

Table Report

ROSS Schema 5

T_AGENCIES

Primary key: AGENCY_ID

Foreign keys	Child	Parent
AGN_DRL_FK	T_DRILLERS.AGN_AGENCY_ID	AGENCY_ID
AGN_PRJ_MANAGING_FK	T_PROJECTS.AGN_AGENCY_ID_MANAGI NG	AGENCY_ID
AGN_PRJ_POSSESSING_FK	T_PROJECTS.AGN_AGENCY_ID_POSSESSI NG	AGENCY_ID

Column details

1. AGENCY_ID

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Not allowed
 Notes: Uniquely identifies an agency.

2. AGENCY_NAME

Physical data type: VARCHAR2(100)
 Portable data type: C-Variable Length(100)
 Allow NULLs: Allowed
 Notes: Name of agency.

Code details

1. AGN_TRG

Type: Trigger
 Code body:

```

create sequence agn_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger AGN_TRG BEFORE INSERT ON "T_AGENCIES" FOR EACH ROW

BEGIN
  if :new.agency_id is null then
    SELECT AGN_SEQ.NEXTVAL
    INTO :NEW.AGENCY_ID
    FROM DUAL;
  end if;
END;
/

```

T_ANALYTICAL_METHODS

Notes: Lookup table of sample analysis methods.
Primary key: ANALYTICAL_METHOD_ID

Foreign keys	Child	Parent
AM_SAMP_FK	T_SAMPLES.AM_ANALYTICAL_METHOD_ID	ANALYTICAL_METHOD_ID

Column details**1. ANALYTICAL_METHOD_ID**

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a sample analysis method.

2. ANALYTICAL_METHOD_NAME

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Name of analysis method.

3. ANALYTICAL_METHOD_DESCRIPTION

Physical data type: VARCHAR2(500)
Portable data type: C-Variable Length(500)
Allow NULLs: Not allowed
Notes: Description of method.

Code details**1. AM_TRIG**

Type: Trigger
Code body:

```

create sequence am_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger AM_TRIG BEFORE INSERT ON "T_ANALYTICAL_METHODS" FOR
EACH ROW

BEGIN
  if :new.analytical_method_id is null then
    SELECT AM_SEQ.NEXTVAL
    INTO :NEW.ANALYTICAL_METHOD_ID
    FROM DUAL;
  end if;
END;
/

```

T_ANGULARITY_LU

Notes: Lookup table of angularity descriptors.
Primary key: ANGULARITY_ID

Foreign keys	Child	Parent
ALU_SAMP_FK	T_SAMPLES.ALU_ANGULARITY_ID	ANGULARITY_ID

Column details**1. ANGULARITY_ID**

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies an angularity

2. ANGULARITY

Physical data type: VARCHAR2(15)
Portable data type: C-Variable Length(15)
Allow NULLs: Not allowed
Notes: Angularity description.

Code details**1. ALU_TRG**

Type: Trigger
Code body:

```

create sequence alu_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger ALU_TRG BEFORE INSERT ON "T_ANGULARITY_LU" FOR EACH
ROW

BEGIN
  if :new.angularity_id is null then
    SELECT ALU_SEQ.NEXTVAL
    INTO :NEW.ANGULARITY_ID
    FROM DUAL;
  end if;
END;
/

```

T_BIBSUMMARY

Primary key: PK_BIBSUMMARY

Column details**1. PK_BIBSUMMARY**

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Not allowed

2. AUTHOR

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed

3. AUTHOR LAST NAME

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed

4. AUTHOR INITIALS

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed

5. TITLE

Physical data type: VARCHAR2(1000)
 Portable data type: C-Variable Length(1000)
 Allow NULLs: Allowed

6. KEYWORDS

Physical data type: VARCHAR2(1000)
 Portable data type: C-Variable Length(1000)
 Allow NULLs: Allowed

7. PAPER_YEAR

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed

8. ABSTRACT

Physical data type: LONG
 Portable data type: C-Large Length
 Allow NULLs: Allowed

9. PUBLISHER

Physical data type: VARCHAR2(1000)
 Portable data type: C-Variable Length(1000)
 Allow NULLs: Allowed

T_CALCULATION_METHODS

Notes: Lookup table of methods used to calculate sample mean, median, standard deviation, skewness and kurtosis.

Primary key: CALCULATION_METHOD_ID

Foreign keys	Child	Parent
CALC_SAMP_MEAN_FK	T_SAMPLES.CALC_CALC_METHOD_ID_M EAN	CALCULATION_METHOD_ID
CALC_SAMP_MEDIAN_FK	T_SAMPLES.CALC_CALC_METHOD_ID_M EDIAN	CALCULATION_METHOD_ID
CALC_SAMP_STD_FK	T_SAMPLES.CALC_CALC_METHOD_ID_ST D	CALCULATION_METHOD_ID
CALC_SAMP_SKEWNESS_FK	T_SAMPLES.CALC_CALC_METHOD_ID_SK EW	CALCULATION_METHOD_ID
CALC_SAMP_KURTOSIS_FK	T_SAMPLES.CALC_CALC_METHOD_ID_K URT	CALCULATION_METHOD_ID

Column details**1. CALCULATION_METHOD_ID**

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a calculation method

2. CALCULATION_METHOD_NAME

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Name of calculation method.

3. CALCULATION_METHOD_DESCRIPTION

Physical data type: VARCHAR2(500)
Portable data type: C-Variable Length(500)
Allow NULLs: Not allowed
Notes: Description of method.

Code details**1. CALC_TRIG**

Type: Trigger
Code body:

```

create sequence calc_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger CALC_TRIG BEFORE INSERT ON "T_CALCULATION_METHODS" FOR
EACH ROW

BEGIN
  if :new.calculation_method_id is null then
    SELECT CALC_SEQ.NEXTVAL
    INTO :NEW.CALCULATION_METHOD_ID
    FROM DUAL;
  end if;
END;
/

```

T_COLLECTION_METHODS

Notes: Lookup table of collection methods.
Primary key: COLLECTION_METHOD_ID

Foreign keys	Child	Parent
CM_CORE_FK	T_CORES.CM_COLLECTION_METHOD_ID	COLLECTION_METHOD_ID
CM_SAMP_FK	T_SAMPLES.CM_COLLECTION_METHOD_ID	COLLECTION_METHOD_ID

Column details

1. COLLECTION_METHOD_ID

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a collection method record.

2. COLLECTION_METHOD

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Name of collection method.

3. COLLECTION_METHOD_DESCRIPTION

Physical data type: VARCHAR2(500)
Portable data type: C-Variable Length(500)
Allow NULLs: Not allowed
Notes: Description of collection method.

Code details

1. CM_TRG

Type: Trigger
Code body:

```

create sequence cm_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "CM_TRG" BEFORE INSERT ON "T_COLLECTION_METHODS" FOR
EACH ROW

BEGIN
  if :new.collection_method_id is null then
    SELECT CM_SEQ.NEXTVAL
    INTO :NEW.COLLECTION_METHOD_ID
    FROM DUAL;
  end if;
END;
/

```

T_COLOR_DESCRIPTOR

Notes: Lookup table of color matrix color descriptors.
Primary key: COLOR_DESCRIPTOR_ID

Foreign keys	Child	Parent
CD_CMTX_FK	T_COLOR_MATRIX.CD_DESCRIPTOR_ID	COLOR_DESCRIPTOR_ID

Column details**1. COLOR_DESCRIPTOR_ID**

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Unique identifier of a color matrix descriptor.

2. COLOR_DESCRIPTOR

Physical data type: VARCHAR2(20)
Portable data type: C-Variable Length(20)
Allow NULLs: Not allowed
Notes: The descriptor portion of a color matrix (e.g.: "Greenish")

Code details**1. CD_TRG**

Type: Trigger

Code body:

```

create sequence cd_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "CD_TRG" BEFORE INSERT ON "T_COLOR_DESCRIPTOR" FOR
EACH ROW

BEGIN
  if :new.color_descriptor_id is null then
    SELECT CD_SEQ.NEXTVAL
    INTO :NEW.COLOR_DESCRIPTOR_ID
    FROM DUAL;
  end if;
END;
/

```

T_COLOR_MATRIX

Notes: Complete list of possible color matrix combinations.
Primary key: COLOR_MATRIX_ID

Foreign keys	Child	Parent
CD_CMTX_FK	CD_DESCRIPTOR_ID	T_COLOR_DESCRIPTOR.COLOR_DESCRIPTOR_ID
CT_CMTX_FK	CT_COLOR_TONE_ID	T_COLOR_TONES.COLOR_TONE_ID
COL_CMTX_FK	COL_COLOR_ID	T_COLORS.COLOR_ID
CMTX_CORE_FK	T_CORE_LAYERS.CMTX_COLOR_MATRIX_ID	COLOR_MATRIX_ID
CMTX_MC_FK	T_MUNSELL_COLORS.CMTX_COLOR_MATRIX_ID	COLOR_MATRIX_ID
CMTX_SAMP_FK	T_SAMPLES.CMTX_COLOR_MATRIX_ID	COLOR_MATRIX_ID

Column details

1. COLOR_MATRIX_ID

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: System assigned unique identifier of a color matrix.

2. CT_COLOR_TONE_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: The tone portion of a color matrix (e.g.: "Dark")

3. CD_DESCRIPTOR_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: The descriptor portion of a color matrix (e.g.: "Greenish")

4. COL_COLOR_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: The color portion of a color matrix (e.g.: "Brown")

Code details

1. CMTX_TRG

Type: Trigger
Code body:

```

create sequence cmtx_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "CMTX_TRG" BEFORE INSERT ON "T_COLOR_MATRIX" FOR EACH
ROW

BEGIN
  if :new.color_matrix_id is null then
    SELECT CMTX_SEQ.NEXTVAL
    INTO :NEW.COLOR_MATRIX_ID
    FROM DUAL;
  end if;
END;
/

```


T_COLOR_TONES

Notes: Lookup table of color matrix color tones.
Primary key: COLOR_TONE_ID

Foreign keys	Child	Parent
CT_CMTX_FK	T_COLOR_MATRIX.CT_COLOR_TONE_ID	COLOR_TONE_ID

Column details**1. COLOR_TONE_ID**

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Unique identifier of a color matrix color tone.

2. COLOR_TONE

Physical data type: VARCHAR2(10)
Portable data type: C-Variable Length(10)
Allow NULLs: Not allowed
Notes: The tone portion of a color matrix (e.g.: "Dark")

Code details**1. CT_TRG**

Type: Trigger
Code body:

```

create sequence ct_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "CT_TRG" BEFORE INSERT ON "T_COLOR_TONES" FOR EACH ROW

BEGIN
  if :new.color_tone_id is null then
    SELECT CT_SEQ.NEXTVAL
    INTO :NEW.COLOR_TONE_ID
    FROM DUAL;
  end if;
END;
/

```

T_COLORS

Notes: Lookup table of color matrix color names.
Primary key: COLOR_ID

Foreign keys	Child	Parent
COL_CMTX_FK	T_COLOR_MATRIX.COL_COLOR_ID	COLOR_ID

Column details**1. COLOR_ID**

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Unique identifier of a color matrix color.

2. COLOR

Physical data type: VARCHAR2(10)
Portable data type: C-Variable Length(10)
Allow NULLs: Not allowed
Notes: The color portion of a color matrix (e.g.: "Brown")

Code details**1. COL_TRG**

Type: Trigger
Code body:

```

create sequence col_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "COL_TRG" BEFORE INSERT ON "T_COLORS" FOR EACH ROW

BEGIN
  if :new.color_id is null then
    SELECT COL_SEQ.NEXTVAL
    INTO :NEW.COLOR_ID
    FROM DUAL;
  end if;
END;
/

```

T_CONTACTS

Primary key: CONTACT_ID

Foreign keys	Child	Parent
CON_PRJ_FK	T_PROJECTS.CON_CONTACT_ID	CONTACT_ID

Column details**1. CONTACT_ID**

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Not allowed
 Notes: Uniquely identifies a contact record.

2. CONTACT_NAME

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed
 Notes: The name of the person.

3. CONTACT_PHONE

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Allowed
 Notes: The phone number of the person.

Code details**1. CON_TRG**

Type: Trigger
 Code body:

```

create sequence con_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "CON_TRG" BEFORE INSERT ON "T_CONTACTS" FOR EACH ROW

BEGIN
  if :new.contact_id is null then
    SELECT CON_SEQ.NEXTVAL
    INTO :NEW.CONTACT_ID
    FROM DUAL;
  end if;
END;
/

```

T_CORE_LAYER_QUALIFIERS

Notes: Stores standardized text qualifiers of core layers.
Primary key: CORE_LAYER_QUALIFIER_ID

Foreign keys	Child	Parent
CL_CLQ_FK	CL_CORE_LAYER_ID	T_CORE_LAYERS.CORE_LAYER_ID
STX_CLQ_FK	STX_SOIL_TEXTURE_ID	T_SOIL_TEXTURES.SOIL_TEXTURE_ID
SD_CLQ_FK	SD_SOIL_DESCRIPTOR_ID	T_SOIL_DESCRIPTOR.SOIL_DESCRIPTOR_ID
ST_CLQ_FK	ST_SOIL_TYPE_ID	T_SOIL_TYPES.SOIL_TYPE_ID
L_CLQ_FK	L_LITHOLOGY_ID	T_LITHOLOGY.LITHOLOGY_ID
S_CLQ_FK	S_SORTING_ID	T_SORTING.SORTING_ID
QLU_CLQ_FK	QLU_QUALIFIER_ID	T_QUALIFIERS_LU.QUALIFIER_ID

Column details

1. CORE_LAYER_QUALIFIER_ID

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Uniquely describes a core layer qualifier record.

2. CL_CORE_LAYER_ID (FK)

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Identifies the core layer qualified by this record.

3. STX_SOIL_TEXTURE_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Allowed
Notes: Sediment texture (grainsize). Foreign key to table of soil textures.

4. SD_SOIL_DESCRIPTOR_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Allowed
Notes: Sediment descriptor (e.g.: the "silty" in "silty sand"). Foreign key to table of soil descriptors.

5. ST_SOIL_TYPE_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Allowed
Notes: Sediment type (e.g.: the "sand" in "silty sand"). Foreign key to table of soil types.

6. L_LITHOLOGY_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Allowed
Notes: Sand lithology (e.g.: the "quartz" in "quartz sand"). Foreign key to table of lithologies.

7. S_SORTING_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Allowed
Notes: Describes the sorting of this qualifier. Foreign key to table T_SORTING.

8. QLU_QUALIFIER_ID (FK)

Physical data type: VARCHAR2(25)
Portable data type: C-Variable Length(25)
Allow NULLs: Allowed
Notes: Describes the relative amount of material for this core layer qualifier. Foreign key to table T_QUALIFIER_LU.

Code details

1. CLQ_TRG

Type:

Trigger

Code body:

```
create sequence clq_seq INCREMENT BY 1 START WITH 1
    MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
    NOCACHE NOORDER;
```

```
create or replace trigger "CLQ_TRG" BEFORE INSERT ON "T_CORE_LAYER_QUALIFIERS"
FOR EACH ROW
```

```
BEGIN
```

```
    if :NEW.CORE_LAYER_QUALIFIER_ID is null then
```

```
        SELECT CLQ_SEQ.NEXTVAL
        INTO :NEW.CORE_LAYER_QUALIFIER_ID
        FROM DUAL;
```

```
    end IF;
```

```
END;
```

```
/
```

T_CORE_LAYERS

Notes: Identified distinct layers within a core.
Primary key: CORE_LAYER_ID

Foreign keys	Child	Parent
CMTX_CORE_FK	CMTX_COLOR_MATRIX_ID	T_COLOR_MATRIX.COLOR_MATRIX_ID
CL_LS_FK	LS_LAYER_STRUCTURE_ID	T_LAYER_STRUCTURES.LAYER_STRUCTURE_ID
CORE_CL_FK	CORE_CORE_ID	T_CORES.CORE_ID
USCS_CL_FK	USCS_USCS_CLASSIFICATION_ID	T_USCS_CLASSIFICATIONS.USCS_CLASSIFICATION_ID
CL_SXC_FK	T_SAMPLES_X_CORELAYERS.CL_CORE_LAYER_ID	CORE_LAYER_ID
CL_CLQ_FK	T_CORE_LAYER_QUALIFIERS.CL_CORE_LAYER_ID	CORE_LAYER_ID

Column details

1. CORE_LAYER_ID

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a core layer record within this table.

2. CORE_CORE_ID (FK)

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Identifies the core that this layer is in.

3. CORE_LAYER_IDENTIFIER

Physical data type: VARCHAR2(20)
Portable data type: C-Variable Length(20)
Allow NULLs: Allowed
Notes: Project-unique core layer identifier assigned by agency performing core drilling. Does not uniquely identify records within this table.

4. LS_LAYER_STRUCTURE_ID (FK)

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Allowed
Notes: Describes the layer structure.

5. USCS_USCS_CLASSIFICATION_ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Identifies the unified soils classification system soil type. Foreign key to table of USCS_CLASSIFICATIONS.

6. CMTX_COLOR_MATRIX_ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Describes the color matrix.

7. BOTTOM_OF_LAYER_INTERVAL

Physical data type: NUMBERPS(5,1)
Portable data type: N-Decimal(5,1)
Allow NULLs: Not allowed
Notes: Ending measurement within a core for a specific core layer. Positive values in feet, greater than or equal to TOP_OF_LAYER_INTERVAL.

8. TOP_OF_LAYER_INTERVAL

DataDictionary

Physical data type: NUMBERPS(5,1)
Portable data type: N-Decimal(5,1)
Allow NULLs: Not allowed
Notes: Beginning measurement within a core for a specific core layer. Positive values in feet, greater than or equal to 0.

9. MUNSELL HUE WET

Physical data type: VARCHAR2(6)
Portable data type: C-Variable Length(6)
Allow NULLs: Allowed
Notes: The hue portion of the Munsell color code for this layer.

10. MUNSELL VALUE WET

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: The value portion of the Munsell color code for this layer.

11. MUNSELL CHROMA WET

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: The chroma portion of the Munsell color code for this layer.

12. CORE LAYER COMMENTS

Physical data type: VARCHAR2(2000)
Portable data type: C-Variable Length(2000)
Allow NULLs: Allowed
Notes: Comments recorded for this layer.

Code details

1. CL TRG

Type: Trigger
Code body:

```
create sequence cl_seq INCREMENT BY 1 START WITH 1
    MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
    NOCACHE NOORDER;

create or replace trigger "CL_TRG" BEFORE INSERT ON "T_CORE_LAYERS" FOR EACH ROW

BEGIN
    if :NEW.CORE_LAYER_ID is null then
        SELECT CL_SEQ.NEXTVAL
        INTO :NEW.CORE_LAYER_ID
        FROM DUAL;
    end if;
END;
/
```

T_CORES

Notes: Sand cores.
 Primary key: CORE_ID

Foreign keys	Child	Parent
CM_CORE_FK	CM_COLLECTION_METHOD_ID	T_COLLECTION_METHODS.COLLECTION_METHOD_ID
PRJ_CORE_FK	PRJ_PROJECT_ID	T_PROJECTS.PROJECT_ID
DRL_COR_FK	DRL_DRILLER_ID	T_DRILLERS.DRILLER_ID
CORE_CL_FK	T_CORE_LAYERS.CORE_CORE_ID	CORE_ID
CORE_SAMP_FK	T_SAMPLES.CORE_CORE_ID	CORE_ID
CORE_LOC_FK	T_LOCATIONS.CORE_CORE_ID	CORE_ID

Column details	
<u>1. CORE ID</u>	
Physical data type:	NUMBERPS(10,0)
Portable data type:	N-Decimal(10,0)
Allow NULLs:	Not allowed
Notes:	Primary key for this table, uniquely identifies a core record within this table. Automatically assigned by database.
<u>2. CM COLLECTION METHOD ID</u> (FK)	
Physical data type:	NUMBERPS(5,0)
Portable data type:	N-Decimal(5,0)
Allow NULLs:	Not allowed
Notes:	Describes the collection method used to take the core. Foreign key to table of collection methods.
<u>3. PRJ PROJECT ID</u> (FK)	
Physical data type:	NUMBERPS(10,0)
Portable data type:	N-Decimal(10,0)
Allow NULLs:	Not allowed
Notes:	Identifies the project this core is associated with.
<u>4. DRL DRILLER ID</u> (FK)	
Physical data type:	NUMBERPS(10,0)
Portable data type:	N-Decimal(10,0)
Allow NULLs:	Allowed
Notes:	Identifies the driller for this core.
<u>5. COLLECTION DATE</u>	
Physical data type:	DATE
Portable data type:	T-Date & Time
Allow NULLs:	Allowed
Notes:	Date core was drilled.
<u>6. CORE TOP ELEVATION</u>	
Physical data type:	NUMBERPS(8,2)
Portable data type:	N-Decimal(8,2)
Allow NULLs:	Not allowed
Notes:	Elevation of top of core. Negative values in feet below MSL.
<u>7. CORE LENGTH</u>	
Physical data type:	NUMBERPS(8,2)
Portable data type:	N-Decimal(8,2)
Allow NULLs:	Allowed
Notes:	Total length of core in feet.
<u>8. CORE DIAMETER</u>	
Physical data type:	NUMBERPS(8,2)
Portable data type:	N-Decimal(8,2)
Allow NULLs:	Allowed
Notes:	Diameter of core in feet.

DataDictionary

9. X COORD

Physical data type: NUMBERPS(38,8)
Portable data type: N-Decimal(38,8)
Allow NULLs: Not allowed
Notes: X coordinate of source data in coordinate system defined in T_PROJECTS.

10. Y COORD

Physical data type: NUMBERPS(38,8)
Portable data type: N-Decimal(38,8)
Allow NULLs: Not allowed
Notes: Y coordinate of source data in coordinate system defined in T_PROJECTS.

11. STATE X

Physical data type: NUMBERPS(9,2)
Portable data type: N-Decimal(9,2)
Allow NULLs: Allowed
Notes: NAD 1983 Florida Stateplane X coordinate (feet).

12. STATE Y

Physical data type: NUMBERPS(9,2)
Portable data type: N-Decimal(9,2)
Allow NULLs: Allowed
Notes: NAD 1983 Florida Stateplane Y coordinate (feet).

13. STATE_ZONE

Physical data type: VARCHAR2(1)
Portable data type: C-Variable Length(1)
Allow NULLs: Allowed
Check clause: (STATE_ZONE in ('N','E','W'))
Value/Range: 'N', 'E', 'W'.
Notes: The stateplane zone (N, E or W) for this coordinate.

14. LONGITUDE

Physical data type: NUMBERPS(8,6)
Portable data type: N-Decimal(8,6)
Allow NULLs: Allowed
Notes: NAD 1983 longitude from which the core was taken (decimal degrees).

15. LATITUDE

Physical data type: NUMBERPS(8,6)
Portable data type: N-Decimal(8,6)
Allow NULLs: Allowed
Notes: NAD 1983 latitude from which the core was taken (decimal degrees).

16. LORAN X

Physical data type: NUMBERPS(9,2)
Portable data type: N-Decimal(9,2)
Allow NULLs: Allowed
Notes: The loran X coordinate from which the core was taken.

17. LORAN Y

Physical data type: NUMBERPS(9,2)
Portable data type: N-Decimal(9,2)
Allow NULLs: Allowed
Notes: The loran Y coordinate from which the core was taken.

18. PENETRATION DEPTH

Physical data type: NUMBERPS(8,2)
Portable data type: N-Decimal(8,2)
Allow NULLs: Allowed
Notes: Depth drilled during extraction, in feet.

19. RECOVERED LENGTH

Physical data type: NUMBERPS(8,2)
Portable data type: N-Decimal(8,2)
Allow NULLs: Allowed

DataDictionary

Notes: Length of sediment recovered in core, in feet.

20. DIRECTION

Physical data type: VARCHAR2(10)
Portable data type: C-Variable Length(10)
Allow NULLs: Allowed
Notes: Direction of core. Usually vertical.

21. OVERBURDEN

Physical data type: NUMBERPS(5,2)
Portable data type: N-Decimal(5,2)
Allow NULLs: Allowed
Notes: Length of overburden, in feet.

22. DEPTH RX

Physical data type: NUMBERPS(38,10)
Portable data type: N-Decimal(38,10)
Allow NULLs: Allowed
Notes: Depth drilled into rock, in feet.

23. GROUNDWATER ELEVATION

Physical data type: NUMBERPS(8,2)
Portable data type: N-Decimal(8,2)
Allow NULLs: Allowed
Notes: Groundwater elevation, in feet; usually null.

24. PERCENT RECOVERED

Physical data type: NUMBERPS(5,2)
Portable data type: N-Decimal(5,2)
Allow NULLs: Allowed
Notes: Percentage of core recovered ((Recovered_length/penetration depth) * 100).

25. CORE IDENTIFIER

Physical data type: VARCHAR2(25)
Portable data type: C-Variable Length(25)
Allow NULLs: Not allowed
Notes: Project-unique core identifier assigned by agency drilling the core.

26. START DATE

Physical data type: DATE
Portable data type: T-Date & Time
Allow NULLs: Allowed
Notes: Date and time that core drilling began.

27. END DATE

Physical data type: DATE
Portable data type: T-Date & Time
Allow NULLs: Allowed
Notes: Date and time that core drilling ended.

28. GEOLOGIST

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Allowed
Notes: Geologist signature.

29. NUMBER CORE BOXES

Physical data type: NUMBERPS(4,0)
Portable data type: N-Decimal(4,0)
Allow NULLs: Allowed
Notes: Number of core boxes used to store core.

Code details

1. CORE TRIG

Type: Trigger

DataDictionary

Code body:

```
create sequence core_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "CORE_PK_TRG" BEFORE INSERT ON "T_CORES" FOR EACH ROW

BEGIN
  if :NEW.CORE_ID is null then
    SELECT CORE_SEQ.NEXTVAL
    INTO :NEW.CORE_ID
    FROM DUAL;
  end if;
END;
/

create or replace trigger "CORE_TRG" BEFORE INSERT or UPDATE ON "T_CORES" FOR EACH
ROW

BEGIN
  if :NEW.PENETRATION_DEPTH is not null and :NEW.RECOVERED_LENGTH is not null then
    if :NEW.PERCENT_RECOVERED is null then
      :NEW.PERCENT_RECOVERED := ( :NEW.RECOVERED_LENGTH /
:NEW.PENETRATION_DEPTH ) * 100;
    else
      if :NEW.PERCENT_RECOVERED <> ROUND ((:NEW.RECOVERED_LENGTH /
:NEW.PENETRATION_DEPTH ) * 100,2) then
        execute immediate
          'insert into t_import_errors values ("T_CORES",' || :NEW.CORE_ID || ', "Percent recovered
is inconsistent with recovered length and penetration depth.",' || sysdate || ')';
        end if;
      end if;
    end if;
  end if;
END;
/
```

T_DRILLERS

Primary key: DRILLER_ID

Foreign keys	Child	Parent
AGN_DRL_FK	AGN_AGENCY_ID	T_AGENCIES.AGENCY_ID
DRL_COR_FK	T_CORES.DRL_DRILLER_ID	DRILLER_ID

Column details**1. DRILLER_ID**

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Not allowed
 Notes: Uniquely identifies a driller.

2. DRILLER_NAME

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Not allowed
 Notes: Name of driller.

3. DRILL_TYPE

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed
 Notes: Type of drill used by driller.

4. AGN_AGENCY_ID (FK)

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed
 Notes: Agency associated with driller. Foreign key to table of agencies.

Code details**1. DRL_TRG**

Type: Trigger
 Code body:

```

create sequence drl_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "DRL_TRG" BEFORE INSERT ON "T_DRILLERS" FOR EACH ROW

BEGIN
  if :new.driller_id is null then
    SELECT DRL_SEQ.NEXTVAL
    INTO :NEW.DRILLER_ID
    FROM DUAL;
  end if;
END;
/

```

T_GUESTBOOK

Primary key: PK_GUESTBOOK

Column details**1. PK_GUESTBOOK**

Physical data type: NUMBERPS(4,0)
 Portable data type: N-Decimal(4,0)
 Allow NULLs: Not allowed

2. GUEST_NAME

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed

3. GUEST_ORG

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed

4. GUEST_EMAIL

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed

5. GUEST_DATE_VISIT

Physical data type: DATE
 Portable data type: T-Date & Time
 Allow NULLs: Not allowed

6. GUEST_COMMENT

Physical data type: VARCHAR2(2000)
 Portable data type: C-Variable Length(2000)
 Allow NULLs: Allowed

7. GUEST_EMAIL_UPDATE

Physical data type: VARCHAR2(1)
 Portable data type: C-Variable Length(1)
 Allow NULLs: Allowed
 Default value: N

T_IMPORT_ERRORS

Notes: Holds errors associated with data upload routines. Stores the primary key value for the record in the specified table and a brief description of the error condition. Some historical data does not meet some basic database rules that new data must adhere to, but still contains valuable information and should not be discarded. These problems will be noted here. When new data is loaded these errors will also be noted here, with the intention that the problem should be corrected.

Primary key:

1. TABLE_NAME
2. KEY_VALUE

Column details**1. TABLE_NAME**

Physical data type: VARCHAR2(30)
Portable data type: C-Variable Length(30)
Allow NULLs: Not allowed
Notes: Name of table containing the record with the error.

2. KEY_VALUE

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Primary key value of record with the error.

3. ERROR_DESCRIPTION

Physical data type: VARCHAR2(255)
Portable data type: C-Variable Length(255)
Allow NULLs: Not allowed
Notes: Brief description of the error.

4. IMPORT_DATE

Physical data type: DATE
Portable data type: T-Date & Time
Allow NULLs: Not allowed
Notes: The date/time that the import error was raised.

T_LABS

Notes: Laboratories that have performed sample analyses.
Primary key: LAB_ID

Foreign keys	Child	Parent
L_SAMP_FK	T_SAMPLES.LAB_LAB_ID	LAB_ID

Column details**1. LAB_ID**

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a laboratory.

2. LAB_NAME

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Name of laboratory.

3. LAB_ADDRESS

Physical data type: VARCHAR2(150)
Portable data type: C-Variable Length(150)
Allow NULLs: Allowed
Notes: Address of laboratory.

Code details**1. LAB_TRG**

Type: Trigger
Code body:

```

create sequence lab_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "LAB_TRG" BEFORE INSERT ON "T_LABS" FOR EACH ROW

BEGIN
  if :new.lab_id is null then
    SELECT LAB_SEQ.NEXTVAL
    INTO :NEW.LAB_ID
    FROM DUAL;
  end if;
END;
/

```

T_LAYER_STRUCTURES

Notes: Lookup table of layer structure descriptions.
Primary key: LAYER_STRUCTURE_ID

Foreign keys	Child	Parent
CL_LS_FK	T_CORE_LAYERS.LS_LAYER_STRUCTURE_ID	LAYER_STRUCTURE_ID

Column details

1. LAYER_STRUCTURE_ID

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a layer structure description.

2. LAYER_STRUCTURE

Physical data type: VARCHAR2(100)
Portable data type: C-Variable Length(100)
Allow NULLs: Not allowed
Notes: Describes a layer structure.

Code details

1. LS_TRG

Type: Trigger
Code body:

```

create sequence ls_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "LS_TRG" BEFORE INSERT ON "T_LAYER_STRUCTURES" FOR EACH
ROW

BEGIN
  if :new.layer_structure_id is null then
    SELECT LS_SEQ.NEXTVAL
    INTO :NEW.LAYER_STRUCTURE_ID
    FROM DUAL;
  end if;
END;
/

```


T_LITHOLOGY

Notes: Lookup table of sand lithology types.
Primary key: LITHOLOGY_ID

Foreign keys	Child	Parent
L_CLQ_FK	T_CORE_LAYER_QUALIFIERS.L_LITHOLO GY_ID	LITHOLOGY_ID

Column details

1. LITHOLOGY ID

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a lithology type.

2. LITHOLOGY

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Describes a sand lithology type.

T_LOCATIONS

Notes: Spatial layer of both core and grab sample locations. Primary key (OBJECTID) will be added and maintained by SDE. One record will be added to this table for each record in both T_CORES and T_SAMPLES. Either, but not both CORE_CORE_ID or SAMP_SAMPLE_ID will be filled in in this table.

Primary key:

Foreign keys	Child	Parent
CORE_LOC_FK	CORE_CORE_ID	T_CORES.CORE_ID
SAMP_LOC_FK	SAMP_SAMPLE_ID	T_SAMPLES.SAMPLE_ID

Column details

<p>1. OBJECTID (U2) Physical data type: Portable data type: Allow NULLs: Notes:</p>	<p>NUMBERPS(38,0) N-Decimal(38,0) Not allowed SDE maintained unique identifier.</p>
<p>2. CORE_CORE_ID (FK,U1) Physical data type: Portable data type: Allow NULLs: Notes:</p>	<p>NUMBERPS(10,0) N-Decimal(10,0) Allowed Identifies the core at this location. Applicable only to core samples.</p>
<p>3. SAMP_SAMPLE_ID (FK,U3) Physical data type: Portable data type: Allow NULLs: Notes:</p>	<p>NUMBERPS(10,0) N-Decimal(10,0) Allowed Identifies the sample at this location. Applicable only to grab samples.</p>

T_MUNSELL_COLORS

Notes: Stores Munsell colors and color matrix value associated with each color.
Primary key: 1. HUE
 2. VALUE
 3. CHROMA

Foreign keys	Child	Parent
CMTX_MC_FK	CMTX_COLOR_MATRIX_ID	T_COLOR_MATRIX.COLOR_MATRIX_ID

Column details**1. HUE**

Physical data type: VARCHAR2(10)
Portable data type: C-Variable Length(10)
Allow NULLs: Not allowed
Notes: The hue portion of a munsell color code (e.g.: "2.5Y")

2. VALUE

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Not allowed
Notes: The value portion of a munsell color code. Numbers 0 to 10.

3. CHROMA

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Not allowed
Notes: The chroma portion of a munsell color code. Numbers 0 to 20.

4. CMTX_COLOR_MATRIX_ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: The color matrix for this munsell color. Foreign key to table of color matrices.

T_PROJECTS

Notes: Sand search projects.
 Primary key: PROJECT_ID

Foreign keys	Child	Parent
AGN_PRJ_MANAGING_FK	AGN_AGENCY_ID_MANAGING	T_AGENCIES.AGENCY_ID
AGN_PRJ_POSSESSING_FK	AGN_AGENCY_ID_POSSESSING	T_AGENCIES.AGENCY_ID
CON_PRJ_FK	CON_CONTACT_ID	T_CONTACTS.CONTACT_ID
PRJ_SAMP_FK	T_SAMPLES.PRJ_PROJECT_ID	PROJECT_ID
PRJ_CORE_FK	T_CORES.PRJ_PROJECT_ID	PROJECT_ID

Column details**1. PROJECT_ID**

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Not allowed
 Notes: System-assigned unique identifier of a project.

2. AGN_AGENCY_ID_POSSESSING (FK)

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed
 Notes: Identifies the agency that possesses the data collected by the project.

3. AGN_AGENCY_ID_MANAGING (FK)

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed
 Notes: Identifies the agency that managed the project.

4. CON_CONTACT_ID (FK)

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed
 Notes: Identifies the primary point of contact for the project.

5. PROJECT_NAME

Physical data type: VARCHAR2(200)
 Portable data type: C-Variable Length(200)
 Allow NULLs: Allowed
 Notes: The name of the project.

6. PROJECT_DATE

Physical data type: DATE
 Portable data type: T-Date & Time
 Allow NULLs: Allowed
 Notes: The date the project was established.

7. PROJECT_LOCATION

Physical data type: VARCHAR2(200)
 Portable data type: C-Variable Length(200)
 Allow NULLs: Allowed
 Notes: General location of the project; a geographic description such as "Florida Panhandle".

8. HORIZONTAL_COORDINATE_SYSTEM

Physical data type: VARCHAR2(50)
 Portable data type: C-Variable Length(50)
 Allow NULLs: Allowed
 Notes: The horizontal coordinate system (e.g. "UTM", "Geographic", "State Plane Florida East")

9. HORIZONTAL_DATUM

Physical data type: VARCHAR2(15)
 Portable data type: C-Variable Length(15)

DataDictionary

Allow NULLs:

Allowed

Notes:

The horizontal datum used for geographic coordinates associated with this project. (e.g.: "NAD 1983")

10. VERTICAL DATUM

Physical data type:

VARCHAR2(15)

Portable data type:

C-Variable Length(15)

Allow NULLs:

Allowed

Notes:

The vertical datum used for depth/elevation measurements for this project.

11. PROJECTION

Physical data type:

VARCHAR2(50)

Portable data type:

C-Variable Length(50)

Allow NULLs:

Allowed

Notes:

???

T_QUALIFIERS_LU

Notes: Lookup table of core layer qualifier types.
Primary key: QUALIFIER_ID

Foreign keys	Child	Parent
QLU_CLQ_FK	T_CORE_LAYER_QUALIFIERS.QLU_QUAL IFIER_ID	QUALIFIER_ID

Column details**1. QUALIFIER_ID**

Physical data type: VARCHAR2(25)
Portable data type: C-Variable Length(25)
Allow NULLs: Not allowed
Notes: Qualifier unique identifier and name.

2. DESCRIPTION

Physical data type: VARCHAR2(250)
Portable data type: C-Variable Length(250)
Allow NULLs: Not allowed
Notes: Describes a qualifier in more detail.

T_SAMPLES

Notes: Sand samples, both grab and core.
 Primary key: SAMPLE_ID

Foreign keys	Child	Parent
PRJ_SAMP_FK	PRJ_PROJECT_ID	T_PROJECTS.PROJECT_ID
CMTX_SAMP_FK	CMTX_COLOR_MATRIX_ID	T_COLOR_MATRIX.COLOR_MATRIX_ID
L_SAMP_FK	LAB_LAB_ID	T_LABS.LAB_ID
SLU_SAMP_FK	SLU_SPHERICITY_ID	T_SPHERICITY_LU.SPHERICITY_ID
ALU_SAMP_FK	ALU_ANGULARITY_ID	T_ANGULARITY_LU.ANGULARITY_ID
CALC_SAMP_MEAN_FK	CALC_CALC_METHOD_ID_MEAN	T_CALCULATION_METHODS.CALCULATION_METHOD_ID
CALC_SAMP_MEDIAN_FK	CALC_CALC_METHOD_ID_MEDIAN	T_CALCULATION_METHODS.CALCULATION_METHOD_ID
CALC_SAMP_STD_FK	CALC_CALC_METHOD_ID_STD	T_CALCULATION_METHODS.CALCULATION_METHOD_ID
CALC_SAMP_SKEWNESS_FK	CALC_CALC_METHOD_ID_SKEW	T_CALCULATION_METHODS.CALCULATION_METHOD_ID
CALC_SAMP_KURTOSIS_FK	CALC_CALC_METHOD_ID_KURT	T_CALCULATION_METHODS.CALCULATION_METHOD_ID
USCS_SAMP_FK	USCS_USCS_CLASSIFICATION_ID	T_USCS_CLASSIFICATIONS.USCS_CLASSIFICATION_ID
CM_SAMP_FK	CM_COLLECTION_METHOD_ID	T_COLLECTION_METHODS.COLLECTION_METHOD_ID
AM_SAMP_FK	AM_ANALYTICAL_METHOD_ID	T_ANALYTICAL_METHODS.ANALYTICAL_METHOD_ID
CORE_SAMP_FK	CORE_CORE_ID	T_CORES.CORE_ID
SAMP_SXC_FK	T_SAMPLES_X_CORELAYERS.SAMP_SAMPLE_ID	SAMPLE_ID
SAMP_LOC_FK	T_LOCATIONS.SAMP_SAMPLE_ID	SAMPLE_ID

Column details**1. SAMPLE_ID**

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Not allowed
 Notes: Uniquely identifies a sample record

2. PRJ_PROJECT_ID (FK)

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Not allowed
 Notes: Identifies the project this sample is from. Foreign key to table of projects.

3. CORE_CORE_ID (FK)

Physical data type: NUMBERPS(10,0)
 Portable data type: N-Decimal(10,0)
 Allow NULLs: Allowed
 Notes: Identifies the core that this sample was taken from, bypassing the association to T_CORE_LAYERS. Applicable for samples taken from cores only.

4. LAB_LAB_ID (FK)

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed
 Notes: Identifies the lab that performed the analysis. Foreign key to labs table.

5. AM_ANALYTICAL_METHOD_ID (FK)

Physical data type: NUMBERPS(5,0)
 Portable data type: N-Decimal(5,0)
 Allow NULLs: Allowed
 Notes: Describes the method used for analyzing the sample. Foreign key to table of analytical methods.

DataDictionary

6. SLU SPHERICITY ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Describes sphericity. Foreign key to table of sphericity.

7. ALU ANGULARITY ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Describes angularity. Foreign key to table of angularity.

8. CM COLLECTION METHOD ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Identifies how the sample was collected. Foreign key to table of collection methods.

9. SAMPLE IDENTIFIER

Physical data type: VARCHAR2(20)
Portable data type: C-Variable Length(20)
Allow NULLs: Not allowed
Notes: Project-unique sample identifier assigned by agency collecting sample. Does not uniquely identify records within this table.

10. SAMPLE DATE

Physical data type: DATE
Portable data type: T-Date & Time
Allow NULLs: Not allowed
Notes: Date sample was taken.

11. SAMPLE COMMENTS

Physical data type: VARCHAR2(4000)
Portable data type: C-Variable Length(4000)
Allow NULLs: Allowed
Notes: Sample comments recorded by agency taking sample.

12. USCS USCS CLASSIFICATION ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Identifies the unified soils classification system soil type. Foreign key to table of USCS_CLASSIFICATIONS.

13. CMTX COLOR MATRIX ID (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Describes the color matrix.

14. MUNSELL HUE DRY

Physical data type: VARCHAR2(6)
Portable data type: C-Variable Length(6)
Allow NULLs: Allowed
Notes: The hue portion of the Munsell color code for this sample when dry

15. MUNSELL VALUE DRY

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: The value portion of the Munsell color code for this sample when dry

16. MUNSELL CHROMA DRY

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed

DataDictionary

Notes:

The chroma portion of the Munsell color code for this sample when dry

17. MUNSELL HUE WET

Physical data type:

VARCHAR2(6)

Portable data type:

C-Variable Length(6)

Allow NULLs:

Allowed

Notes:

The hue portion of the Munsell color code for this sample when wet

18. MUNSELL VALUE WET

Physical data type:

NUMBERPS(4,2)

Portable data type:

N-Decimal(4,2)

Allow NULLs:

Allowed

Notes:

The value portion of the Munsell color code for this sample when wet

19. MUNSELL CHROMA WET

Physical data type:

NUMBERPS(4,2)

Portable data type:

N-Decimal(4,2)

Allow NULLs:

Allowed

Notes:

The chroma portion of the Munsell color code for this sample when wet

20. MUNSELL HUE WASHED

Physical data type:

VARCHAR2(6)

Portable data type:

C-Variable Length(6)

Allow NULLs:

Allowed

Notes:

The hue portion of the Munsell color code for this sample when washed

21. MUNSELL VALUE WASHED

Physical data type:

NUMBERPS(4,2)

Portable data type:

N-Decimal(4,2)

Allow NULLs:

Allowed

Notes:

The value portion of the Munsell color code for this sample when washed

22. MUNSELL CHROMA WASHED

Physical data type:

NUMBERPS(4,2)

Portable data type:

N-Decimal(4,2)

Allow NULLs:

Allowed

Notes:

The chroma portion of the Munsell color code for this sample when washed

23. MUNSELL HUE UNKNOWN

Physical data type:

VARCHAR2(6)

Portable data type:

C-Variable Length(6)

Allow NULLs:

Allowed

Notes:

The hue portion of the Munsell color code for this sample when wet/washed/dry is unknown

24. MUNSELL VALUE UNKNOWN

Physical data type:

NUMBERPS(4,2)

Portable data type:

N-Decimal(4,2)

Allow NULLs:

Allowed

Notes:

The value portion of the Munsell color code for this sample when wet/washed/dry is unknown

25. MUNSELL CHROMA UNKNOWN

Physical data type:

NUMBERPS(4,2)

Portable data type:

N-Decimal(4,2)

Allow NULLs:

Allowed

Notes:

The chroma portion of the Munsell color code for this sample when wet/washed/dry is unknown

26. ANALYSIS DATE

Physical data type:

DATE

Portable data type:

T-Date & Time

Allow NULLs:

Allowed

Notes:

Data sample was analyzed, if known.

27. LAB REMARKS

Physical data type:

VARCHAR2(4000)

Portable data type:

C-Variable Length(4000)

Allow NULLs:

Allowed

DataDictionary

Notes:

Sample comments recorded by laboratory performing analysis.

28. X COORD

Physical data type:

NUMBERPS(38,8)

Portable data type:

N-Decimal(38,8)

Allow NULLs:

Allowed

Notes:

X coordinate of source data in coordinate system defined in T_PROJECTS.

29. Y COORD

Physical data type:

NUMBERPS(38,8)

Portable data type:

N-Decimal(38,8)

Allow NULLs:

Allowed

Notes:

X coordinate of source data in coordinate system defined in T_PROJECTS.

30. STATE X

Physical data type:

NUMBERPS(9,2)

Portable data type:

N-Decimal(9,2)

Allow NULLs:

Allowed

Notes:

NAD 1983 Florida Stateplane X coordinate (feet).

31. STATE Y

Physical data type:

NUMBERPS(9,2)

Portable data type:

N-Decimal(9,2)

Allow NULLs:

Allowed

Notes:

NAD 1983 Florida Stateplane Y coordinate (feet).

32. STATE_ZONE

Physical data type:

VARCHAR2(1)

Portable data type:

C-Variable Length(1)

Allow NULLs:

Allowed

Check clause:

(STATE_ZONE in ('E','W','N'))

Value/Range:

'E', 'W', 'N'.

Notes:

The stateplane zone (N, E or W) for this coordinate.

33. LORAN X

Physical data type:

NUMBERPS(9,2)

Portable data type:

N-Decimal(9,2)

Allow NULLs:

Allowed

Notes:

The loran X coordinate from which the sample was taken.

34. LORAN Y

Physical data type:

NUMBERPS(9,2)

Portable data type:

N-Decimal(9,2)

Allow NULLs:

Allowed

Notes:

The loran Y coordinate from which the sample was taken.

35. LONGITUDE

Physical data type:

NUMBERPS(8,6)

Portable data type:

N-Decimal(8,6)

Allow NULLs:

Allowed

Notes:

NAD 1983 longitude from which the sample was taken (decimal degrees).

36. LATITUDE

Physical data type:

NUMBERPS(8,6)

Portable data type:

N-Decimal(8,6)

Allow NULLs:

Allowed

Notes:

NAD 1983 latitude from which the sample was taken (decimal degrees).

37. RANGE MONUMENT

Physical data type:

VARCHAR2(10)

Portable data type:

C-Variable Length(10)

Allow NULLs:

Allowed

Notes:

Range monument relative to which the sample location is located.

38. RM TRANSECT LOCATION

Physical data type:

VARCHAR2(10)

DataDictionary

Portable data type: C-Variable Length(10)
Allow NULLs: Allowed
Notes: Transect location of the sample, relative to the range monument.

39. TOP OF SAMPLE INTERVAL

Physical data type: NUMBERPS(8,3)
Portable data type: N-Decimal(8,3)
Allow NULLs: Allowed
Check clause: (TOP_OF_SAMPLE_INTERVAL >= 0)
Value/Range: 0..*
Notes: Beginning measurement within a core for a specific sample interval. Not applicable for grab samples, required for core samples. Positive values in feet.

40. BOTTOM OF SAMPLE INTERVAL

Physical data type: NUMBERPS(8,3)
Portable data type: N-Decimal(8,3)
Allow NULLs: Allowed
Notes: Ending measurement within a core for a specific sample interval. Not applicable for grab samples, required for core samples. Positive values in feet, greater than or equal to TOP_OF_SAMPLE_INTERVAL.

41. GRAB ELEVATION

Physical data type: NUMBERPS(8,3)
Portable data type: N-Decimal(8,3)
Allow NULLs: Allowed
Check clause: (GRAB_ELEVATION <= 0)
Value/Range: *.0
Notes: Elevation of grab samples only, not applicable for core samples. Negative values in feet below mean sea level

42. MEAN

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Mean grain size in phi units, calculated using the Moment method.

43. MEDIAN

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Median grain size in phi units, calculated using the Moment method.

44. STD

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Standard deviation of mean grain size, calculated using the Moment method.

45. SKEWNESS

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Defines the symmetry of the grain size distribution, calculated using the Moment method.

46. KURTOSIS

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: A measure of the relative peakedness of the distribution of grain sizes, calculated using the Moment method.

47. MEAN ORIGINAL

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Mean grain size in phi units, calculated using the method identified in

DataDictionary

CM_CALC_METHOD_ID_MEAN.

48. MEDIAN ORIGINAL

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Mean grain size in phi units, calculated using the method identified in CM_CALC_METHOD_ID_MEDIAN.

49. STD ORIGINAL

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Standard deviation of mean grain size, calculated using the method identified in CM_CALC_METHOD_ID_MEAN.

50. SKEWNESS ORIGINAL

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: Defines the symmetry of the grain size distribution, calculated using the method identified in CM_CALC_METHOD_ID_MEAN.

51. KURTOSIS ORIGINAL

Physical data type: NUMBERPS(4,2)
Portable data type: N-Decimal(4,2)
Allow NULLs: Allowed
Notes: A measure of the relative peakedness of the distribution of grain size, calculated using the method identified in CM_CALC_METHOD_ID_MEAN.

52. CALC CALC METHOD ID MEAN (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Calculation method used to determine the mean. Foreign key to table of calculation methods. Usually all calculation methods will be identical, but some historic data may have different methods used for mean/median/etc.

53. CALC CALC METHOD ID MEDIAN (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Calculation method used to determine the median. Foreign key to table of Calculation methods. Usually all calculation methods will be identical, but some historic data may have different methods used for mean/median/etc.

54. CALC CALC METHOD ID STD (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Calculation method used to determine the standard deviation. Foreign key to table of Calculation methods. Usually all calculation methods will be identical, but some historic data may have different methods used for mean/median/etc.

55. CALC CALC METHOD ID SKEW (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed
Notes: Calculation method used to determine the skewness. Foreign key to table of Calculation methods. Usually all calculation methods will be identical, but some historic data may have different methods used for mean/median/etc.

56. CALC CALC METHOD ID KURT (FK)

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Allowed

DataDictionary

Notes:

Calculation method used to determine the kurtosis. Foreign key to table of Calculation methods. Usually all calculation methods will be identical, but some historic data may have different methods used for mean/median/etc.

57. PCT FINES

Physical data type:

NUMBERPS(5,2)

Portable data type:

N-Decimal(5,2)

Allow NULLs:

Allowed

Notes:

Percentage of sample, by weight, of fines (particles < 4.5 phi).

58. PCT PAN FRACTION

Physical data type:

NUMBERPS(5,2)

Portable data type:

N-Decimal(5,2)

Allow NULLs:

Allowed

Notes:

Percentage of sample, by weight, of remaining material collected in basal sieve nest pan.

59. PCT CARBONATE

Physical data type:

NUMBERPS(5,2)

Portable data type:

N-Decimal(5,2)

Allow NULLs:

Allowed

Notes:

Percentage of carbonate in sample. by weight?

60. PCT SHELL FRAGMENTS

Physical data type:

NUMBERPS(5,2)

Portable data type:

N-Decimal(5,2)

Allow NULLs:

Allowed

Notes:

Percentage of shell fragments in sample. by weight?

61. PCT HEAVY MINERALS

Physical data type:

NUMBERPS(5,2)

Portable data type:

N-Decimal(5,2)

Allow NULLs:

Allowed

Notes:

Percentage of heavy minerals content in sample. by weight?

62. PCT ORGANICS

Physical data type:

NUMBERPS(5,2)

Portable data type:

N-Decimal(5,2)

Allow NULLs:

Allowed

Notes:

Percentage of organic content in sample. by weight?

63. CARBONATE DISSOLVED

Physical data type:

VARCHAR2(1)

Portable data type:

C-Variable Length(1)

Allow NULLs:

Allowed

Check clause:

(CARBONATE DISSOLVED in ('Y','N'))

Value/Range:

'Y'.

Notes:

Flag whether carbonate was dissolved before analysis. Values "Y", "N" or null.

64. HEAVY MINERALS DISSOLVED

Physical data type:

VARCHAR2(1)

Portable data type:

C-Variable Length(1)

Allow NULLs:

Allowed

Check clause:

(HEAVY_MINERALS DISSOLVED in ('Y','N'))

Value/Range:

'Y'.

Notes:

Flag whether heavy minerals were removed before analysis. Values "Y", "N" or null.

65. ORGANICS REMOVED

Physical data type:

VARCHAR2(1)

Portable data type:

C-Variable Length(1)

Allow NULLs:

Allowed

Check clause:

(ORGANICS_REMOVED in ('Y','N'))

Value/Range:

'Y'.

Notes:

Flag whether organics were removed before analysis. Values "Y", "N" or null

66. SHELL FRAGMENTS REMOVED

Physical data type:

VARCHAR2(1)

DataDictionary

Portable data type: C-Variable Length(1)
Allow NULLs: Allowed
Check clause: (SHELL_FRAGMENTS_REMOVED in ('Y','N'))
Value/Range: 'Y'.
Notes: Flag whether shell fragments were removed before analysis. Values "Y", "N" or null.

67. PHI N 12

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent retained of sample of grain size -12 phi.

68. PHI N 10

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent retained of sample of grain size -10 phi.

69. PHI N 08

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -8 phi.

70. PHI N 07

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -7 phi.

71. PHI N 06 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -6.75 phi.

72. PHI N 06 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -6.5 phi.

73. PHI N 06 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -6.25 phi.

74. PHI N 06

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -6 phi.

75. PHI N 05 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -5.75 phi.

76. PHI N 05 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -5.5 phi.

DataDictionary

77. PHI N 05 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -5.25 phi.

78. PHI N 05

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -5 phi.

79. PHI N 04 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -4.75 phi.

80. PHI N 04 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -4.5 phi.

81. PHI N 04 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -4.25 phi.

82. PHI N 04

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -4 phi.

83. PHI N 03 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -3.75 phi.

84. PHI N 03 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -3.5 phi.

85. PHI N 03 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -3.25 phi.

86. PHI N 03

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -3 phi.

87. PHI N 02 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -2.75 phi.

DataDictionary

88. PHI N 02 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -2.5 phi.

89. PHI N 02 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -2.25 phi.

90. PHI N 02

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -2 phi.

91. PHI N 01 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -1.75 phi.

92. PHI N 01 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -1.5 phi.

93. PHI N 01 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -1.25 phi.

94. PHI N 01

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -1 phi.

95. PHI N 00 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -0.75 phi.

96. PHI N 00 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -0.5 phi.

97. PHI N 00 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size -0.25 phi.

98. PHI 00

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 0 phi.

DataDictionary

99. PHI 00 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 0.25 phi.

100. PHI 00 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 0.5 phi.

101. PHI 00 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 0.75 phi.

102. PHI 01

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 1.0 phi.

103. PHI 01 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 1.25 phi.

104. PHI 01 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 1.5 phi.

105. PHI 01 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 1.75 phi.

106. PHI 02

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 2.0 phi.

107. PHI 02 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 2.25 phi.

108. PHI 02 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 2.5 phi.

109. PHI 02 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 2.75 phi.

DataDictionary

110. PHI 03

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 3.0 phi.

111. PHI 03 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 3.25 phi.

112. PHI 03 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 3.5 phi.

113. PHI 03 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 3.75 phi.

114. PHI 04

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 4.0 phi.

115. PHI 04 25

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 4..25 phi.

116. PHI 04 50

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 4.5 phi.

117. PHI 04 75

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 4.75 phi.

118. PHI 05

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 5 phi.

119. PHI 06

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 6 phi.

120. PHI 07

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 7 phi.

DataDictionary

121. PHI 08

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 8 phi.

122. PHI 09

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 9 phi.

123. PHI 10

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 10 phi.

124. PHI 11

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 11 phi.

125. PHI 12

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 12 phi.

126. PHI 13

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 13 phi.

127. PHI 14

Physical data type: NUMBERPS(5,3)
Portable data type: N-Decimal(5,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size 14 phi.

128. USCS COBBLE

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the USCS cobble range. Automatically populated by trigger on insert, used for querying.

129. USCS COARSE GRAVEL

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the USCS coarse gravel range. Automatically populated by trigger on insert, used for querying.

130. USCS FINE GRAVEL

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the USCS fine gravel range. Automatically populated by trigger on insert, used for querying.

131. USCS COARSE SAND

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)

DataDictionary

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the USCS coarse sand range. Automatically populated by trigger on insert, used for querying.

132. USCS MEDIUM SAND

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the USCS medium sand range. Automatically populated by trigger on insert, used for querying.

133. USCS FINE SAND

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the USCS fine sand range. Automatically populated by trigger on insert, used for querying.

134. USCS SILT

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the USCS silt range. Automatically populated by trigger on insert, used for querying.

135. USCS CLAY

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the USCS clay range. Automatically populated by trigger on insert, used for querying.

136. WW BOULDER

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the Wentworth boulder range. Automatically populated by trigger on insert, used for querying.

137. WW COBBLE

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the Wentworth cobble range. Automatically populated by trigger on insert, used for querying.

138. WW GRAVEL

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the Wentworth gravel range. Automatically populated by trigger on insert, used for querying.

139. WW PEBBLE

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the Wentworth pebble range. Automatically populated by trigger on insert, used for querying.

140. WW VERY COARSE SAND

Physical data type:

NUMBERPS(6,3)

Portable data type:

N-Decimal(6,3)

Allow NULLs:

Allowed

Notes:

Weight percent of sample of grain size in the Wentworth very coarse sand range. Automatically populated by trigger on insert, used for querying.

DataDictionary

141. WW COARSE SAND

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth coarse sand range. Automatically populated by trigger on insert, used for querying.

142. WW MEDIUM SAND

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth medium sand range. Automatically populated by trigger on insert, used for querying.

143. WW FINE SAND

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth fine sand range. Automatically populated by trigger on insert, used for querying.

144. WW VERY FINE SAND

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth very fine sand range. Automatically populated by trigger on insert, used for querying.

145. WW SILT

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth silt range. Automatically populated by trigger on insert, used for querying.

146. WW CLAY

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth clay range. Automatically populated by trigger on insert, used for querying.

147. WW COLLOID

Physical data type: NUMBERPS(6,3)
Portable data type: N-Decimal(6,3)
Allow NULLs: Allowed
Notes: Weight percent of sample of grain size in the Wentworth colloid range. Automatically populated by trigger on insert, used for querying.

Code details

1. SAMP TRG

Type: Trigger
Code body:

```
create sequence samp_seq INCREMENT BY 1 START WITH 1
    MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
    NOCACHE NOORDER;

create or replace trigger "SAMP_PK_TRG"
BEFORE INSERT ON "T_SAMPLES"
FOR EACH ROW
BEGIN
    if :NEW.SAMPLE_ID is null then
        SELECT SAMP_SEQ.NEXTVAL
        INTO :NEW.SAMPLE_ID
        FROM DUAL;
```

```

        end if;
    END;
/

CREATE OR REPLACE TRIGGER "SAMP_TRG"
BEFORE INSERT OR UPDATE ON "T_SAMPLES"
FOR EACH ROW
declare
    vNum number;
    strMsg varchar2(255);
    exc exception;
    C_PROJECT_ID number;
begin
    -- Make sure that data matches between samples and cores.
    if :new.core_core_id is not null then
        select prj_project_id into c_project_id
            from t_cores
            where core_id = :new.core_core_id;
        if c_project_id <> :new.prj_project_id then
            strMsg := 'Mismatched project ids in T_SAMPLES and T_CORES';
            raise exc;
        end if;
    end if;

    -- The following block will not raise an exception, but instead populate
    -- the T_IMPORT_ERRORS table.
    strMsg := '';
    -- Require grab sample elevation if it is a grab sample (4).
    if :new.CM_COLLECTION_METHOD_ID = 4 then
        IF :new.grab_elevation is null then
            strMsg := 'Grab sample elevation required. ';
        end if;
    else
        -- Require top < bottom for core samples.
        if :new.top_of_sample_interval > :new.bottom_of_sample_interval then
            strMsg := 'Top interval must be less than or equal to bottom interval. ';
        end if;
    end if;

    -- Validate PHI breakdowns
    vNum := NVL(:NEW.PHI_N_12,0) + NVL(:NEW.PHI_N_10,0) + NVL(:NEW.PHI_N_08,0) +
    NVL(:NEW.PHI_N_07,0) + NVL(:NEW.PHI_N_06_75,0) +
    NVL(:NEW.PHI_N_06_50,0) + NVL(:NEW.PHI_N_06_25,0) + NVL(:NEW.PHI_N_06,0) +
    NVL(:NEW.PHI_N_05_75,0) + NVL(:NEW.PHI_N_05_50,0) +
    NVL(:NEW.PHI_N_05_25,0) + NVL(:NEW.PHI_N_05,0) + NVL(:NEW.PHI_N_04_75,0) +
    NVL(:NEW.PHI_N_04_50,0) + NVL(:NEW.PHI_N_04_25,0) +
    NVL(:NEW.PHI_N_04,0) + NVL(:NEW.PHI_N_03_75,0) + NVL(:NEW.PHI_N_03_50,0) +
    NVL(:NEW.PHI_N_03_25,0) + NVL(:NEW.PHI_N_03,0) +
    NVL(:NEW.PHI_N_02_75,0) + NVL(:NEW.PHI_N_02_50,0) + NVL(:NEW.PHI_N_02_25,0) +
    NVL(:NEW.PHI_N_02,0) + NVL(:NEW.PHI_N_01_75,0) +
    NVL(:NEW.PHI_N_01_50,0) + NVL(:NEW.PHI_N_01_25,0) + NVL(:NEW.PHI_N_01,0) +
    NVL(:NEW.PHI_N_00_75,0) + NVL(:NEW.PHI_N_00_50,0) +
    NVL(:NEW.PHI_N_00_25,0) + NVL(:NEW.PHI_00,0) + NVL(:NEW.PHI_00_25,0) +
    NVL(:NEW.PHI_00_50,0) + NVL(:NEW.PHI_00_75,0) +
    NVL(:NEW.PHI_01,0) + NVL(:NEW.PHI_01_25,0) + NVL(:NEW.PHI_01_50,0) +
    NVL(:NEW.PHI_01_75,0) + NVL(:NEW.PHI_02,0) +
    NVL(:NEW.PHI_02_25,0) + NVL(:NEW.PHI_02_50,0) + NVL(:NEW.PHI_02_75,0) +
    NVL(:NEW.PHI_03,0) + NVL(:NEW.PHI_03_25,0) +
    NVL(:NEW.PHI_03_50,0) + NVL(:NEW.PHI_03_75,0) + NVL(:NEW.PHI_04,0) +
    NVL(:NEW.PHI_04_25,0) + NVL(:NEW.PHI_04_50,0) +
    NVL(:NEW.PHI_04_75,0) + NVL(:NEW.PHI_05,0) + NVL(:NEW.PHI_06,0) +
    NVL(:NEW.PHI_07,0) + NVL(:NEW.PHI_08,0) + NVL(:NEW.PHI_09,0) +
    NVL(:NEW.PHI_10,0) + NVL(:NEW.PHI_11,0) + NVL(:NEW.PHI_12,0) +
    NVL(:NEW.PHI_13,0) + NVL(:NEW.PHI_14,0) +
    NVL(:NEW.PCT_FINES,0) + NVL(:NEW.PCT_PAN_FRACTION,0);
    if vNum < 99 or vNum > 101 then

```

DataDictionary

```
    strMsg := strMsg || 'Sum of weight percentages (sum of all phi_*_* columns + pct_pan_fraction +
pct_fines) is invalid (' || to_char(vNum) ||)';
    end if;

    if strMsg <> " then
        execute immediate 'insert into t_import_errors values ("T_SAMPLES",' || :new.sample_id ||',' ||
strMsg || ',' || sysdate)';
    end if;

    -- Calculate USCS and Wentworth Classifications
    :NEW.USCS_COBBLE := ROUND(NVL(:NEW.PHI_N_12,0) + NVL(:NEW.PHI_N_10,0) +
NVL(:NEW.PHI_N_08,0) + NVL(:NEW.PHI_N_07,0) + NVL(:NEW.PHI_N_06_75,0) +
NVL(:NEW.PHI_N_06_50,0) + NVL(:NEW.PHI_N_06_25,0), 3);
    :NEW.USCS_COARSE_GRAVEL := ROUND(NVL(:NEW.PHI_N_06,0) +
NVL(:NEW.PHI_N_05_75,0) + NVL(:NEW.PHI_N_05_50,0) + NVL(:NEW.PHI_N_05_25,0) +
NVL(:NEW.PHI_N_05,0) + NVL(:NEW.PHI_N_04_75,0) + NVL(:NEW.PHI_N_04_50,0) +
NVL(:NEW.PHI_N_04_25,0), 3);
    :NEW.USCS_FINE_GRAVEL := ROUND(NVL(:NEW.PHI_N_04,0) +
NVL(:NEW.PHI_N_03_75,0) + NVL(:NEW.PHI_N_03_50,0) + NVL(:NEW.PHI_N_03_25,0) +
NVL(:NEW.PHI_N_03,0) + NVL(:NEW.PHI_N_02_75,0) + NVL(:NEW.PHI_N_02_50,0) +
NVL(:NEW.PHI_N_02_25,0), 3);
    :NEW.USCS_COARSE_SAND := ROUND(NVL(:NEW.PHI_N_02,0) +
NVL(:NEW.PHI_N_01_75,0) + NVL(:NEW.PHI_N_01_50,0) + NVL(:NEW.PHI_N_01_25,0) +
NVL(:NEW.PHI_N_01,0), 3);
    :NEW.USCS_MEDIUM_SAND := ROUND(NVL(:NEW.PHI_N_00_75,0) +
NVL(:NEW.PHI_N_00_50,0) + NVL(:NEW.PHI_N_00_25,0) + NVL(:NEW.PHI_00,0) +
NVL(:NEW.PHI_00_25,0) + NVL(:NEW.PHI_00_50,0) + NVL(:NEW.PHI_00_75,0) +
NVL(:NEW.PHI_01,0) + NVL(:NEW.PHI_01_25,0), 3);
    :NEW.USCS_FINE_SAND := ROUND(NVL(:NEW.PHI_01_50,0) + NVL(:NEW.PHI_01_75,0) +
NVL(:NEW.PHI_02,0) + NVL(:NEW.PHI_02_25,0) + NVL(:NEW.PHI_02_50,0) +
NVL(:NEW.PHI_02_75,0) + NVL(:NEW.PHI_03,0) + NVL(:NEW.PHI_03_25,0) +
NVL(:NEW.PHI_03_50,0) + NVL(:NEW.PHI_03_75,0), 3);
    :NEW.USCS_SILT := ROUND(NVL(:NEW.PHI_04,0) + NVL(:NEW.PHI_04_25,0) +
NVL(:NEW.PHI_04_50,0) + NVL(:NEW.PHI_04_75,0) + NVL(:NEW.PHI_05,0) +
NVL(:NEW.PHI_06,0) + NVL(:NEW.PHI_07,0) + NVL(:NEW.PHI_08,0), 3);
    :NEW.USCS_CLAY := ROUND(NVL(:NEW.PHI_09,0) + NVL(:NEW.PHI_10,0) +
NVL(:NEW.PHI_11,0) + NVL(:NEW.PHI_12,0) + NVL(:NEW.PHI_13,0) + NVL(:NEW.PHI_14,0),
3);
    :NEW.WW_BOULDER := ROUND(NVL(:NEW.PHI_N_12,0) + NVL(:NEW.PHI_N_10,0) +
NVL(:NEW.PHI_N_08,0), 3);
    :NEW.WW_COBBLE := ROUND(NVL(:NEW.PHI_N_07,0) + NVL(:NEW.PHI_N_06_75,0) +
NVL(:NEW.PHI_N_06_50,0) + NVL(:NEW.PHI_N_06_25,0) + NVL(:NEW.PHI_N_06,0), 3);
    :NEW.WW_PEBBLE := ROUND(NVL(:NEW.PHI_N_05_75,0) + NVL(:NEW.PHI_N_05_50,0) +
NVL(:NEW.PHI_N_05_25,0) + NVL(:NEW.PHI_N_05,0) + NVL(:NEW.PHI_N_04_75,0) +
NVL(:NEW.PHI_N_04_50,0) + NVL(:NEW.PHI_N_04_25,0) + NVL(:NEW.PHI_N_04,0) +
NVL(:NEW.PHI_N_03_75,0) + NVL(:NEW.PHI_N_03_50,0) + NVL(:NEW.PHI_N_03_25,0) +
NVL(:NEW.PHI_N_03,0) + NVL(:NEW.PHI_N_02_75,0) + NVL(:NEW.PHI_N_02_50,0) +
NVL(:NEW.PHI_N_02_25,0) + NVL(:NEW.PHI_N_02,0), 3);
    :NEW.WW_GRAVEL := ROUND(NVL(:NEW.PHI_N_01_75,0) + NVL(:NEW.PHI_N_01_50,0) +
NVL(:NEW.PHI_N_01_25,0) + NVL(:NEW.PHI_N_01,0), 3);
    :NEW.WW_VERY_COARSE_SAND := ROUND(NVL(:NEW.PHI_N_00_75,0) +
NVL(:NEW.PHI_N_00_50,0) + NVL(:NEW.PHI_N_00_25,0) + NVL(:NEW.PHI_00,0), 3);
    :NEW.WW_COARSE_SAND := ROUND(NVL(:NEW.PHI_00_25,0) + NVL(:NEW.PHI_00_50,0)
+ NVL(:NEW.PHI_00_75,0) + NVL(:NEW.PHI_01,0), 3);
    :NEW.WW_MEDIUM_SAND := ROUND(NVL(:NEW.PHI_01_25,0) + NVL(:NEW.PHI_01_50,0)
+ NVL(:NEW.PHI_01_75,0) + NVL(:NEW.PHI_02,0), 3);
    :NEW.WW_FINE_SAND := ROUND(NVL(:NEW.PHI_02_25,0) + NVL(:NEW.PHI_02_50,0) +
NVL(:NEW.PHI_02_75,0) + NVL(:NEW.PHI_03,0), 3);
    :NEW.WW_VERY_FINE_SAND := ROUND(NVL(:NEW.PHI_03_25,0) +
NVL(:NEW.PHI_03_50,0) + NVL(:NEW.PHI_03_75,0) + NVL(:NEW.PHI_04,0), 3);
    :NEW.WW_SILT := ROUND(NVL(:NEW.PHI_04_25,0) + NVL(:NEW.PHI_04_50,0) +
NVL(:NEW.PHI_04_75,0) + NVL(:NEW.PHI_05,0) + NVL(:NEW.PHI_06,0) +
NVL(:NEW.PHI_07,0) + NVL(:NEW.PHI_08,0), 3);
    :NEW.WW_CLAY := ROUND(NVL(:NEW.PHI_09,0) + NVL(:NEW.PHI_10,0) +
NVL(:NEW.PHI_11,0) + NVL(:NEW.PHI_12,0), 3);
    :NEW.WW_COLLOID := ROUND(NVL(:NEW.PHI_13,0) + NVL(:NEW.PHI_14,0), 3);
```

DataDictionary

```
EXCEPTION  
  WHEN exc  
  THEN  
    raise_application_error(-20000, strMsg);  
end;  
/
```


T_SAMPLES_X_CORELAYERS

Notes: Association table to allow for many-to-many relationships between samples and core layers. Possible relationships include one or more samples taken from a single core layer, a sample taken from across more than one layer within a single core, and virtual samples, where one sample is used to describe more than one layer within a single core.

Primary key:

1. SAMP_SAMPLE_ID
2. CL_CORE_LAYER_ID

Foreign keys	Child	Parent
CL_SXC_FK	CL_CORE_LAYER_ID	T_CORELAYERS.CORE_LAYER_ID
SAMP_SXC_FK	SAMP_SAMPLE_ID	T_SAMPLES.SAMPLE_ID

Column details

1. SAMP_SAMPLE_ID (FK)

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Foreign key to table of samples. First part of joint primary key in this association table.

2. CL_CORE_LAYER_ID (FK)

Physical data type: NUMBERPS(10,0)
Portable data type: N-Decimal(10,0)
Allow NULLs: Not allowed
Notes: Foreign key to table of core layers. Second part of joint primary key in this association table.

3. VIRTUAL_SAMPLE

Physical data type: VARCHAR2(1)
Portable data type: C-Variable Length(1)
Allow NULLs: Not allowed
Notes: Flags whether the related sample record is a virtual sample within the related core layer record. Values "Y" or "N"?

Code details

1. SXC_TRG

Type: Trigger
Code body:

```
CREATE OR REPLACE TRIGGER SXC_TRG
BEFORE INSERT OR UPDATE ON T_SAMPLES_X_CORELAYERS
FOR EACH ROW
declare
    CL_CORE_ID number;
    CL_PROJECT_ID number;
    SAMP_CORE_ID number;
    SAMP_PROJECT_ID number;
begin
    select core_core_id, prj_project_id
    into cl_core_id, cl_project_id
    from t_core_layers, t_cores
    where core_layer_id = :new.cl_core_layer_id
    and core_core_id = core_id
    and rownum = 1;

    select core_core_id, prj_project_id
    into samp_core_id, samp_project_id
    from t_core_layers cl, t_cores cor
    where cl.core_layer_id = :new.cl_core_layer_id
    and cl.core_core_id = cor.core_id
    and rownum = 1;

    if samp_core_id <> cl_core_id then
        raise_application_error(-20001, 'Mismatched Core IDs in associated records in T_SAMPLES and
T_CORELAYERS');
    end if;
```

DataDictionary

```
if samp_project_id <> cl_project_id then
  raise_application_error(-20001, 'Mismatched Project IDs in associated records in T_SAMPLES and
T_CORES');
end if;
end;
/
```

T_SITE_INFO

Notes: Used internally by ROSS website.

Primary key:

Column details**1. PK_SITEINFO**

Physical data type: NUMBERPS(3,0)

Portable data type: N-Decimal(3,0)

Allow NULLs: Not allowed

2. SITE_QUESTION

Physical data type: VARCHAR2(500)

Portable data type: C-Variable Length(500)

Allow NULLs: Allowed

3. SITE_INFO

Physical data type: VARCHAR2(4000)

Portable data type: C-Variable Length(4000)

Allow NULLs: Allowed

4. USERMAN

Physical data type: VARCHAR2(1)

Portable data type: C-Variable Length(1)

Allow NULLs: Allowed

5. USERMAN_LOCATION

Physical data type: VARCHAR2(100)

Portable data type: C-Variable Length(100)

Allow NULLs: Allowed

T_SITE_METADATA**Notes:**

Metadata table used internally by ROSS website. Controls columns that appear in database queries.

Primary key:**Column details****1. COLUMN_NAME**

Physical data type: VARCHAR2(30)
 Portable data type: C-Variable Length(30)
 Allow NULLs: Not allowed

2. ALIAS

Physical data type: VARCHAR2(45)
 Portable data type: C-Variable Length(45)
 Allow NULLs: Allowed

3. DESCRIPTION

Physical data type: VARCHAR2(2000)
 Portable data type: C-Variable Length(2000)
 Allow NULLs: Allowed

4. DISPLAY_ORDER

Physical data type: NUMBERPS(3,0)
 Portable data type: N-Decimal(3,0)
 Allow NULLs: Not allowed
 Default value: 999

5. DISPLAY_YN

Physical data type: CHAR(1)
 Portable data type: C-Fixed Length(1)
 Allow NULLs: Not allowed
 Default value: Y

6. SAMPLE_DATA_YN

Physical data type: CHAR(1)
 Portable data type: C-Fixed Length(1)
 Allow NULLs: Not allowed
 Default value: N

7. CORE_DATA_YN

Physical data type: CHAR(1)
 Portable data type: C-Fixed Length(1)
 Allow NULLs: Not allowed
 Default value: N

8. DISPLAY_GROUP

Physical data type: VARCHAR2(30)
 Portable data type: C-Variable Length(30)
 Allow NULLs: Allowed

T_SOIL_DESCRIPTORS

Notes: Lookup table of soil descriptors.
Primary key: SOIL_DESCRIPTOR_ID

Foreign keys	Child	Parent
SD_CLQ_FK	T_CORE_LAYER_QUALIFIERS.SD_SOIL_D ESCRIPTOR_ID	SOIL_DESCRIPTOR_ID

Column details**1. SOIL_DESCRIPTOR_ID**

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a soil descriptor.

2. SOIL_DESCRIPTOR

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Soil descriptor.

T_SOIL_TEXTURES

Notes: Lookup table of soil textures.
Primary key: SOIL_TEXTURE_ID

Foreign keys	Child	Parent
STX_CLQ_FK	T_CORE_LAYER_QUALIFIERS.STX_SOIL_TEXTURE_ID	SOIL_TEXTURE_ID

Column details**1. SOIL_TEXTURE_ID**

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a soil texture.

2. SOIL_TEXTURE

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Soil texture.

T_SOIL_TYPES

Notes: Lookup table of soil types.

Primary key: SOIL_TYPE_ID

Foreign keys	Child	Parent
ST_CLQ_FK	T_CORE_LAYER_QUALIFIERS.ST_SOIL_T YPE_ID	SOIL_TYPE_ID

Column details**1. SOIL_TYPE_ID**

Physical data type: NUMBERPS(3,0)

Portable data type: N-Decimal(3,0)

Allow NULLs: Not allowed

Notes: Uniquely identifies a soil type.

2. SOIL_TYPE

Physical data type: VARCHAR2(50)

Portable data type: C-Variable Length(50)

Allow NULLs: Not allowed

Notes: Soil type.

T_SORTING

Notes: Sand sorting descriptors.
Primary key: SORTING_ID

Foreign keys	Child	Parent
S_CLQ_FK	T_CORE_LAYER_QUALIFIERS.S_SORTING_ID	SORTING_ID

Column details**1. SORTING_ID**

Physical data type: NUMBERPS(3,0)
Portable data type: N-Decimal(3,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a sorting record.

2. SORTING

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Describes a soil sorting (e.g.: "Extremely Poorly Sorted").

3. STANDARD DEVIATION

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Not allowed
Notes: Standard deviation associated with this sorting type.

T_SPHERICITY_LU

Notes: Lookup table of sphericity types.
Primary key: SPHERICITY_ID

Foreign keys	Child	Parent
SLU_SAMP_FK	T_SAMPLES.SLU_SPHERICITY_ID	SPHERICITY_ID

Column details**1. SPHERICITY_ID**

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a sphericity descriptor.

2. SPHERICITY

Physical data type: VARCHAR2(15)
Portable data type: C-Variable Length(15)
Allow NULLs: Not allowed
Notes: Sphericity descriptor.

Code details**1. SLU_TRG**

Type: Trigger
Code body:

```

create sequence slu_seq INCREMENT BY 1 START WITH 1
  MAXVALUE 1.0E28 MINVALUE 1 NOCYCLE
  NOCACHE NOORDER;

create or replace trigger "SLU_TRG" BEFORE INSERT ON "T_SPHERICITY_LU" FOR EACH
ROW

BEGIN
  if :new.sphericity_id is null then
    SELECT SLU_SEQ.NEXTVAL
    INTO :NEW.SPHERICITY_ID
    FROM DUAL;
  end if;
END;
/

```

T_USCS_CLASSIFICATIONS

Notes: Unified Soils Classification System classifications.
Primary key: USCS_CLASSIFICATION_ID

Foreign keys	Child	Parent
USCS_CL_FK	T_CORE_LAYERS.USCS_USCS_CLASSIFICATION_ID	USCS_CLASSIFICATION_ID
USCS_SAMP_FK	T_SAMPLES.USCS_USCS_CLASSIFICATION_ID	USCS_CLASSIFICATION_ID

Column details**1. USCS_CLASSIFICATION_ID**

Physical data type: NUMBERPS(5,0)
Portable data type: N-Decimal(5,0)
Allow NULLs: Not allowed
Notes: Uniquely identifies a USCS classification record.

2. CLASSIFICATION_NAME

Physical data type: VARCHAR2(10)
Portable data type: C-Variable Length(10)
Allow NULLs: Not allowed
Notes: The name of the classification.

3. CLASSIFICATION_DESCRIPTION

Physical data type: VARCHAR2(100)
Portable data type: C-Variable Length(100)
Allow NULLs: Not allowed
Notes: The description of the classification.

4. PHI_RANGE

Physical data type: VARCHAR2(50)
Portable data type: C-Variable Length(50)
Allow NULLs: Allowed
Notes: The range of grain sizes (in phi) for this soil classification.