Query Sample Purpose

The purpose of this sample is to illustrate the most critical tables, attributes and relationships in the Soil Data Mart database. The vast majority of queries executed against the Soil Data Mart database will be some variant of this "fundamental" query.

Sample Query Syntax

The following query returns selected attributes from tables "sacatalog", "legend", "mapunit", "component" and "chorizon", for the SSURGO map units in a four square mile area of interest in Hall County Nebraska. For additional information about constraining a query to an area of interest, please see the sample query document titled "Constraining a Query to an Area of Interest".

```
--Sample query begins…
--Note that a pair of dashes denotes the beginning of a comment.
SELECT
saversion, saverest, -- attributes from table "sacatalog"
l.areasymbol, l.areaname, l.lkey, -- attributes from table "legend"
musym, muname, museq, mu.mukey, -- attributes from table "mapunit"
comppct_r, compname, localphase, slope_r, c.cokey, -- attributes from
table "component"
hzdept_r, hzdepb_r, ch.chkey -- attributes from table "chorizon"

FROM sacatalog sac
  INNER JOIN legend l ON l.areasymbol = sac.areasymbol
    INNER JOIN mapunit mu ON mu.lkey = l.lkey
    AND mu.mukey IN

('107559','107646','107674','107682','107707','107794','107853','107854','
107865','107867','107869','107870','107871')
      LEFT OUTER JOIN component c ON c.mukey = mu.mukey
        LEFT OUTER JOIN chorizon ch ON ch.cokey = c.cokey

--WHERE…

--ORDER BY l.areaname, museq, comppct_r DESC, compname, hzdept_r --
standard soil report ordering
--Sample query ends.
```

Fundamental Query Description

Tables "legend", "mapunit", "component" and "chorizon" make up what we refer to as the "backbone" of our tabular data model, and these tables represent the root table of each soil entities represented in the Soil Data Mart database, i.e. survey area (legend), map unit, map unit component, and soil horizon or layer.

Keep in mind that the tables in the Soil Data Mart database that record the non-spatial soil data include records for both SSURGO and STATSGO. The results of this query are constrained to SSURGO because each of the map units referenced in the Where clause happens to be a SSURGO map unit. For additional information about discriminating between SSURGO and STATSGO, please see the sample query document titled "Discriminating between SSURGO and STATSGO".

Let's examine this query in detail.

Select Clause

This sample query includes only a few attributes from each table in this query. Obviously it would be a trivial matter to include any number of additional attributes from any table in this query. The more attributes you select, the larger your result set is going to be.

Why did I select these attributes? These selections include those attributes that you might want to select in any query for the corresponding table. With the exception of one table, these are the attributes that tend to distinguish one instance of an entity (legend or survey area, map unit, component, soil horizon or layer) from another.

Let's look at the attributes on a table by table basis.

Table "sacatalog"

sacatalog.saversion

"saversion" is the version number of the corresponding survey area. A survey area corresponds to a record in table "sacatalog" or a record in table "legend".

sacatalog.saverest

"saverest" is the date and time when that survey area version was established.

Neither of these attributes is adequate to identify a survey area, but they are important nonetheless. Soil data is versioned by survey area. Should you ever need to identify the vintage of a survey area (legend), map unit or component, you need to record one of these two attributes. Either attribute will suffice to unambiguously

identify the vintage of the corresponding survey area, map unit or component. Data for a survey area can be updated numerous times over the course of a year.

Table "legend"

legend.areasymbol

"areasymbol", for SSURGO data, is a five character string that unambiguously identifies a SSURGO survey area. The first two characters are the alphabetic postal code of the state or territory with administrative responsibility for the corresponding survey area. The last three characters are a zero filled three digit integer number that distinguishes survey areas administered by the same state or territory. When a survey area shares an exact coincidence with a single county, that three digit number corresponds to that county's numeric FIPS code. For survey areas that don't have a one to one coincidence with a county, the three digit part of the area symbol is typically 510 or higher.

legend.areaname

"areaname" for either SSURGO or STATSGO data, identifies the geographic area that participates in the corresponding survey area. For survey areas that contain all or part of multiple counties, the survey area name typically includes the name of each county that coincides with that survey area, and also indicates if all or part of that county is coincident with that survey area. Although we don't have a unique constraint on the combination of dataset (SSURGO or STATSGO) and "areaname", at the time this was written, all survey area names for a given dataset were unique. Given our nomenclature standards, you could reasonably expect a survey area name for a given dataset to be unique.

Keep in mind that the entire STATSGO dataset represents a single survey area whose area symbol is "US" and whose area name is "United States".

legend.lkey

"lkey" is a surrogate key that unambiguously identifies any SSURGO or STATSGO survey area or legend, which also implies that it can be used to unambiguously identify a record in table "legend". Although this is a character field, its value always represents a valid integer number.

Table "mapunit"

mapunit.musym

"musym" is a map unit's symbol. A map unit symbol contains six or fewer characters, which are typically limited to digits and letters. The letters may include both upper case and lower case. Within a survey area, a map unit symbol unambiguously identifies a map unit. Since many analyses will likely include map units from

more than one survey area, this field alone has limited value. But for either SSURGO or STATSGO, the combination of survey area symbol and map unit symbol is unique.

mapunit.muname

"muname" is a map unit's name. A map unit name typically includes the name of the major soil components in that map unit, and often includes a slope range. Within a survey area, map unit names tend to be unique, but no such constraint is enforced.

mapunit.museq

"museq" is an integer value that logically orders the map units within a given survey area. This is handy to have when creating a report that displays map units by survey area. For additional details, see the section titled "Order By Clause".

mapunit.mukey

"mukey" is a surrogate key that unambiguously identifies any SSURGO or STATSGO map unit, which also implies that it can be used to unambiguously identify a record in table "mapunit". Although this is a character field, its value always represents a valid integer number. On a map for a single survey area, "musym" can be used to uniquely identify each map unit. On a map that includes map units from more than one survey area, "mukey" is the only single field that can be used to identify a map unit.

Table "component"

For a component of a map unit, there is no set of soil business related attributes that is guaranteed to unambiguously distinguish one map unit component from another. The combination of the attributes below, with the exception of "cokey", *tends* to distinguish components of the same map unit, and probably does so for the vast majority of map units.

component.comppct_r

"comppct_r" is an integer value that denotes the average or representative percent composition of the corresponding component in the corresponding map unit.

A number of soil attributes are recorded as three related values that we refer to as "low, representative value or RV, and high". The low and high values denote the typical range of values of that attribute in the corresponding map unit component or soil horizon or layer. The representative value denotes the average, or "expected value" of that attribute in the corresponding map unit component or soil horizon or layer. In soil reports, the components of a map unit are typically sorted in descending order on "RV percent composition" so that the components that occupy the largest percent of the map unit are listed first.

The sum of "comppct_r" for the components of a given map unit may be less than 100. For some map units, some minor components may not be explicitly recorded. It is also possible for the sum of "comppct_r" for the components of a given map unit to exceed 100, because some components may not occur in all map units.

component.compname

"compname" records a component's name. A map unit component may either be a soil or non-soil entity. Non-soil entities include things like "Rock outcrop", "Riverwash" and "Gravel pits", and in such a case these terms will serve as the component name. In most cases, a map unit component corresponds to a "soil series" name. A soil series is a soil that can be well defined and is mapped extensively enough to have been given its own name.

component.localphase

"localphase" corresponds to a "phase name". A phase name can be associated with a soil series name to indicate how a particular instance of that soil series differs from that soil series in general. Phase names includes terms such as "stony", "eroded", "cool", etc.

component.slope_r

"slope_r" is an integer value that denotes the average or representative percent slope of the corresponding component in the corresponding map unit. Slope is another of those attributes for which a low, high and representative value is recorded.

component.cokey

"cokey" is a surrogate key that unambiguously identifies a SSURGO or STATSGO map unit component, which also implies that it can be used to unambiguously identify a record in table "component". This field is a character string that contains two integer values separated by a colon. The reason for this format isn't germane to this discussion.

Table "chorizon"

For a soil horizon or layer of a map unit component, there is no set of soil business related attributes that is guaranteed to unambiguously distinguish one soil horizon or layer from another. For most map unit components, the representative horizon depth to top and depth to bottom values describe a contiguous, non-overlapping set of soil horizons or layers, but this constraint is not enforced. With one exception, any overlaps or gaps based on representative horizon depth to top and depth to bottom would typically correspond to a data error. The exception is for something referred to as a transitional soil horizon. A transitional soil horizon has two distinct sets of characteristics, but those characteristics cannot be differentiated vertically throughout the

soil horizon. For such a soil horizon, two separate soil horizons are recorded, each with their own set of characteristics, but with the same representative depth to top and depth to bottom.

chorizon.hzdept_r

"hzdept_r" is an integer value that denotes the average or representative depth to the top of the corresponding soil horizon or layer, in centimeters. Horizon depth to top is another of those attributes for which a low, high and representative value is recorded. In soil reports, soil horizons or layer are sorted in ascending order by this value. The representative depth to top of the surface soil horizon or layer is zero.

chorizon.hzdepb_r

"hzdepb_r" is an integer value that denotes the average or representative depth to the bottom of the corresponding soil horizon or layer, in centimeters. Horizon depth to bottom is another of those attributes for which a low, high and representative value is recorded.

chorizon.chkey

"chkey" is a surrogate key that unambiguously identifies a SSURGO or STATSGO soil horizon or layer, which also implies that it can be used to unambiguously identify a record in table "chorizon". This field is a character string that contains two integer values separated by a colon. The reason for this format isn't germane to this discussion.

From Clause

Keep in mind the following:

1. A record in table "sacatalog" should always have one and only one corresponding record in table "legend"

2. A record in table "legend" should always have more than one corresponding record in table "mapunit". At the time this was written, the number of map units in a SSURGO survey area varied from 7 to 727, with an average of 89, and the Soil Data Mart database contained 3037 SSURGO survey areas.

3. A record in table "mapunit" may or may not have a corresponding record in table "component". Map units with names like "Dam", "Denied Access", "Gravel Pit", "Quarry", "Not Complete", "Water", etc., often do not have a corresponding record in table "component".

4. A record in table "component" may or may not have a corresponding record in table "chorizon". As is the case for map units, a component can also correspond to a term like those listed under item 3. In such a case, that component typically does not have a corresponding record in table "chorizon". In

addition, a map unit may include both "major" and "minor" components. It is not uncommon for a "minor" component to have no corresponding record in table "chorizon".

In the query above, the joins are specified in the "From clause" rather than in the "Where clause". This capability was defined as part of the SQL-92 standard. I still struggle with using this syntax myself, but I highly recommend becoming familiar with this syntax because it allows a level of control that can't be replicated using only a Where clause. I found it difficult to find what I considered a thorough reference that fully documents the capabilities of this syntax. The only book I ever found that I considered adequate is Advanced ANSI SQL Data Modeling and Structure Processing by Michael M. David. I was able to order this book through amazon.com, but it appears that amazon.com doesn't normally keep this book in stock.

Anyway, the From clause above takes items 1 to 4 into account, assuring you that all map units and components will be returned. In the Soil Data Mart database, all of the tables to which you have access are joined on a single column that has the same name in both related tables. In the vast majority of cases, the name of that column ends with "key", and the prefix reflects the table for which that column is the primary key. For example legend.lkey is the primary key of table "legend", mapunit.mukey is the primary key of table "mapunit", etc.

The From clause also includes the constraint "AND mu.mukey IN ('107559', '107674', '107682', '107707', '107853', '107865', '107867')". This is what constrains the results of this query to a set of SSURGO map units in a four square mile area of interest in Hall County Nebraska. I used Web Soil Survey to create this area of interest and I chose to display national map unit symbols (mapunit.mukey) rather than survey area specific map unit symbols (mapunit.musym). For additional information on discriminating between SSURGO and STATSGO, please see the sample query document titled "Discriminating between SSURGO and STATSGO".

Where Clause

I didn't illustrate anything specific in the commented out Where clause. I included it to illustrate that additional constraints could be added by using a Where clause.

Order By Clause

Notice that the Order By clause is commented out by prefacing it with two dashes. Including an Order By clause for a large result set can have a serious impact on overall query performance. I included the commented out Order By clause in this example only to show the sorting that is employed by most Soil Data Mart reports.

Column mapunit.museq can be used to logically sequence the map units in a particular survey area, but cannot be used to logically sequence the map units in multiple survey areas. This is suitable for soil reports in the Soil Data Mart because most reports include a control break on survey area, therefore map units from more than one survey area are never included in a given report section. A survey area is equivalent to a record in table "legend".

Although a map unit symbol is always a character string, that string can include letters and digits, letters only or digits only. Some survey areas may include map unit symbols that represent two or more of these categories.

As most of you are aware, sorting a set of character strings that all represent integer values doesn't produce the desired result, because "1", "10" and "100" will all precede "2", "20" and "200".

So what do I mean when I say "logically sequenced"? To sort the map unit symbols for a survey area, we break each map unit symbol into any preceding numeric part, if any, the purely alphabetic part, if any, and a trailing numeric part, if any. For example, a symbol like "27Ac2" would be broken into "27", "Ac" and "2". After breaking each map unit symbol into components, we then sort by numeric part 1, if any, using a numeric sort, the alphabetic part, if any, using a lexigraphical sort, and numeric part 2, if any, using a numeric sort, and then we assign the corresponding sort sequence to mapunit.museq. We preserve this number in the Soil Data Mart database so that we don't have to constantly resort map units each time a report is generated.

Obviously this scheme breaks down if a map unit symbol contains more than 2 distinct numeric parts or more than one distinct alphabetic part, but fortunately, to the best of my knowledge, none of our existing map unit symbols violate these constraints.

Note that within a soil report, the components for a given map unit are typically sorted in descending order by their representative percent composition. It is possible for a map unit to contain two components with different names, with the same representative percent composition, so we then sort on component name, ascending. While it is also possible for a map unit to contain two components with the same name and the same representative percent composition, that possibility is rare enough that we chose to not deal with it.