as SandCnt, COUNT(SiltPct) as SiltCnt, Count(ClayPct) as ClayCnt,
SUM(DECODE(RockEstimate,0,0,1)) as RockCnt, SUM(DECODE(RootEstimate,0,0,1)) AS RootCntFROM
GLOBE.SOILMOISTURE_SITEWHERE UTtimeStamp < TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as
(VersionDate,'DD')
```

1122 View SOILMOISTURE_SITE_DU2

1122.1 SQL Query for view SOILMOISTURE_SITE_DU2

```
select SOILMOISTURE_SITE_DU1.*,
(surftypcnt+surfcovcnt+cancovcnt+sandcnt+siltcnt+claycnt+rockcnt+rootcnt) as DixonCnt
from SOILMOISTURE_SITE_DU1
```

1123 View SOILMOISTURE_SITE_MC1

1123.1 SQL Query for view SOILMOISTURE_SITE_MC1

```
select 'MD' as MeasureCode, SUM(DECODE(SurfaceType,0,0,1)) as SurfTypCnt,
SUM(DECODE(SurfaceCover,0,0,1)) as SurfCovCnt, SUM(DECODE(CanopyCover,0,0,1)) as CanCovCnt,
COUNT(SandPct) as SandCnt, COUNT(SiltPct) as SiltCnt, Count(ClayPct) as ClayCnt,
SUM(DECODE(RockEstimate,0,0,1)) as RockCnt, SUM(DECODE(RootEstimate,0,0,1)) AS RootCntFROM as
GLOBE.SOILMOISTURE_SITE
```

1124 View SOILMOISTURE_SITE_MC2

1124.1 SQL Query for view SOILMOISTURE_SITE_MC2

```
select SOILMOISTURE_SITE_MC1.*,
(surftypcnt+surfcovcnt+cancovcnt+sandcnt+siltcnt+claycnt+rockcnt+rootcnt) as DixonCnt
from SOILMOISTURE_SITE_MC1
```

1125 View SOILMOISTURE_SITE_RPRT_VU

1125.1 SQL Query for view SOILMOISTURE_SITE_RPRT_VU

```
select Distinct as SchoolID, 'SM' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from SOILMOISTURE_SITE
group by SchoolID
```

1126 View SOILMOISTURE_SITE_VU1

1126.1 SQL Query for view SOILMOISTURE_SITE_VU1

```
select
 Schoolid,
 Siteid,
 TRUNC(VersionDate,'MM') AS Month,
 'MD' AS MeasureCode,
 COUNT(*) AS RowCnt,
 SUM(DECODE(SurfaceType,0,0,1)) AS SurfTypCnt,
 SUM(DECODE(SurfaceCover,0,0,1)) AS SurfCovCnt,
 SUM(DECODE(CanopyCover,0,0,1)) AS CanCovCnt,
 COUNT(SandPct) AS SandCnt,
 COUNT(SiltPct) AS SiltCnt,
 Count(ClayPct) AS ClayCnt,
 SUM(DECODE(RockEstimate,0,0,1)) AS RockCnt,
 SUM(DECODE(RootEstimate,0,0,1)) AS RootCnt,
 MIN(VersionDate) AS BegDate,
 MAX(VersionDate) AS EndDate,
 MIN(UTtimestamp) AS BegRDate,
 MAX(UTtimestamp) AS EndRDate,
 ROUND(AVG(UTtimestamp-VersionDate),2) AS LagTimeAvg
FROM GLOBE.SOILMOISTURE_SITE
WHERE UTtimeStamp < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(VersionDate,'MM')
```

1127 View SOILMOISTURE_SITE_VU2

1127.1 SQL Query for view SOILMOISTURE_SITE_VU2

```
select SOILMOISTURE_SITE_VU1.*,
(surftypcnt+surfcovcnt+cancovcnt+sandcnt+siltcnt+claycnt+rockcnt+rootcnt) as DixonCnt
from SOILMOISTURE_SITE_VU1
```

1128 View SOILTEMP_DAILYAVG

1128.1 SQL Query for view SOILTEMP_DAILYAVG

```
select SCHOOLID, SITEID, SOLARDAYMEAS, DEPTH, AVG(SOIL_TEMP.SOILTEMP) as AvgTemp,
COUNT(SOIL_TEMP.SOILTEMP) as CntInAvgTemp, Avg(SOIL_TEMP.AIRTEMP) as AvgAirTemp,
MIN(SOIL_TEMP.SOLARHOURMEAS) as BeginSolarHour, MAX(SOIL_TEMP.SOLARHOURMEAS) as EndSolarHour,
Min(SOIL_TEMP.UTTIMEMEASURED) as BeginUTtimeMeas, Max(SOIL_TEMP.UTTIMEMEASURED) as EndUTtimeMeas
from SOIL_TEMP
where TEMPUSAGE = 1 or TEMPUSAGE = 3
having Count ( SoilTemp )  > 0
group by SCHOOLID, SITEID, SOLARDAYMEAS, DEPTH
```

1129 View SOILTEMP_DIURNAL

1129.1 SQL Query for view SOILTEMP_DIURNAL

```
select SCHOOLID, SITEID, SOLARDAYMEAS, DEPTH, SAMPLENBR, SOLARHOURMEAS, SOLARMINMEAS, SOILTEMP,
AIRTEMP, UTTIMEMEASURED
from SOIL_TEMP
where TEMPUSAGE = 2 or TEMPUSAGE = 3
```

1130 View SOIL_CHAR_DR1

1131 View SOIL_CHAR_DR2

1131.1 SQL Query for view SOIL_CHAR_DR2

```
select SOIL_CHAR_DR1.*,
(MoisCnt+StrucCnt+MColorCnt+SColorCnt+ConsCnt+TexFldCnt+RockCnt+RootCnt+CarbCnt+2*RowCnt) as DixonCnt
from SOIL_CHAR_DR1
```

1132 View SOIL_CHAR_DU1

1133 View SOIL_CHAR_DU2

1133.1 SQL Query for view SOIL_CHAR_DU2

```
select SOIL_CHAR_DU1.*,
(MoisCnt+StrucCnt+MColorCnt+SColorCnt+ConsCnt+TexFldCnt+RockCnt+RootCnt+CarbCnt+2*RowCnt) as DixonCnt
from SOIL_CHAR_DU1
```

1134 View SOIL_CHAR_MC1

1134.1 SQL Query for view SOIL_CHAR_MC1

```
select 'SC' as MeasureCode, Count(*) as RowCnt, COUNT(DECODE(MoistureEst,0,NULL,1)) as MoisCnt,
COUNT(DECODE(SoilStructureEst,0,NULL,1)) as StrucCnt, COUNT(MainColorCode) as MColorCnt,
COUNT(SecondColorCode) as SColorCnt, COUNT(DECODE(ConsistenceEst,0,NULL,1)) as ConsCnt,
COUNT(DECODE(TextureField,0,NULL,1)) as TexFldCnt, COUNT(DECODE(RockEstimate,0,NULL,1)) as RockCnt,
COUNT(DECODE(RootEstimate,0,NULL,1)) as RootCnt, COUNT(DECODE(Carbonates,0,NULL,1)) as CarbCnt
from GLOBE.SOIL_CHAR
```

1135 View SOIL_CHAR_MC2

1135.1 SQL Query for view SOIL_CHAR_MC2

```
select SOIL_CHAR_MC1.*,
(MoisCnt+StrucCnt+MColorCnt+SColorCnt+ConsCnt+TexFldCnt+RockCnt+RootCnt+CarbCnt+2*RowCnt) as DixonCnt
from SOIL_CHAR_MC1
```

1136 View SOIL_CHAR_RPRT_VU

1136.1 SQL Query for view SOIL_CHAR_RPRT_VU

```
select Distinct as SchoolID, 'SC' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from SOIL_CHAR
group by SchoolID
```

1137 View SOIL_CHAR_VU1

1137.1 SQL Query for view SOIL_CHAR_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(UTtimeMeasured,'MM') AS Month,
 'SC' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(DECODE(MoistureEst,0,NULL,1)) AS MoisCnt,
 COUNT(DECODE(SoilStructureEst,0,NULL,1)) AS StrucCnt,
 COUNT(MainColorCode) AS MColorCnt,
 COUNT(SecondColorCode) AS SColorCnt,
 COUNT(DECODE(ConsistenceEst,0,NULL,1)) AS ConsCnt,
 COUNT(DECODE(TextureField,0,NULL,1)) AS TexFldCnt,
 COUNT(DECODE(RockEstimate,0,NULL,1)) AS RockCnt,
 COUNT(DECODE(RootEstimate,0,NULL,1)) AS RootCnt,
 COUNT(DECODE(Carbonates,0,NULL,1)) AS CarbCnt,
 MIN(UTtimeMeasured) AS BegDate,
 MAX(UTtimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.SOIL_CHAR
where UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(UTtimeMeasured,'MM')
```

1138 View SOIL_CHAR_VU2

1138.1 SQL Query for view SOIL_CHAR_VU2

```
select SOIL_CHAR_VU1.*, (MoisCnt+StrucCnt+MColorCnt+SColorCnt+ConsCnt+TexFldCnt+
RockCnt+RootCnt+CarbCnt+2*RowCnt) as DixonCnt
from SOIL_CHAR_VU1
```

1139 View SOIL_INFILTRATION_DR1

1140 View SOIL_INFILTRATION_DR2

1140.1 SQL Query for view SOIL_INFILTRATION_DR2

```
select SOIL_INFILTRATION_DR1.*, (2*wlevcnt+2*rdiamcnt+3*swccnt) as DixonCnt
from SOIL_INFILTRATION_DR1
```

1141 View SOIL_INFILTRATION_DU1

1142 View SOIL_INFILTRATION_DU2

1142.1 SQL Query for view SOIL_INFILTRATION_DU2

```
select SOIL_INFILTRATION_DU1.*, (2*wlevcnt+2*rdiamcnt+3*swccnt) as DixonCnt
from SOIL_INFILTRATION_DU1
```

1143 View SOIL_INFILTRATION_MC1

1143.1 SQL Query for view SOIL_INFILTRATION_MC1

```
select 'SF' as MeasureCode, COUNT(*) as RowCnt, COUNT(SatWaterContent) as SWCCnt,
COUNT(UpperWaterLevel+LowerWaterLevel) as WLevCnt, COUNT(InnerRingDiam+OuterRingDiam) as RDiamCnt
from GLOBE.SOIL_INFILTRATION
```

1144 View SOIL_INFILTRATION_MC2

1144.1 SQL Query for view SOIL_INFILTRATION_MC2

```
select SOIL_INFILTRATION_MC1.*, (2*wlevcnt+2*rdiamcnt+3*swccnt) as DixonCnt
from SOIL_INFILTRATION_MC1
```

1145 View SOIL_INFILTRATION_RPRT_VU

1145.1 SQL Query for view SOIL_INFILTRATION_RPRT_VU

```
select Distinct as SchoolID, 'SF' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from SOIL_INFILTRATION
group by SchoolID
```

1146 View SOIL_INFILTRATION_VU1

1146.1 SQL Query for view SOIL_INFILTRATION_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'SF' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(SatWaterContent) AS SWCCnt,
 COUNT(UpperWaterLevel+LowerWaterLevel) AS WLevCnt,
 COUNT(InnerRingDiam+OuterRingDiam) AS RDiamCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.SOIL_INFILTRATION
where UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1147 View SOIL_INFILTRATION_VU2

1147.1 SQL Query for view SOIL_INFILTRATION_VU2

```
select SOIL_INFILTRATION_VU1.*, (2*wlevcnt+2*rdiamcnt+3*swccnt) as DixonCnt
from SOIL_INFILTRATION_VU1
```

1148 View SOIL_MOISTURE_RPRT_VU

1148.1 SQL Query for view SOIL_MOISTURE_RPRT_VU

```
select Distinct as SchoolID, 'SM' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from SOIL_MOISTURE
group by SchoolID
```

1149 View SOIL_MOISTURE_VU1

1149.1 SQL Query for view SOIL_MOISTURE_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'SM' AS MeasureCode,
 COUNT(*) AS RowCnt,
 ROUND(AVG(DECODE(Depth,5,1,null)*WaterContent),2) AS SM05Avg,
```

```
 COUNT(DECODE(Depth,5,1,null)*WaterContent) AS SM05Cnt,
 ROUND(AVG(DECODE(Depth,10,1,null)*WaterContent),2) AS SM10Avg,
 COUNT(DECODE(Depth,10,1,null)*WaterContent) AS SM10Cnt,
 ROUND(AVG(DECODE(Depth,30,1,null)*WaterContent),2) AS SM30Avg,
 COUNT(DECODE(Depth,30,1,null)*WaterContent) AS SM30Cnt,
 ROUND(AVG(DECODE(Depth,60,1,null)*WaterContent),2) AS SM60Avg,
 COUNT(DECODE(Depth,60,1,null)*WaterContent) AS SM60Cnt,
 ROUND(AVG(DECODE(Depth,90,1,null)*WaterContent),2) AS SM90Avg,
 COUNT(DECODE(Depth,90,1,null)*WaterContent) AS SM90Cnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.SOIL_MOISTURE
where UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1150 View SOIL_TEMP_DR1

1151 View SOIL_TEMP_DR2

1151.1 SQL Query for view SOIL_TEMP_DR2

```
select SOIL_TEMP_DR1.*, STCnt as DixonCnt
from SOIL_TEMP_DR1
```

1152 View SOIL_TEMP_DU1

1153 View SOIL_TEMP_DU2

1153.1 SQL Query for view SOIL_TEMP_DU2

```
select SOIL_TEMP_DU1.*, STCnt as DixonCnt
from SOIL_TEMP_DU1
```

1154 View SOIL_TEMP_HR0

1154.1 SQL Query for view SOIL_TEMP_HR0

```
select
 SchoolId,
 SiteId,
 SolarDay
from (select
       ST.SchoolId,
       ST.SiteId,
       TRUNC(SolarDayMeas) as SolarDay,
       Depth
      from
        GLOBE.SOIL_TEMP ST,
        GLOBE.SITE_LOCATION SL
      where
       ST.SchoolId = SL.SchoolId AND
       ST.SiteId = SL.SiteId AND
       SL.DataSource in (1,2) AND
       SolarDayMeas >= TO_DATE('20000601','YYYYMMDD') AND
       SoilTemp is not null
      group by ST.SchoolId, ST.SiteId, TRUNC(SolarDayMeas), Depth )
group by SchoolId, SiteId, SolarDay
having count(*) >= 2
```

1155 View SOIL_TEMP_HR1

1155.1 SQL Query for view SOIL_TEMP_HR1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDay, 'MM') as Month,
 Count(*) as TempDays
from
 (select * from SOIL_TEMP_HR0
  UNION
  select * from LOGGER_HR0)
group by
 SchoolId, SiteId, TRUNC(SolarDay, 'MM')
```

1156 View SOIL_TEMP_HR2

1156.1 SQL Query for view SOIL_TEMP_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.TempDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 SOIL_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.TempDays) >= 4*30*(.70)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.TempDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 SOIL_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.TempDays) >= 4*30*(.70)
```

1157 View SOIL_TEMP_MC1

1157.1 SQL Query for view SOIL_TEMP_MC1

```
select 'ST' as MeasureCode, COUNT(*) as RowCnt, COUNT(SoilTemp) as STCnt
from GLOBE.SOIL_TEMP
```

1158 View SOIL_TEMP_MC2

1158.1 SQL Query for view SOIL_TEMP_MC2

```
select SOIL_TEMP_MC1.*, STCnt as DixonCnt
from SOIL_TEMP_MC1
```

1159 View SOIL_TEMP_RPRT_VU

1159.1 SQL Query for view SOIL_TEMP_RPRT_VU

```
select Distinct as SchoolID, 'ST' as MeasureCode, MIN(UTTimeReported) as FirstReport,
```

```
MAX(UTTimeReported) as LastReport
from SOIL_TEMP
group by SchoolID
```

1160 View SOIL_TEMP_VU1

1160.1 SQL Query for view SOIL_TEMP_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(SolarDayMeas,'MM') AS Month,
 'ST' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT(SoilTemp) AS STCnt,
 Count(SoilTemp*DECODE(TempUsage,2,1,3,1,Null)) AS DSTCnt,
 ROUND(AVG(SoilTemp*DECODE(TempUsage,1,1,3,1,Null)*DECODE(Depth,5,1,Null)),1) AS AST5Avg,
 Count(SoilTemp*DECODE(TempUsage,1,1,3,1,Null)*DECODE(Depth,5,1,Null)) AS AST5Cnt,
 ROUND(AVG(SoilTemp*DECODE(TempUsage,1,1,3,1,Null)*DECODE(Depth,10,1,Null)),1) AS AST10Avg,
 Count(SoilTemp*DECODE(TempUsage,1,1,3,1,Null)*DECODE(Depth,10,1,Null)) AS AST10Cnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.SOIL_TEMP
where UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(SolarDayMeas,'MM')
```

1161 View SOIL_TEMP_VU2

1161.1 SQL Query for view SOIL_TEMP_VU2

```
select SOIL_TEMP_VU1.*, STCnt as DixonCnt
from SOIL_TEMP_VU1
```

1162 View SURFACEWATER_SITE_DR1

1163 View SURFACEWATER_SITE_DR2

1163.1 SQL Query for view SURFACEWATER_SITE_DR2

```
select SURFACEWATER_SITE_DR1.*,
(sorfcnt+mwtypcnt+mwwidcnt+swtypcnt+swsizcnt+swareacnt+swdepcnt+wloctypcnt+turbecnt+botfcnt+bankcnt+be
drcnt+okitcnt+omodcnt+akitcnt+amodcnt+nkitcnt+nmodcnt+skitcnt+smodcnt) as DixonCnt
from SURFACEWATER_SITE_DR1
```

1164 View SURFACEWATER_SITE_DU1

1165 View SURFACEWATER_SITE_DU2

1165.1 SQL Query for view SURFACEWATER_SITE_DU2

```
select SURFACEWATER_SITE_DU1.*,
(sorfcnt+mwtypcnt+mwwidcnt+swtypcnt+swsizcnt+swareacnt+swdepcnt+wloctypcnt+turbecnt+botfcnt+bankcnt+be
drcnt+okitcnt+omodcnt+akitcnt+amodcnt+nkitcnt+nmodcnt+skitcnt+smodcnt) as DixonCnt
from SURFACEWATER_SITE_DU1
```

1166 View SURFACEWATER_SITE_MC1

1166.1 SQL Query for view SURFACEWATER_SITE_MC1

```
select 'MD' as MeasureCode, SUM(DECODE(SaltOrFresh,0,0,1)) as SorFCnt,
SUM(DECODE(MovingWaterType,0,0,1)) as MWTypCnt, COUNT(MovingWaterWidth) as MWWidCnt,
SUM(DECODE(StandWaterType,0,0,1)) as SWTypCnt, SUM(DECODE(StandWaterSize,0,0,1)) as SWSizCnt,
COUNT(StandWaterArea) as SWAreaCnt, COUNT(StandWaterDepth) as SWDepCnt,
SUM(DECODE(WaterLocationType,0,0,1)) as WLocTypCnt, SUM(DECODE(TurbidityEst,0,0,1)) as TurbECnt,
```

```
SUM(DECODE(SeeBottomFlag,'0',0,1)) as BotFCnt, SUM(DECODE(BankMaterial,0,0,1)) as BankCnt,
SUM(DECODE(BedRockType,0,0,1)) as BedRCnt, COUNT(OxygenKitMfg) as OKitCnt, COUNT(OxygenKitModel) as
OModCnt, COUNT(AlkalinityKitMfg) as AKitCnt, COUNT(AlkalinityKitModel) as AModCnt,
COUNT(NitrateKitMfg) as NKitCnt, COUNT(NitrateKitModel) as NModCnt, COUNT(SalinityKitMfg) as SKitCnt,
COUNT(SalinityKitModel) as SModCnt
from GLOBE.SURFACEWATER_SITE
```

1167 View SURFACEWATER_SITE_MC2

1167.1 SQL Query for view SURFACEWATER_SITE_MC2

```
select SURFACEWATER_SITE_MC1.*,
(sorfcnt+mwtypcnt+mwwidcnt+swtypcnt+swsizcnt+swareacnt+swdepcnt+wloctypcnt+turbecnt+botfcnt+bankcnt+be
drcnt+okitcnt+omodcnt+akitcnt+amodcnt+nkitcnt+nmodcnt+skitcnt+smodcnt) as DixonCnt
from SURFACEWATER_SITE_MC1
```

1168 View SURFACEWATER_SITE_RPRT_VU

1168.1 SQL Query for view SURFACEWATER_SITE_RPRT_VU

```
select Distinct as SchoolID, 'SW' as MeasureCode, MIN(UTTimeStamp) as FirstReport, MAX(UTTimeStamp) as
LastReport
from SURFACEWATER_SITE
group by SchoolID
```

1169 View SURFACEWATER_SITE_VU1

1169.1 SQL Query for view SURFACEWATER_SITE_VU1

```
select
 Schoolid,
 Siteid,
 TRUNC(VersionDate,'MM') AS Month,
 'MD' AS MeasureCode, COUNT(*) AS RowCnt,
 SUM(DECODE(SaltOrFresh,0,0,1)) AS SorFCnt,
 SUM(DECODE(MovingWaterType,0,0,1)) AS MWTypCnt,
 COUNT(MovingWaterWidth) AS MWWidCnt,
 SUM(DECODE(StandWaterType,0,0,1)) AS SWTypCnt,
 SUM(DECODE(StandWaterSize,0,0,1)) AS SWSizCnt,
 COUNT(StandWaterArea) AS SWAreaCnt,
 COUNT(StandWaterDepth) AS SWDepCnt,
 SUM(DECODE(WaterLocationType,0,0,1)) AS WLocTypCnt,
 SUM(DECODE(TurbidityEst,0,0,1)) AS TurbECnt,
 SUM(DECODE(SeeBottomFlag,'0',0,1)) AS BotFCnt,
 SUM(DECODE(BankMaterial,0,0,1)) AS BankCnt,
 SUM(DECODE(BedRockType,0,0,1)) AS BedRCnt,
 COUNT(OxygenKitMfg) AS OKitCnt,
 COUNT(OxygenKitModel) AS OModCnt,
 COUNT(AlkalinityKitMfg) AS AKitCnt,
 COUNT(AlkalinityKitModel) AS AModCnt,
 COUNT(NitrateKitMfg) AS NKitCnt,
 COUNT(NitrateKitModel) AS NModCnt,
 COUNT(SalinityKitMfg) AS SKitCnt,
 COUNT(SalinityKitModel) AS SModCnt,
 MIN(VersionDate) AS BegDate,
 MAX(VersionDate) AS EndDate,
 MIN(UTtimestamp) AS BegRDate,
 MAX(UTtimestamp) AS EndRDate,
 ROUND(AVG(UTtimestamp-VersionDate),2) ASLagTimeAvg
from SURFACEWATER_SITE
where UTtimeStamp < TRUNC (SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(VersionDate,'MM')
```

1170 View SURFACEWATER_SITE_VU2

1170.1 SQL Query for view SURFACEWATER_SITE_VU2

```
select SURFACEWATER_SITE_VU1.*,
(sorfcnt+mwtypcnt+mwwidcnt+swtypcnt+swsizcnt+swareacnt+swdepcnt+wloctypcnt+turbecnt+botfcnt+bankcnt+be
drcnt+okitcnt+omodcnt+akitcnt+amodcnt+nkitcnt+nmodcnt+skitcnt+smodcnt) as DixonCnt
```

```
from SURFACEWATER_SITE_VU1
```

1171 View SURFACE_WATER_DR1

1171.1 SQL Query for view SURFACE_WATER_DR1

```
select TRUNC(UTtimeReported,'DD') as Day, 'SW' as MeasureCode, COUNT(*) as RowCnt, COUNT(WaterTemp) as
WTempCnt, Count(Acidity) as WpHCnt, COUNT(Conductivity) as ConCnt, COUNT(Alkalinity) as AlkCnt,
COUNT(DissolvedOxy) as OxyCnt, COUNT(HydAvgSalinity) as HydAvgSalCnt, COUNT(TitrSalinity) as
TitrSalCnt, COUNT(NitriteNitrate) as NitNitCnt, COUNT(NitriteOnly) as NitOnlyCnt,
COUNT(DECODE(CloudCover,9,NULL,1)) as CloudCnt, COUNT(BeforeTideHour + AfterTideHour) as TideCnt,
COUNT((Nitritenitrate - NitriteOnly)) AS NO3CntFROM GLOBE.SURFACE_WATERWHERE UTtimeReported < TRUNC
(SYSDATE,'DD') + 22/24GROUP BY TRUNC as (UTtimeReported,'DD')
```

1172 View SURFACE_WATER_DR2

1172.1 SQL Query for view SURFACE_WATER_DR2

```
select SURFACE_WATER_DR1.*,
(3*wtempcnt+3*wphcnt+3*concnt+3*alkcnt+3*oxycnt+6*hydavgsalcnt+3*titrsalcnt+3*nitnitcnt+3*nitonlycnt+c
loudcnt+2*tidecnt) as DixonCnt
from SURFACE_WATER_DR1
```

1173 View SURFACE_WATER_DU1

1173.1 SQL Query for view SURFACE_WATER_DU1

```
select TRUNC(UTtimeMeasured,'DD') as Day, 'SW' as MeasureCode, COUNT(*) as RowCnt, COUNT(WaterTemp) as
WTempCnt, Count(Acidity) as WpHCnt, COUNT(Conductivity) as ConCnt, COUNT(Alkalinity) as AlkCnt,
COUNT(DissolvedOxy) as OxyCnt, COUNT(HydAvgSalinity) as HydAvgSalCnt, COUNT(TitrSalinity) as
TitrSalCnt, COUNT(NitriteNitrate) as NitNitCnt, COUNT(NitriteOnly) as NitOnlyCnt,
COUNT(DECODE(CloudCover,9,NULL,1)) as CloudCnt, COUNT(BeforeTideHour + AfterTideHour) as TideCnt,
COUNT((Nitritenitrate - NitriteOnly)) AS NO3CntFROM GLOBE.SURFACE_WATERWHERE UTtimeReported < TRUNC
(SYSDATE,'DD') + 22/24GROUP BY TRUNC as (UTtimeMeasured,'DD')
```

1174 View SURFACE_WATER_DU2

1174.1 SQL Query for view SURFACE_WATER_DU2

```
select SURFACE_WATER_DU1.*,
(3*wtempcnt+3*wphcnt+3*concnt+3*alkcnt+3*oxycnt+6*hydavgsalcnt+3*titrsalcnt+3*nitnitcnt+3*nitonlycnt+c
loudcnt+2*tidecnt) as DixonCnt
from SURFACE_WATER_DU1
```

1175 View SURFACE_WATER_HR1

1175.1 SQL Query for view SURFACE_WATER_HR1

```
select SchoolId, SiteId, TRUNC(DayMeasured,'WW') AS Week, 'SW' AS MeasureCode,
DECODE(MIN(SourceState),0,Null,1) AS BadSource,
COUNT(*) AS RowCnt, COUNT(WaterTemp) AS WTempCnt, Count(Acidity) AS WpHCnt,
COUNT(TurbiditySecchi) AS SecCnt,
COUNT(TurbidityTube) AS TubeCnt,
COUNT(Conductivity) AS ConCnt, COUNT(Alkalinity) AS AlkCnt,
COUNT(DissolvedOxy) AS OxyCnt, COUNT(HydAvgSalinity) AS HydAvgSalCnt,
COUNT(TitrSalinity) AS TitrSalCnt,
COUNT((Nitritenitrate - NitriteOnly)) AS NO3Cnt
from GLOBE.SURFACE_WATER
where DayMeasured >= TO_DATE('1999 02 01','YYYY MM DD')

group by SchoolId, SiteId, TRUNC(DayMeasured,'WW')
Having TRAnd DayMeasured < TRUNC(SYSDATE,'MM') + 6UNC(DayMeasured,'WW') >= TO_DATE('1999 02 01','YYYY
MM DD')
AND TRUNC(DayMeasured,'WW') < TRUNC(SYSDATE,'MM')
```

1176 View SURFACE_WATER_HR2

1176.1 SQL Query for view SURFACE_WATER_HR2

```
select SchoolId, SiteId, Week, BadSource, RowCnt, WTempCnt, WpHCnt, DECODE(SecCnt+TubeCnt,0,0,1) as
Trans, DECODE(OxyCnt,0,0,1) as Oxy, DECODE(AlkCnt,0,0,1) as Alk, DECODE(NO3Cnt,0,0,1) as Nit,
DECODE(HydAvgSalCnt+TitrSalCnt+ConCnt,0,0,1) as Sal, DECODE(SecCnt+TubeCnt,0,0,1) +DECODE
(OxyCnt,0,0,1) +DECODE (AlkCnt,0,0,1) +DECODE (NO3Cnt,0,0,1) +DECODE
(HydAvgSalCnt+TitrSalCnt+ConCnt,0,0,1) as SumCnt
from SURFACE_WATER_HR1
```

1177 View SURFACE_WATER_HR3

1177.1 SQL Query for view SURFACE_WATER_HR3

```
select BS.SchoolId, BS.SiteId, BS.Month, NVL(GS.GoodSourceCnt,0) as GScnt, BS.BadSourceCnt as BScnt,
DX.DixonCnt
from SURFACE_WATER_HRGS GS, SURFACE_WATER_HRBS BS, SURFACE_WATER_HRDX DX
where BS.SchoolId = GS.SchoolId  (
+
)  AND  BS.SiteId  =  GS.SiteId  (
+
)  AND  BS.Month  =  GS.Month  (
+
)  AND  BS.SchoolId  =  DX.SchoolId  AND  GS.SiteId  =  DX.SiteId  AND  GS.Month  =  DX.Month
```

1178 View SURFACE_WATER_HR4

1178.1 SQL Query for view SURFACE_WATER_HR4

```
select
2*TRUNC(Round(MONTHS_BETWEEN(Month,TO_DATE('19990201', 'YYYYMMDD'))/4,1))+1 as HRnbr,
SchoolId, SiteId,
MIN(Month) as BeginMonth,
MAX(Month) as EndMonth,
SUM(GScnt) as GScnt, SUM(BScnt) as BScnt, SUM(DixonCnt) as DixonCnt
from SURFACE_WATER_HR3
GROUP BY SchoolId, SiteId, 2*TRUNC(Round(MONTHS_BETWEEN(Month,TO_DATE('19990201', 'YYYYMMDD'))/4,1
))+1
HAVING MIN(Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM') AND SUM(GScnt) > 7
UNION ALL
select
2*TRUNC(Round((MONTHS_BETWEEN(Month,TO_DATE('19990201', 'YYYYMMDD'))+2)/4,1)) as HRnbr,
SchoolId, SiteId,
MIN(Month) as BeginMonth,
MAX(Month) as EndMonth,
SUM(GScnt) as GScnt, SUM(BScnt) as BScnt, SUM(DixonCnt) as DixonCnt
from SURFACE_WATER_HR3
GROUP BY SchoolId, SiteId, 2*TRUNC(Round((MONTHS_BETWEEN(Month,TO_DATE('19990201', 'YYYYMMDD'))+2)
/4,1))
HAVING Min(Month) > TO_DATE('19990201', 'YYYYMMDD')
AND MIN(Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
AND SUM(GScnt) > 7
```

1179 View SURFACE_WATER_HR5

1179.1 SQL Query for view SURFACE_WATER_HR5

```
select SURFACE_WATER_HR4.*, DECODE(SITE_LOCATION.DataSource,0,'Unchecked',1,'GPS',2,'Other') as
LocMethod
from SURFACE_WATER_HR4, GLOBE.SITE_LOCATION
where SURFACE_WATER_HR4.SchoolId = SITE_LOCATION.SCHOOLID  AND  SURFACE_WATER_HR4.SiteId  =
SITE_LOCATION.SITEID
```

1180 View SURFACE_WATER_HR6

1180.1 SQL Query for view SURFACE_WATER_HR6

```
select HRnbr, SchoolId, SUM(DixonCnt) as DixonCnt,
TRUNC(TO_DATE('19990201','YYYYMMDD')+2*(HRnbr-1)*31,'MM') as BeginMonth,
TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM') as EndMonth
from SURFACE_WATER_HR5
where LocMethod = 'GPS'
or LocMethod = 'Other'
group by HRnbr, SchoolId,
TRUNC(TO_DATE('19990201','YYYYMMDD')+2*(HRnbr-1)*31,'MM'),
TRUNC(TO_DATE('19990201','YYYYMMDD')+(2*(HRnbr+1)-1)*31,'MM')
```

1181 View SURFACE_WATER_HRBS

1181.1 SQL Query for view SURFACE_WATER_HRBS

```
select
 SchoolId,
 SiteId,
 TRUNC(Week,'MM') as Month,
 COUNT(BadSource) AS BadSourceCnt
from SURFACE_WATER_HR2
group by SchoolId, SiteId, TRUNC(Week,'MM')
```

1182 View SURFACE_WATER_HRDX

1182.1 SQL Query for view SURFACE_WATER_HRDX

```
select SchoolId, SiteId, Month, DixonCnt
from GLOBE.SURFACE_WATER_CNT
where Month > =  TO_DATE  (
'19990201'
,
'YYYYMMDD'
)  AND  Month  <  TRUNC  (
SYSDATE
,
'MM'
)
```

1183 View SURFACE_WATER_HRGS

1183.1 SQL Query for view SURFACE_WATER_HRGS

```
select
 SchoolId,
 SiteId,
 TRUNC(Week,'MM') AS Month,
 COUNT(*) AS GoodSourceCnt
from
 SURFACE_WATER_HR2
where
 WTempCnt > 0 AND
 WpHCnt > 0 AND
 SumCnt >= 2
group by SchoolId, SiteId, TRUNC(Week,'MM')
```

1184 View SURFACE_WATER_MC1

1184.1 SQL Query for view SURFACE_WATER_MC1

```
select 'SW' as MeasureCode, COUNT(*) as RowCnt, COUNT(WaterTemp) as WTempCnt, Count(Acidity) as
WpHCnt, COUNT(Conductivity) as ConCnt, COUNT(Alkalinity) as AlkCnt, COUNT(DissolvedOxy) as OxyCnt,
COUNT(HydAvgSalinity) as HydAvgSalCnt, COUNT(TitrSalinity) as TitrSalCnt, COUNT(NitriteNitrate) as
NitNitCnt, COUNT(NitriteOnly) as NitOnlyCnt, COUNT(DECODE(CloudCover,9,NULL,1)) as CloudCnt,
COUNT(BeforeTideHour + AfterTideHour) as TideCnt, COUNT((Nitritenitrate - NitriteOnly)) AS NO3CntFROM
as GLOBE.SURFACE_WATER
```

1185 View SURFACE_WATER_MC2

1185.1 SQL Query for view SURFACE_WATER_MC2

```
select SURFACE_WATER_MC1.*,
(3*wtempcnt+3*wphcnt+3*concnt+3*alkcnt+3*oxycnt+6*hydavgsalcnt+3*titrsalcnt+3*nitnitcnt+3*nitonlycnt+c
loudcnt+2*tidecnt) as DixonCnt
from SURFACE_WATER_MC1
```

1186 View SURFACE_WATER_RPRT_VU

1186.1 SQL Query for view SURFACE_WATER_RPRT_VU

```
select Distinct as SchoolID, 'SW' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from SURFACE_WATER
group by SchoolID
```

1187 View SURFACE_WATER_VU1

1187.1 SQL Query for view SURFACE_WATER_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'SW' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(WaterTemp) AS WTempCnt,
 Count(Acidity) AS WpHCnt,
 COUNT(Conductivity) AS ConCnt,
 COUNT(Alkalinity) AS AlkCnt,
 COUNT(DissolvedOxy) AS OxyCnt,
 COUNT(HydAvgSalinity) AS HydAvgSalCnt,
 COUNT(TitrSalinity) AS TitrSalCnt,
 COUNT(NitriteNitrate) AS NitNitCnt,
 COUNT(NitriteOnly) AS NitOnlyCnt,
 COUNT(DECODE(CloudCover,9,NULL,1)) AS CloudCnt,
 COUNT(BeforeTideHour + AfterTideHour) AS TideCnt,
 ROUND(AVG(WaterTemp),1) AS WaterTAvg,
 MIN(WaterTemp) as WTempMin,
 MAX(WaterTemp) as WTempMax,
 ROUND(STDDEV(WaterTemp),1) as WTempStd,
 ROUND(AVG(Acidity),1) AS WpHAvg,
 MIN(Acidity) as WpHMin,
 MAX(Acidity) as WpHMax,
 ROUND(STDDEV(Acidity),1) as WpHStd,
 ROUND(AVG(Conductivity)) AS ConAvg,
 ROUND(AVG(Alkalinity)) AS AlkAvg,
 ROUND(AVG(DissolvedOxy),1) AS OxyAvg,
 ROUND(AVG(HydAvgSalinity),1) AS HASalAvg,
 ROUND(AVG(TitrSalinity),1) AS TSalAvg,
 ROUND(AVG(TurbiditySecchi),2) AS SecchiAvg,
 ROUND(AVG(TurbidityTube)) AS TubeAvg,
 ROUND(AVG((Nitritenitrate - NitriteOnly)),1) AS NO3Avg,
 COUNT((Nitritenitrate - NitriteOnly)) AS NO3Cnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported - UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.SURFACE_WATER
where UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1188 View SURFACE_WATER_VU2

1188.1 SQL Query for view SURFACE_WATER_VU2

```
select SURFACE_WATER_VU1.*, (3*wtempcnt+3*wphcnt+3*concnt+3*alkcnt+3*oxycnt+
6*hydavgsalcnt+3*titrsalcnt+3*nitnitcnt+3*nitonlycnt+  cloudcnt+2*tidecnt) as DixonCnt
from SURFACE_WATER_VU1
```

1189 View SoilNPK_HR1

1189.1 SQL Query for view SoilNPK_HR1

```
select
 SchoolId,
 SiteId,
 MAX(Day) as Day
from
    (select
     SC.SchoolId,
     SC.SiteId,
     SC.HorizonNbr,
     TRUNC(SC.UTtimeMeasured) as Day
    from
     GLOBE.SOIL_CHAR SC,
     GLOBE.SITE_LOCATION SL
    where
     SC.SchoolId = SL.SchoolId AND
     SC.SiteId = SL.SiteId AND
     SC.NitrogenEst <> 0 AND
     SC.PhosphorusEst <> 0 AND
     SC.PotassiumEst <> 0 AND
     SL.DataSource in (1,2) AND
     SC.UTtimeMeasured >= TO_DATE('20000601','YYYYMMDD'))
group by SchoolId, SiteId
having count(*) > 4
```

1190 View SoilNPK_HR2

1190.1 SQL Query for view SoilNPK_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId
from SoilNPK_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId
from SoilNPK_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId
from SoilNPK_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
```

```
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId
from SoilNPK_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId
from SoilNPK_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
UNION ALL
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId
from SoilNPK_HR1 A
group by A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
having
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1191 View TEMPERATURE_LOGGER_VU1

1191.1 SQL Query for view TEMPERATURE_LOGGER_VU1

```
select
  SchoolId,
  SiteId,
  TRUNC(DDdate,'MM') as Month,
  'TL' as MeasureCode,
  SUM(RowCnt) RowCnt,
  SUM(DECODE(ATCnt,1,3,0,1,NULL,0,3)) ATCnt,
  SUM(DECODE(ST05Cnt,1,3,0,1,NULL,0,3)) ST05Cnt,
  SUM(DECODE(ST10Cnt,1,3,0,1,NULL,0,3)) ST10Cnt,
  SUM(DECODE(ST50Cnt,1,3,0,1,NULL,0,3)) ST50Cnt,
  ROUND(AVG(ATAvg),1) ATAvg,
  SUM(ATACnt) ATACnt,
  ROUND(AVG(ST05Avg),1) ST05Avg,
  SUM(ST05ACnt) ST05ACnt,
  ROUND(AVG(ST10Avg),1) ST10Avg,
  SUM(ST10ACnt) ST10ACnt,
  ROUND(AVG(ST50Avg),1) ST50Avg,
  SUM(ST50ACnt) ST50ACnt,
  MIN(BegDate) BegDate,
  MAX(EndDate) EndDate,
```

```
  MIN(BegRDate) BegRDate,
  MAX(EndRDate) EndRDate,
  ROUND(AVG(LagTimeAvg),2) LagTimeAvg
from (
  select schoolid,siteid,TRUNC(HHdate,'DD') DDdate,
    Sum(RowCnt) RowCnt,
    TRUNC(DECODE(SUM(ATCnt),0,NULL,SUM(ATCnt))/(1+23*4)) ATCnt,
    TRUNC(DECODE(SUM(ST05Cnt),0,NULL,SUM(ST05Cnt))/24) ST05Cnt,
    TRUNC(DECODE(SUM(ST10Cnt),0,NULL,SUM(ST10Cnt))/24) ST10Cnt,
    TRUNC(DECODE(SUM(ST50Cnt),0,NULL,SUM(ST50Cnt))/24) ST50Cnt,
    AVG(ATAvg) ATAvg,
    SUM(ATACnt) ATACnt,
    AVG(ST05Avg) ST05Avg,
    SUM(ST05ACnt) ST05ACnt,
    AVG(ST10Avg) ST10Avg,
    SUM(ST10ACnt) ST10ACnt,
    AVG(ST50Avg) ST50Avg,
    SUM(ST50ACnt) ST50ACnt,
    MIN(BegDate) AS BegDate,
    MAX(EndDate) AS EndDate,
    MIN(BegRDate) AS BegRDate,
    MAX(EndRDate) AS EndRDate,
    AVG(LagTimeAvg) AS LagTimeAvg
  from (
    select schoolid,siteid,TRUNC(UTtimemeasured,'HH24') HHdate,
      COUNT(*) RowCnt,
      DECODE(COUNT(CurrentTemp),NULL,0,COUNT(CurrentTemp)) ATCnt,
      DECODE(COUNT(SoilTemp5cm),0,0,NULL,0,1) ST05Cnt,
      DECODE(COUNT(SoilTemp10cm),0,0,NULL,0,1) ST10Cnt,
      DECODE(COUNT(SoilTemp50cm),0,0,NULL,0,1) ST50Cnt,
      AVG(CurrentTemp) ATAvg,
      COUNT(CurrentTemp) ATACnt,
      AVG(SoilTemp5cm) ST05Avg,
      COUNT(SoilTemp5cm) ST05ACnt,
      AVG(SoilTemp10cm) ST10Avg,
      COUNT(SoilTemp10cm) ST10ACnt,
      AVG(SoilTemp50cm) ST50Avg,
      COUNT(SoilTemp50cm) ST50ACnt,
      MIN(UTtimemeasured) AS BegDate,
      MAX(UTtimemeasured) AS EndDate,
      MIN(UTTimereported) AS BegRDate,
      MAX(UTtimereported) AS EndRDate,
      AVG(UTtimereported-UTtimemeasured) AS LagTimeAvg
    from GLOBE.TEMPERATURE_LOGGER
    where UTtimereported < TRUNC(SYSDATE,'DD') + 22/24
    group by SchoolId, SiteId, TRUNC(UTtimemeasured,'HH24'))
  group by SchoolId, SiteId, TRUNC(HHdate,'DD'))
group by SchoolId, SiteId, TRUNC(DDdate,'MM')
```

1192 View TEMPERATURE_LOGGER_VU2

1192.1 SQL Query for view TEMPERATURE_LOGGER_VU2

```
select
  TEMPERATURE_LOGGER_VU1.*,
  (atcnt+st05cnt+st10cnt+st50cnt) as dixoncnt
from TEMPERATURE_LOGGER_VU1
```

1193 View TMI_ActiveSchool250DixonCnt

1193.1 SQL Query for view TMI_ActiveSchool250DixonCnt

```
select
 S.SchoolId,
 S.LifeCycleCode,
 D.DixonCnt,
 L.Latitude,
 L.Longitude,
 L.DataSource,
 S.SchoolName,
 S.AddressLine1,
 S.AddressLine2,
 S.City,
 S.StateProvCode,
 S.PostalCode,
```

```
 S.CountryCode,
 C.GlobeName as CountryName,
 S.DirectorFullName
from
 GLOBE.GLOBE_SCHOOL S,
 GLOBE.GLOBE_COUNTRY C,
 GLOBE.DIXON_SUMMARY_CNT D,
 TMI_SchoolLocationviaGPS L
where
 S.SchoolId = D.SchoolId AND
 S.CountryCode = C.CountryCode AND
 S.SchoolId = L.SchoolId(+) AND
 S.LifeCycleCode = 'A' AND
 D.DixonCnt >= 250
```

1194 View TMI_CurrentGoodImage

1194.1 SQL Query for view TMI_CurrentGoodImage

```
select
 SchoolId,
 MAX(RequestDate) as MaxRequestDate,
 MAX(ShipDate) as MaxShipDate,
 Count(*) as NbrRequests
from
 GLOBE.TM_IMAGE
where
 ShipDate IS NOT NULL
group by
 SchoolId
```

1195 View TMI_ImageEligibleSchools

1195.1 SQL Query for view TMI_ImageEligibleSchools

```
select
 SchoolId,
 DixonCnt,
 Latitude,
 Longitude,
 DataSource as LocCode,
 TRUNC(SYSDATE) as RequestDate,
 NULL as PathNbr,
 NULL as RowNbr,
 NULL as ImageDate,
 NULL as ShipDate,
 NULL as Datasource,
 NULL as Comments,
 SchoolName,
 AddressLine1 as AddrLine1,
 AddressLine2 as AddrLine2,
 City,
 StateProvCode,
 PostalCode,
 CountryCode,
 CountryName,
 DirectorFullName,
 GLOBE.ListTeachers(SchoolId) as TeacherList
from
 TMI_PotentiallyEligibleSchools
where
 Latitude IS NOT NULL
 AND
 GLOBE.ListTeachers(SchoolId) IS NOT NULL
```

1196 View TMI_PotentiallyEligibleSchools

1196.1 SQL Query for view TMI_PotentiallyEligibleSchools

```
select *
from
  (select
     S.*,
```

```
     I.SchoolId as HasCurrentImage
   from
     TMI_ActiveSchool250DixonCnt S,
     TMI_CurrentGoodImage I
   where
     S.SchoolId = I.SchoolId(+) )
where
 HasCurrentImage IS NULL
```

1197 View TMI_SchoolLocationViaGPS

1197.1 SQL Query for view TMI_SchoolLocationViaGPS

```
select
 SchoolId,
 SiteId,
 Latitude,
 Longitude,
 Elevation,
 DataSource,
 UTtimestamp
from
 GLOBE.SITE_LOCATION
where
  SiteId='SCH-01' AND
  SchoolId in (select distinct SchoolId
               from GLOBE.SITE_LOCATION
               where DataSource = 1)
```

1198 View TREEBIOM_HR1

1198.1 SQL Query for view TREEBIOM_HR1

```
select
 A.SchoolId,
 A.SiteId,
 TRUNC(A.DayMeasured) AS Day
from
 globe.tree_biometry A,
 globe.treebiom_ws B,
 globe.land_cover C
where
 A.SchoolId = B.SchoolId AND A.SiteId = B.SiteId AND A.DayMeasured = B.DayMeasured AND
 A.SchoolId = C.SchoolId AND A.SiteId = C.SiteId AND
 (C.MUCCode LIKE '0%' OR C.MUCCode LIKE '1%') AND
 A.DayMeasured >= TO_DATE('19980601','YYYYMMDD') AND
 A.DomGenus IS NOT NULL AND
 A.DomSpecies IS NOT NULL AND
 A.CanopyPlus IS NOT NULL AND
 A.CanopyMinus IS NOT NULL AND
group by A.Schoolid, A.SiteId, TRUNC(A.Daymeasured)
having (COUNT(DomHeight) + COUNT(CodomHeight)) >= 10 AND
 (COUNT(DomCircum) + COUNT(CodomCircum)) >= 10
```

1199 View TREEBIOM_HR1_LocP

1199.1 SQL Query for view TREEBIOM_HR1_LocP

```
select
 A.SchoolId,
 A.SiteId,
 A.Day,
 DECODE(B.DataSource,0,'Unchecked',1,'GPS',2,'Other') as LocMethod,
 Count(C.PhotoId) as Photos
from
 TREEBIOM_HR1 A,
 GLOBE.SITE_LOCATION B,
 GLOBE.SITE_PHOTO C
where
```

```
 A.SchoolId = B.SchoolId AND A.SchoolId = C.SchoolId AND
 A.SiteId = B.SiteId AND A.SiteId = C.SiteId
group by A.SchoolId, A.SiteId, A.Day,
DECODE(B.DataSource,0,'Unchecked',1,'GPS',2,'Other')
```

1200 View TREEBIOM_HR2

1200.1 SQL Query for view TREEBIOM_HR2

```
select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From TREEBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*0)/12)+0+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From TREEBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*1)/12)+1+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From TREEBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*2)/12)+2+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From TREEBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
```

```
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*3)/12)+3+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From TREEBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*4)/12)+4+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
 UNION ALL
 select
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) as HRnbr,
 A.SchoolId, A.SiteId,
 MIN(TRUNC(Day,'MM')) as BeginMonth,
 MAX(TRUNC(Day,'MM')) as EndMonth
 From TREEBIOM_HR1_LocP A
 Where A.LocMethod In ('GPS','Other')
 AND A.Photos >= 4
 Group By A.SchoolId, A.SiteId,
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1)
 HAVING
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) >= 1 AND
 (6*FLOOR((MONTHS_BETWEEN(TRUNC(Day,'MM'),
 TO_DATE('1998 06 01','YYYY MM DD'))-2*5)/12)+5+1) <=
 TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
 TO_DATE('1999 04 01','YYYY MM DD')))/2)
```

1201 View TREEBIOM_HR3

1201.1 SQL Query for view TREEBIOM_HR3

```
(select
 Hrnbr, SchoolId
 From treebiom_hr2
 Group by Hrnbr, SchoolId
 Having COUNT(SchoolId) >= 2
)
```

1202 View TREEBIOM_WS_DR1

1203 View TREEBIOM_WS_DR2

1203.1 SQL Query for view TREEBIOM_WS_DR2

```
select TREEBIOM_WS_DR1.*, (dhtcnt+cdhtcnt+dcmcnt+cdcmcnt) as DixonCnt
from TREEBIOM_WS_DR1
```

1204 View TREEBIOM_WS_DU1

1205 View TREEBIOM_WS_DU2

1205.1 SQL Query for view TREEBIOM_WS_DU2

```
select TREEBIOM_WS_DU1.*, (dhtcnt+cdhtcnt+dcmcnt+cdcmcnt) as DixonCnt
from TREEBIOM_WS_DU1
```

1206 View TREEBIOM_WS_MC1

1206.1 SQL Query for view TREEBIOM_WS_MC1

```
select 'TB' as MeasureCode, COUNT(*) as RowCnt, COUNT(DomHeight) as DHtCnt, COUNT(DomCircum) as
DCmCnt, COUNT(CodomHeight) as CdHtCnt, COUNT(CodomCircum) as CdCmCnt
from GLOBE.TREEBIOM_WS
```

1207 View TREEBIOM_WS_MC2

1207.1 SQL Query for view TREEBIOM_WS_MC2

```
select TREEBIOM_WS_MC1.*, (dhtcnt+cdhtcnt+dcmcnt+cdcmcnt) as DixonCnt
from TREEBIOM_WS_MC1
```

1208 View TREEBIOM_WS_RPRT_VU

1208.1 SQL Query for view TREEBIOM_WS_RPRT_VU

```
select Distinct as SchoolID, 'TB' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from TREEBIOM_WS
group by SchoolID
```

1209 View TREEBIOM_WS_VU1

1209.1 SQL Query for view TREEBIOM_WS_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'TB' AS MeasureCode,
 COUNT(*) AS RowCnt,
 COUNT(DomHeight) AS DHtCnt,
 COUNT(DomCircum) AS DCmCnt,
 COUNT(CodomHeight) AS CdHtCnt,
 COUNT(CodomCircum) AS CdCmCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
from GLOBE.TREEBIOM_WS
where UTtimeReported < TRUNC ( SYSDATE , 'DD' ) + 22/24
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1210 View TREEBIOM_WS_VU2

1210.1 SQL Query for view TREEBIOM_WS_VU2

```
select TREEBIOM_WS_VU1.*, (dhtcnt+cdhtcnt+dcmcnt+cdcmcnt) as DixonCnt
from TREEBIOM_WS_VU1
```

1211 View TREEGRASSBIOM_HR4

1211.1 SQL Query for view TREEGRASSBIOM_HR4

```
select distinct a.Hrnbr, a.SchoolId
from treebiom_hr2 a, grassbiom_hr2 b
where a.hrnbr = b.hrnbr  AND  a.schoolid  =  b.schoolid
```

1212 View TREE_BIOMETRY_DR1

1212.1 SQL Query for view TREE_BIOMETRY_DR1

```
select TRUNC(UTtimeReported,'DD') as Day, 'TB' as MeasureCode, COUNT(*) as RowCnt,
SUM(DECODE(NVL(DomGenus,'0'),'0',0,1)*DECODE(NVL(DomSpecies,'0'),'0',0,1)) as DGSpCnt,
SUM(DECODE(NVL(CodomGenus,'0'),'0',0,1)*DECODE(NVL(CodomSpecies,'0'),'0',0,1)) as CdGSpCnt,
SUM(NVL(CanopyPlus,0)) +SUM (NVL(CanopyMinus,0)) AS CnyCntFROM GLOBE.TREE_BIOMETRYWHERE UTtimeReported
< TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as (UTtimeReported,'DD')
```

1213 View TREE_BIOMETRY_DR2

1213.1 SQL Query for view TREE_BIOMETRY_DR2

```
select TREE_BIOMETRY_DR1.*, (cnycnt+dgspcnt+cdgspcnt) as DixonCnt
from TREE_BIOMETRY_DR1
```

1214 View TREE_BIOMETRY_DU1

1214.1 SQL Query for view TREE_BIOMETRY_DU1

```
select TRUNC(UTtimeMeasured,'DD') as Day, 'TB' as MeasureCode, COUNT(*) as RowCnt,
SUM(DECODE(NVL(DomGenus,'0'),'0',0,1)*DECODE(NVL(DomSpecies,'0'),'0',0,1)) as DGSpCnt,
SUM(DECODE(NVL(CodomGenus,'0'),'0',0,1)*DECODE(NVL(CodomSpecies,'0'),'0',0,1)) as CdGSpCnt,
SUM(NVL(CanopyPlus,0)) +SUM (NVL(CanopyMinus,0)) AS CnyCntFROM GLOBE.TREE_BIOMETRYWHERE UTtimeReported
< TRUNC (SYSDATE,'DD') + 22/24GROUP BY TRUNC as (UTtimeMeasured,'DD')
```

1215 View TREE_BIOMETRY_DU2

1215.1 SQL Query for view TREE_BIOMETRY_DU2

```
select TREE_BIOMETRY_DU1.*, (cnycnt+dgspcnt+cdgspcnt) as DixonCnt
from TREE_BIOMETRY_DU1
```

1216 View TREE_BIOMETRY_MC1

1216.1 SQL Query for view TREE_BIOMETRY_MC1

```
select 'TB' as MeasureCode, COUNT(*) as RowCnt,
SUM(DECODE(NVL(DomGenus,'0'),'0',0,1)*DECODE(NVL(DomSpecies,'0'),'0',0,1)) as DGSpCnt,
SUM(DECODE(NVL(CodomGenus,'0'),'0',0,1)*DECODE(NVL(CodomSpecies,'0'),'0',0,1)) as CdGSpCnt,
SUM(NVL(CanopyPlus,0)) +SUM (NVL(CanopyMinus,0)) AS CnyCntFROM as GLOBE.TREE_BIOMETRY
```

1217 View TREE_BIOMETRY_MC2

1217.1 SQL Query for view TREE_BIOMETRY_MC2

```
select TREE_BIOMETRY_MC1.*, (cnycnt+dgspcnt+cdgspcnt) as DixonCnt
from TREE_BIOMETRY_MC1
```

1218 View TREE_BIOMETRY_RPRT_VU

1218.1 SQL Query for view TREE_BIOMETRY_RPRT_VU

```
select Distinct as SchoolID, 'TB' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from TREE_BIOMETRY
group by SchoolID
```

1219 View TREE_BIOMETRY_VU1

1219.1 SQL Query for view TREE_BIOMETRY_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'TB' AS MeasureCode,
 COUNT(*) AS RowCnt,
 SUM(DECODE(NVL(DomGenus,'0'),'0',0,1)*DECODE(NVL(DomSpecies,'0'),'0',0,1)) AS DGSpCnt,
 SUM(DECODE(NVL(CodomGenus,'0'),'0',0,1)*DECODE(NVL(CodomSpecies,'0'),'0',0,1)) AS CdGSpCnt,
 SUM(NVL(CanopyPlus,0))+SUM (NVL(CanopyMinus,0)) AS CnyCnt,
 MIN(UTTimeMeasured) AS BegDate,
 MAX(UTTimeMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported-UTtimeMeasured),2) AS LagTimeAvg
FROM GLOBE.TREE_BIOMETRY
WHERE UTtimeReported < TRUNC (SYSDATE,'DD') + 22/24
GROUP BY SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1220 View TREE_BIOMETRY_VU2

1220.1 SQL Query for view TREE_BIOMETRY_VU2

```
select TREE_BIOMETRY_VU1.*, (cnycnt+dgspcnt+cdgspcnt) as DixonCnt
from TREE_BIOMETRY_VU1
```

1221 View TURBIDITY_WS_DR1

1222 View TURBIDITY_WS_DR2

1222.1 SQL Query for view TURBIDITY_WS_DR2

```
select TURBIDITY_WS_DR1.*, (3*secchicnt+tubecnt) as DixonCnt
from TURBIDITY_WS_DR1
```

1223 View TURBIDITY_WS_DU1

1224 View TURBIDITY_WS_DU2

1224.1 SQL Query for view TURBIDITY_WS_DU2

```
select TURBIDITY_WS_DU1.*, (3*secchicnt+tubecnt) as DixonCnt
from TURBIDITY_WS_DU1
```

1225 View TURBIDITY_WS_MC1

1225.1 SQL Query for view TURBIDITY_WS_MC1

```
select 'SW' as MeasureCode, COUNT(*) as RowCnt, COUNT((SecchiDepth1 + SecchiDepth2 + DistToSurface))
as SecchiCnt, COUNT(TubeDepth) as TubeCnt
from GLOBE.TURBIDITY_WS
```

1226 View TURBIDITY_WS_MC2

1226.1 SQL Query for view TURBIDITY_WS_MC2

```
select TURBIDITY_WS_MC1.*, (3*secchicnt+tubecnt) as DixonCnt
from TURBIDITY_WS_MC1
```

1227 View TURBIDITY_WS_RPRT_VU

1227.1 SQL Query for view TURBIDITY_WS_RPRT_VU

```
select Distinct as SchoolID, 'SW' as MeasureCode, MIN(UTTimeReported) as FirstReport,
MAX(UTTimeReported) as LastReport
from TURBIDITY_WS
group by SchoolID
```

1228 View TURBIDITY_WS_VU1

1228.1 SQL Query for view TURBIDITY_WS_VU1

```
select
 SchoolId,
 SiteId,
 TRUNC(DayMeasured,'MM') AS Month,
 'SW' AS MeasureCode, COUNT(*) AS RowCnt,
 COUNT((SecchiDepth1 + SecchiDepth2 + DistToSurface)) AS SecchiCnt,
 COUNT(TubeDepth) AS TubeCnt,
 MIN(DayMeasured) AS BegDate,
 MAX(DayMeasured) AS EndDate,
 MIN(UTtimeReported) AS BegRDate,
 MAX(UTtimeReported) AS EndRDate,
 ROUND(AVG(UTtimeReported - DayMeasured),2) AS LagTimeAvg
from GLOBE.TURBIDITY_WS
where UTtimeReported < TRUNC (SYSDATE,'DD')
group by SchoolId, SiteId, TRUNC(DayMeasured,'MM')
```

1229 View TURBIDITY_WS_VU2

1229.1 SQL Query for view TURBIDITY_WS_VU2

```
select TURBIDITY_WS_VU1.*, (3*secchicnt+tubecnt) as DixonCnt
from TURBIDITY_WS_VU1
```

1230 View Teacher_FirstActive

1230.1 SQL Query for view Teacher_FirstActive

```
select TeacherId, TRUNC(StatusChgDate) as FirstCWdate, SessionId
from GLOBE.REGISTRATION
where ( TeacherId
,
StatusChgDate ) IN  (
Select
TeacherId
,
M in (
StatusChgDate
)
From
GLOBE.REGISTRATION
Where
RegisterCode
=
'CW'
Group
By
TeacherId )
```

1231 View Teacher_FirstSchool

1231.1 SQL Query for view Teacher_FirstSchool

```
select TeacherId, SchoolId, TRUNC(DateBegin) as DateBegin
from GLOBE.TEACHER_TO_SCHOOL
where ( TeacherId
,
DateBeg in ) IN  (
Select
TeacherId
,
M in (
DateBeg
in
)
From
GLOBE.TEACHER_TO_SCHOOL
Group
By
TeacherId )
```

1232 View Teacher_PreService

1232.1 SQL Query for view Teacher_PreService

```
select FA.TeacherId, FA.SessionId, FA.FirstCWdate, FS.SchoolId as FirstSchool, FS.DateBegin as
FirstSchoolDate
from Teacher_FirstSchool FS, Teacher_FirstActive FA
where FA.TeacherId = FS.TeacherId  (
+
)  And  (
FS.DateBeg
in
Is
Null
OR
FA.FirstCWdate
<
FS.DateBeg
in
)
```

1233 View Trees_HR0

1233.1 SQL Query for view Trees_HR0

```
select
 TB.SchoolId,
 TB.SiteId,
 DayMeasured as Day,
 'T' as TorG
from
 GLOBE.TREE_BIOMETRY TB,
 GLOBE.SITE_LOCATION SL
where
 TB.SchoolId = SL.SchoolId AND
 TB.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 TB.DayMeasured >= TO_DATE('20000601','YYYYMMDD') AND
 TB.DomHeightAvg is not null AND
 TB.DomCircumAvg is not null
```

1234 View Turbidity_HR1

1234.1 SQL Query for view Turbidity_HR1

```
select
 SW.SchoolId,
 SW.SiteId,
 DayMeasured
from
 GLOBE.SURFACE_WATER SW,
 GLOBE.SITE_LOCATION SL
where
```

```
 SW.SchoolId = SL.SchoolId AND
 SW.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 SW.DayMeasured >= TO_DATE('20010201','YYYYMMDD') AND
 (SW.TurbiditySecchi is not null OR SW.TurbidityTube is not null)
```

1235 View Turbidity_HR2

1235.1 SQL Query for view Turbidity_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.DayMeasured,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 Count(T.DayMeasured) as GoodDays
from
 Turbidity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.DayMeasured) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 Count(T.DayMeasured) >= 8
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 Count(T.DayMeasured) as GoodDays
from
 Turbidity_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.DayMeasured,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.DayMeasured) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 Count(T.DayMeasured) >= 8
```

1236 View View_642

1237 View WATER_TEMP_HR1

1237.1 SQL Query for view WATER_TEMP_HR1

```
select
 SWC.SchoolId,
 SWC.SiteId,
 Month,
 WTempCnt as TempDays
from
 GLOBE.SURFACE_WATER_CNT SWC,
 GLOBE.SITE_LOCATION SL
where
 SWC.SchoolId = SL.SchoolId AND
 SWC.SiteId = SL.SiteId AND
 SL.DataSource in (1,2) AND
 Month >= TO_DATE('20010201','YYYYMMDD')  AND
 Month  <  TRUNC(SYSDATE,'MM')
```

1238 View WATER_TEMP_HR2

1238.1 SQL Query for view WATER_TEMP_HR2

```
select
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('19990201','YYYYMMDD'))/4,1))+1 as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.TempDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
```

```
 MAX(T.Month) as EndMonth
from
 WATER_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round(MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))/4,1))+1
having
 MIN(T.Month) < TRUNC(TRUNC(SYSDATE,'MM') - 2*33,'MM')
 AND
 SUM(T.TempDays) >= 4*30*(.70)
UNION ALL
select
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1)) as HRnbr,
 T.SchoolId,
 T.SiteId,
 SUM(T.TempDays) as GoodDays,
 MIN(T.Month) as BeginMonth,
 MAX(T.Month) as EndMonth
from
 WATER_TEMP_HR1 T
group by
 T.SchoolId, T.SiteId,
 2*TRUNC(Round((MONTHS_BETWEEN(T.Month,TO_DATE('1999 02 01','YYYY MM DD'))+2)/4,1))
having
 MIN(T.Month) > TO_DATE('1999 02 01','YYYY MM DD')
 AND
 SUM(T.TempDays) >= 4*30*(.70)
```

1239 View WORKSHOP_VIEW

1239.1 SQL Query for view WORKSHOP_VIEW

```
select WORKSHOP_SESSION.WORKSHOPNAME, WORKSHOP_SESSION.LOCATIONCITY, WORKSHOP_SESSION.MAXENROLLMENT,
WORKSHOP_SESSION.BEGINDATE, REGISTRATION.SESSIONID, GLOBE_TEACHER.WORKTELEPHONE,
GLOBE_TEACHER.HOMETELEPHONE, REGISTRATION.TEACHERID, REGISTRATION.REGISTERCODE,
REGISTRATION.STATUSCHGDATE, REGISTRATION.CREDITSCODE, REGISTRATION.COMMENTS
from WORKSHOP_SESSION, REGISTRATION, GLOBE_TEACHER
where WORKSHOP_SESSION.SESSIONID = REGISTRATION.SESSIONID
order by WORKSHOP_SESSION.SESSIONID ASC
```

1240 View WWW_USERS_VIEW

1240.1 SQL Query for view WWW_USERS_VIEW

```
select
 SCHOOLID AS UserID,
 SCHOOLNAME AS Name,
 CITY, STATEPROVCODE, COUNTRYCODE,
 'SCHOOL' AS TypeOfUser
from GLOBE_SCHOOL
where LIFECYCLECODE ! = 'C' )
union
(Select TeacherID, TeacherPrefix || ' ' || TeacherOtherNames || ' ' || TeacherSurname,
Globe_School.City, Globe_School.StateProvCode, Globe_School.CountryCode, 'PERSON' AS TypeOfuser FROM
Globe_Teacher, Globe_School WHERE Globe_Teacher.CurrentSchoolid(+) = Globe_School.SchoolIDand
TeacherID like 'ZZ%' )
union
(Select FranchiseID, FranchiseName, City, StateProvCode, CountryCode, 'FRANCHISE' AS TypeOfUser FROM
Globe_Franchise ) )
```

1241 View ZIPLOC_MISMATCH

1241.1 SQL Query for view ZIPLOC_MISMATCH

```
select Distinct GS.SchoolId
from Globe.Site_Location SL, GLOBE_SCHOOL GS, ZIP_INFO
where
 GS.COUNTRYCODE = 'US' and
 SL.SchoolId = GS.SchoolId and
 SL.SiteId = 'SCH-01' and
 Substr(GS.PostalCode,1,5 )= ZIP_INFO.ZIP and
 (ABS(SL.Latitude-Zip_Info.Latitude) > .3 or ABS(SL.Longitude-Zip_Info.Longitude ) > .3 )
```

1242 View ZIPMISSING_MISMATCH

1242.1 SQL Query for view ZIPMISSING_MISMATCH

```
select DISTINCT GS.SchoolId
from GLOBE_SCHOOL GS, ZIP_INFO
where COUNTRYCODE = 'US' and SUBSTR  (
GS.PostalCode
,
1
,
5 )  = Zip_Info.Zip  (
+
) and ZIP_INFO.ZIP is null
```

1243 View ZIPSTATECITY_MISMATCH

1243.1 SQL Query for view ZIPSTATECITY_MISMATCH

```
select DISTINCT GS.SchoolId
from GLOBE_SCHOOL GS, ZIP_INFO
where GS.COUNTRYCODE = 'US' and GS.CITY = Zip_Info.City  (
+
) and GS.STATEPROVCODE = Zip_Info.State  (
+
) and Substr  (
GS.PostalCode
,
1
,
5 )  = Zip_Info.Zip  (
+
) and ZIP_INFO.STATE is null
```

1244 View ZIP_TO_COUNTY

1244.1 SQL Query for view ZIP_TO_COUNTY

```
select DISTINCT Zip, County
from ZIP_INFO;
```

1245 Procedure CalcAOT

1245.1 Text of procedure CalcAOT

```
create or replace function calcAOT
  (v_inv0 IN number,
   v_inlam IN number,
   v_dvolt IN number,
   v_svolt IN number,
   v_press IN number,
   v_date IN date,
   v_lat IN number,
   v_long IN number)
   return number
IS
   v_monbr              number;
   v_jd                 number;
   v_pi                 number;
   v_dr                 number;
   v_t                  number;
   v_l0                 number;
   v_m                  number;
   v_mrad               number;
   v_e                  number;
   v_c                  number;
   v_lsav               number;
   v_ltrue              number;
   v_f                  number;
   v_radius             number;
   v_sidetime           number;
   v_ssave              number;
   v_obliq              number;
   v_rtasc              number;
   v_declin             number;
   v_hrangle            number;
   v_solelev            number;
   v_cosz               number;
   v_airm               number;
   v_aa                 number;
   v_bb                 number;
   v_cc                 number;
   v_dd                 number;
   v_tmplam             number;
   v_rayleigh           number;
   v_v0                 number;
   v_aot                number;
   v_transpct           number;
   v_aottrans           char;

BEGIN

  v_pi := 3.141592654;

  v_monbr := ceil((to_number(to_char(v_date,'MM'))-14)/12);
  v_jd := to_number(to_char(v_date,'DD')) - 32075
    + floor(1461*(to_number(to_char(v_date,'YYYY'))+4800+v_monbr)/4)
    + floor(367*(to_number(to_char(v_date,'MM'))-2-v_monbr*12)/12)
    - floor(3*(floor((to_number(to_char(v_date,'YYYY'))+4900+v_monbr)/100))/4);
  v_jd := (v_jd - 0.5 + to_number(to_char(v_date,'HH24'))/24
    + to_number(to_char(v_date,'MI'))/1440
    + to_number(to_char(v_date,'SS'))/86400);

  -- Solar position, ecliptic coordinates

  v_dr := v_pi/180.0;
  v_t := (v_jd - 2451545.0) / 36525.0;
  v_l0 := 280.46645 + 36000.76983  * v_t + 0.0003032 * v_t * v_t;
  v_m := 357.52910+35999.05030*v_t-0.0001559*v_t*v_t-0.00000048*v_t*v_t*v_t;
  v_mrad := v_m*v_dr;
  v_e := 0.016708617-0.000042037*v_t-0.0000001236*v_t*v_t;
  v_c := (1.914600-0.004817*v_t-0.000014*v_t*v_t)*sin(v_mrad)
    +(0.019993-0.000101*v_t)*sin(2.*v_mrad)+0.000290*sin(3.*v_mrad);

  -- Replacement code for L_true=fmod(L0+c,360.)
  v_lsav := (v_l0 + v_c) / 360.0;
  IF v_lsav < 0. THEN
```

```
        v_ltrue := (v_l0 + v_c)- ceil(v_lsav)*360;
  ELSE
        v_ltrue := (v_l0 + v_c)- floor(v_lsav)*360;
  END IF;

  IF v_ltrue < 0. THEN
        v_ltrue := v_ltrue + 360;
  END IF;

  v_f := v_mrad + v_c * v_dr;
  v_radius := 1.000001018 * (1.0 - v_e * v_e) / (1.0 + v_e * cos(v_f));

  --Sidereal time
  v_sidetime := 280.46061837+360.98564736629*(v_jd-2451545.0)
    + 0.000387933*v_t*v_t-v_t*v_t*v_t/38710000;

  -- Replacement code for Sidereal=fmod(Sidereal,360.)
  v_ssave := v_sidetime/360;
  IF v_ssave < 0 THEN
        v_sidetime := v_sidetime - ceil(v_ssave)*360;
  ELSE
        v_sidetime := v_sidetime - floor(v_ssave)*360;
  END IF;

  IF v_sidetime < 0 THEN
        v_sidetime := v_sidetime + 360;
  END IF;

  -- Obliquity
  v_obliq :=23.+26./60.+21.448/3600.-46.8150/3600.
    *v_t-0.00059/3600.*v_t*v_t+0.001813/3600.*v_t*v_t*v_t;

  -- Ecliptic to equatorial
  v_rtasc := atan2(sin(v_ltrue*v_dr)*cos(v_obliq*v_dr),
        cos(v_ltrue*v_dr));
  v_declin := asin(sin(v_obliq*v_dr)*sin(v_ltrue*v_dr));
  v_hrangle := v_sidetime+v_long-v_rtasc/v_dr;
  v_solelev := asin(sin(v_lat*v_dr)*sin(v_declin)+
        cos(v_lat*v_dr)*cos(v_declin)*cos(v_hrangle*v_dr));
  v_solelev := v_solelev/v_dr;

  -- Relative air mass
  v_cosz := cos((90.-v_solelev)*v_dr);
  v_airm := (1.002432*v_cosz*v_cosz+0.148386*v_cosz+0.0096467)
    /(v_cosz*v_cosz*v_cosz+0.149864*v_cosz*v_cosz+0.0102963*v_cosz+0.000303978);



  -- Rayleigh coefficient
  IF v_inlam <= 500 THEN
    v_aa := 0.00650362;
    v_bb := 3.55212;
    v_cc := 1.35579;
    v_dd := 0.11563;
  ELSE
    v_aa := 0.00864627;
    v_bb := 3.99668;
    v_cc := 0.0110298;
    v_dd := 0.0271393;
  END IF;

  v_tmplam := v_inlam / 1000;
  v_rayleigh := v_aa * power(v_tmplam,(-v_bb-v_cc*v_tmplam-v_dd/v_tmplam));

  -- AOT calculation
  IF v_inv0 IS NOT NULL  THEN
           v_v0 := v_inv0;
  ELSE
           v_v0 := 2.5;
  END IF;

  v_aot := (ln(v_v0/v_radius/v_radius)-ln(v_svolt-v_dvolt)
    - v_rayleigh*v_press/1013.25*v_airm)/v_airm;

  v_transpct := 100*exp(-v_aot);

--  v_aottrans := to_char(v_aot) || ',' || to_char(v_transpct);
  return v_aot;
end calcAOT;
/
```

```
alter procedure %PROC% compile
/
```

1246 Procedure CalcAZTransPct

1246.1 Text of procedure CalcAZTransPct

```
create or replace calcAZTransPct
  (v_aot IN number)
   return number
IS
   v_transpct           number;

BEGIN

  v_transpct := 100*exp(-v_aot);

  return v_transpct;
end calcAZTransPct;
/

alter procedure %PROC% compile
/
```

1247 Procedure DiffsolarNoonSQL

1247.1 Text of procedure DiffsolarNoonSQL

```
create or replace function diffsolarnoon
        (v_date IN date,
         v_lon IN number,
         v_min IN number)
        return number
IS
        v_diffsolarnoon number;
        v_uttime        number;
        v_truesolarnoon number;
BEGIN
        v_uttime := uttime(v_date, v_lon);
        v_truesolarnoon := truesolarnoon(v_date, v_lon);
        v_diffsolarnoon := round(v_uttime - v_truesolarnoon);
        IF v_min IS NULL THEN
            IF v_diffsolarnoon < 0 THEN
                    IF v_diffsolarnoon > -3600 THEN
                           v_diffsolarnoon := 0;
                    ELSE
                           v_diffsolarnoon := v_diffsolarnoon + 3600;
                    END IF;
                END IF;
        END IF;
        return(v_diffsolarnoon);
end diffsolarnoon;
/

alter procedure %PROC% compile
/
```

1248 Procedure EqtimeSQL

1248.1 Text of procedure EqtimeSQL

```
create or replace function eqtime
        (tunix IN number)
        return number
-- Compute the equation of time in seconds
-- for the supplied unix timestamp
-- Equations taken from:
-- Meeus, Jean. _Astronomical Formulae for Calculators, Second Ed._
-- Richmond, VA: Willmann-Bell, 1982.
IS
        v_eqtime        number;
```

```
        dtor            number := 57.29577951;
        T               number;
        L               number;
        M               number;
        e1              number;
        E2              number;
        y               number;
        EQ              number;
BEGIN
-- Compute T directly from unix timestamp.
-- n.b. 25567.5/36525 == 0.7 is offset between epoch
-- of T (1899123112) and epoch of tunix (1970010100).
        T := tunix/86400/36525 + 0.7;
        L := (0.0003025 * T + 36000.76892)* T + 279.69668;
        M := ((-0.0000033 * T - 0.000150) * T + 35999.04975) * T + 358.47583;
        e1 := (-0.000000126 * T - 0.0000418) * T + 0.01675104;
        E2 := ((0.000000503 * T - 0.00000164) * T - 0.0130125 * T) + 23.452294;
        y := power(tan(0.5 * E2/ dtor), 2);
-- Compute EQ in radians.
        EQ := y * sin(2*L/dtor)
               - 2*e1*sin(M/dtor)
               + 4*e1*y*sin(M/dtor)*cos(2*L/dtor)
               - 0.5*y*y*sin(4*L/dtor)
               - 1.25*e1*e1*sin(2*M/dtor);
-- Convert EQ from radians to seconds of time.
        v_eqtime := 240*dtor*EQ;
        return(v_eqtime);
end eqtime;
/

alter procedure %PROC% compile
/
```

1249 Procedure ListMonthsCert

1249.1 Text of procedure ListMonthsCert

```
create or replace FUNCTION  ListMonthsCert
    ( SchoolIdParm   IN   VarChar2,
      CertNbr        IN   Integer,
      HonorRollParm  IN   VarChar2 )
RETURN VarChar2 IS
  CURSOR MonthCursor IS
    SELECT CA."End Month"
    FROM
      GLOBE.HR_CATEGORIES_ALL CA
    WHERE
        CA."SchoolId" = SchoolIdParm
      AND
        CA."Honor Roll" = HonorRollParm
      AND
        TRUNC((CA."HRnbr"+5)/6) = CertNbr
    ORDER BY CA."HRnbr" ;
  MonthTxt VarChar2(15) ;
  StringList VarChar2(500) ;
  MonthCnt Integer ;
BEGIN
  StringList := '' ;
  SELECT Count(*) INTO MonthCnt
    FROM
      GLOBE.HR_CATEGORIES_ALL CA
    WHERE
        CA."SchoolId" = SchoolIdParm
      AND
        CA."Honor Roll" = HonorRollParm
      AND
        TRUNC((CA."HRnbr"+5)/6) = CertNbr ;
  IF MonthCnt = 0 THEN RETURN StringList; END IF ;
  OPEN MonthCursor ;
  FOR i IN 1..MonthCnt LOOP
    FETCH MonthCursor INTO MonthTxt ;
    StringList := StringList || MonthTxt ;
    IF i = MonthCnt THEN
      BEGIN
        CLOSE MonthCursor ;
        RETURN StringList ;
      END ;
```

```
    END IF ;
    IF i = MonthCnt - 1
      THEN StringList := StringList || ' and ' ;
      ELSE StringList := StringList || ', ' ;
    END IF ;
  END LOOP ;
END ListMonthsCert ;
/

alter function ListMonthsCert compile
/
```

1250 Procedure ListMonthsNew

1250.1 Text of procedure ListMonthsNew

```
create or replace FUNCTION ListMonthsNew
    ( SchoolIdParm   IN   VarChar2,
      HonorRollParm  IN   VarChar2 )
RETURN VarChar2 IS
  CURSOR MonthCursor IS
    SELECT CN."End Month"
    FROM
      GLOBE.HR_CATEGORIES_NEW CN
    WHERE
        CN."SchoolId" = SchoolIdParm
      AND
        CN."Honor Roll" = HonorRollParm
    ORDER BY CN."HRnbr" ;
  MonthTxt VarChar2(15) ;
  StringList VarChar2(500) ;
  MonthCnt Integer ;
BEGIN
  StringList := '' ;
  SELECT Count(*) INTO MonthCnt
    FROM
      GLOBE.HR_CATEGORIES_NEW CN
    WHERE
        CN."SchoolId" = SchoolIdParm
      AND
        CN."Honor Roll" = HonorRollParm ;
  IF MonthCnt = 0 THEN RETURN StringList; END IF ;
  OPEN MonthCursor ;
  FOR i IN 1..MonthCnt LOOP
    FETCH MonthCursor INTO MonthTxt ;
    StringList := StringList || MonthTxt ;
    IF i = MonthCnt THEN
      BEGIN
        CLOSE MonthCursor ;
        RETURN StringList ;
      END ;
    END IF ;
    IF i = MonthCnt - 1
      THEN StringList := StringList || ' and ' ;
      ELSE StringList := StringList || ', ' ;
    END IF ;
  END LOOP ;
END ListMonthsNew ;
/

alter function ListMonthsNew compile
/
```

1251 Procedure ListMonthsPrev

1251.1 Text of procedure ListMonthsPrev

```
create or replace FUNCTION ListMonthsPrev
    ( SchoolIdParm   IN   VarChar2,
      HonorRollParm  IN   VarChar2 )
RETURN VarChar2 IS
  CURSOR MonthCursor IS
    SELECT CN."End Month"
    FROM
      GLOBE.HR_CATEGORIES_NEW CN
```

```
    WHERE
        CN."SchoolId" = SchoolIdParm
      AND
        CN."Honor Roll" = HonorRollParm
      AND
        CN."HRnbr" < TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
        TO_DATE('19990401','YYYYMMDD')))/2)-5
    ORDER BY CN."HRnbr" ;
  MonthTxt VarChar2(15) ;
  StringList VarChar2(500) ;
  MonthCnt Integer ;
BEGIN
  StringList := '' ;
  SELECT Count(*) INTO MonthCnt
    FROM
      GLOBE.HR_CATEGORIES_NEW CN
    WHERE
        CN."SchoolId" = SchoolIdParm
      AND
        CN."Honor Roll" = HonorRollParm ;
  IF MonthCnt = 0 THEN RETURN StringList; END IF ;
  OPEN MonthCursor ;
  FOR i IN 1..MonthCnt LOOP
    FETCH MonthCursor INTO MonthTxt ;
    StringList := StringList || MonthTxt ;
    IF i = MonthCnt THEN
      BEGIN
        CLOSE MonthCursor ;
        RETURN StringList ;
      END ;
    END IF ;
    IF i = MonthCnt - 1
      THEN StringList := StringList || ' and ' ;
      ELSE StringList := StringList || ', ' ;
    END IF ;
  END LOOP ;
END ListMonthsPrev ;
/

alter function ListMonthsPrev compile
/
```

1252 Procedure ListMonthsYear

1252.1 Text of procedure ListMonthsYear

```
create or replace FUNCTION  ListMonthsYear
    ( SchoolIdParm   IN   VarChar2,
      HonorRollParm  IN   VarChar2 )
RETURN VarChar2 IS
  CURSOR MonthCursor IS
    SELECT CA."End Month"
    FROM
      GLOBE.HR_CATEGORIES_ALL CA
    WHERE
        CA."SchoolId" = SchoolIdParm
      AND
        CA."Honor Roll" = HonorRollParm
      AND
        CA."HRnbr" >= TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
        TO_DATE('19990401','YYYYMMDD')))/2) - 5
    ORDER BY CA."HRnbr" ;
  MonthTxt VarChar2(15) ;
  StringList VarChar2(500) ;
  MonthCnt Integer ;
BEGIN
  StringList := '' ;
  SELECT Count(*) INTO MonthCnt
    FROM
      GLOBE.HR_CATEGORIES_ALL CA
    WHERE
        CA."SchoolId" = SchoolIdParm
      AND
        CA."Honor Roll" = HonorRollParm
      AND
        CA."HRnbr" >= TRUNC((MONTHS_BETWEEN(TRUNC(SYSDATE,'MM'),
        TO_DATE('19990401','YYYYMMDD')))/2) - 5 ;
```

```
  IF MonthCnt = 0 THEN RETURN StringList; END IF ;
  OPEN MonthCursor ;
  FOR i IN 1..MonthCnt LOOP
    FETCH MonthCursor INTO MonthTxt ;
    StringList := StringList || MonthTxt ;
    IF i = MonthCnt THEN
      BEGIN
        CLOSE MonthCursor ;
        RETURN StringList ;
      END ;
    END IF ;
    IF i = MonthCnt - 1
      THEN StringList := StringList || ' and ' ;
      ELSE StringList := StringList || ', ' ;
    END IF ;
  END LOOP ;
END ListMonthsYear ;
/

alter function ListMonthsYear compile
/
```

1253 Procedure ListTeachers

1253.1 Text of procedure ListTeachers

```
create or replace FUNCTION       ListTeachers (SchoolIdParm IN VarChar2)
RETURN VarChar2 IS
  CURSOR TeacherName IS
    SELECT GT.TeacherOtherNames, GT.TeacherSurname
    FROM
      GLOBE.TEACHER_TO_SCHOOL TS,
      GLOBE.GLOBE_TEACHER GT
    WHERE
        TS.SchoolId = SchoolIdParm
      AND
        TS.DateEnd IS NULL
      AND
        TS.TeacherId = GT.TeacherId
      AND
        GT.LifeCycleCode <> 'I'
    ORDER BY GT.TeacherSurname ;
  FirstName GLOBE.GLOBE_TEACHER.TeacherOtherNames%TYPE ;
  LastName GLOBE.GLOBE_TEACHER.TeacherSurname%TYPE ;
  StringList VarChar2(500) ;
  TeacherCnt Integer ;
BEGIN
  StringList := '' ;
  SELECT Count(*) INTO TeacherCnt
    FROM
      GLOBE.TEACHER_TO_SCHOOL TS,
      GLOBE.GLOBE_TEACHER GT
    WHERE
        TS.SchoolId = SchoolIdParm
      AND
        TS.DateEnd IS NULL
      AND
        TS.TeacherId = GT.TeacherId
      AND
        GT.LifeCycleCode <> 'I' ;
  IF TeacherCnt = 0 THEN RETURN StringList; END IF ;
  OPEN TeacherName ;
  FOR i IN 1..TeacherCnt LOOP
    FETCH TeacherName INTO FirstName, LastName ;
    StringList := StringList || FirstName
                  || ' ' || LastName ;
    IF i = TeacherCnt THEN
      BEGIN
        CLOSE TeacherName ;
        RETURN StringList ;
      END ;
    END IF ;
    IF i = TeacherCnt - 1
      THEN StringList := StringList || ' and ' ;
      ELSE StringList := StringList || ', ' ;
    END IF ;
  END LOOP ;
```

```
END ListTeachers ;
/

alter function ListTeachers compile
/
```

1254 Procedure LocalnoonSQL

1254.1 Text of procedure LocalnoonSQL

```
create or replace function localnoon
        (v_date IN date,
         v_lon IN number)
        return number
IS
        tunix           number;
        EQ              number;
        noon            number;
        hh              number;
        v_uttolocal     number;
        v_solartime     number;
        v_solarhour     number;
BEGIN
        hh := to_number(substr(to_char(v_date, 'HH24:YY-MON-DD'), 1, 2));
        tunix := uttime(v_date, v_lon);
        EQ := eqtime(tunix) / 3600;
        noon := 12 - EQ - v_lon/15;
        IF (noon - hh) > 12.3 THEN
                noon := noon - 24;
        END IF;
        IF (hh - noon) > 12.3 THEN
                noon := noon + 24;
        END IF;
-- Iterate once to ensure self-consistency
        tunix := tunix + 3600*(noon - hh);
        EQ := eqtime(tunix)/3600;
        noon := 12 - EQ - v_lon/15;
        IF (noon - hh) > 12.3 THEN
                noon := noon - 24;
        END IF;
        IF (hh - noon) > 12.3 THEN
                noon := noon + 24;
        END IF;
        return(noon);
end localnoon;
/

alter procedure %PROC% compile
/
```

1255 Procedure TruesolarnoonSQL

1255.1 Text of procedure TruesolarnoonSQL

```
create or replace function truesolarnoon
        (v_date IN date,
         v_lon IN number)
        return number
IS
        v_uttime        number;
        v_truesolarnoon number;
BEGIN
        v_truesolarnoon := localnoon(v_date, v_lon);
        v_uttime := uttime(v_date, v_lon);
        v_truesolarnoon := v_uttime - (mod(v_uttime, 86400))
                                + trunc(3600 * v_truesolarnoon);
        return(v_truesolarnoon);
end truesolarnoon;
/

alter procedure %PROC% compile
/
```

1256 Procedure UTtimeSQL

1256.1 Text of procedure UTtimeSQL

```
create or replace function uttime
        (v_date IN date,
         v_lon IN number)
        return number
IS
        v_uttime        number;
begin
        v_uttime := round((v_date - to_date('197001010000', 'YYYYMMDDHH24MI')) * 86400);
        return(v_uttime);
end uttime;
/

alter procedure %PROC% compile
/
```