

More SQL – Pivot, Sub Queries Spatial, etc.

GEOG/ECOL/ENR 5050

Fall 2016

Instructor: Dr. Shannon E. Albeke

A Helpful Reference

- <http://beginner-sql-tutorial.com/sql.htm>

UNPIVOT

- Great method for taking horizontal data and transforming it to our preferred method of storing vertical data
- First test to make sure we are pivoting the columns we want....note that the value column must all be of the same data type
- The Source we want to pivot

```
SELECT Plot, PlotArray, SampDate, SpecCode, SampleID, Parameter,  
Meas  
FROM (SELECT * FROM tblFieldData) AS flat  
UNPIVOT (Meas FOR Parameter IN(SVL, HF_mm, Tail_mm, Body_mm, Wt_g))  
AS unpvt
```

- Now push the result into a new table

```
SELECT Plot, PlotArray, SampDate, SpecCode, SampleID, Parameter,  
Meas  
INTO tblFieldMeas  
FROM (SELECT * FROM tblFieldData) AS flat  
UNPIVOT (Meas FOR Parameter IN(SVL, HF_mm, Tail_mm, Body_mm, Wt_g))  
AS unpvt
```

PIVOT

- Great method for taking vertical data and displaying in a flat format (just like the Pivot Table in Excel)
- First must create the data that you want to pivot (i.e. taking the average value for a species and measurement)

- The Source we want to pivot

```
SELECT SpecCode, Parameter, AVG(Meas) AS Meas  
FROM tblFieldMeas  
GROUP BY SpecCode, Parameter
```

- Then Pivot the above result

```
SELECT SpecCode, SVL, HF_mm, Tail_mm, Body_mm, Wt_g  
FROM (SELECT SpecCode, Parameter, AVG(Meas) AS Meas FROM  
tblFieldMeas GROUP BY SpecCode, Parameter) AS vert  
PIVOT (AVG(Meas) FOR Parameter IN (SVL, HF_mm, Tail_mm,  
Body_mm, Wt_g)) AS pvt
```

PIVOT

- Sometimes you want to make your Columns Dynamic. This is somewhat complex, but here is a link to an example of how to do it.
- <https://www.mssqltips.com/sqlservertip/2783/script-to-create-dynamic-pivot-queries-in-sql-server/>
- DECLARE @columns NVARCHAR(MAX), @sql NVARCHAR(MAX);
SET @columns = N'';
SELECT @columns += N', p.' + QUOTENAME(Parameter)
FROM (SELECT p.Parameter FROM tblFieldMeas AS p
GROUP BY p.Parameter) AS x;

```
SET @sql = N'  
SELECT SpecCode, ' + STUFF(@columns, 1, 2, '') + '  
FROM  
(SELECT SpecCode, Parameter, AVG(Meas) AS Meas  
FROM tblFieldMeas GROUP BY SpecCode, Parameter) AS j  
PIVOT  
(Avg(Meas) FOR Parameter IN (  
+ STUFF(REPLACE(@columns, ', p.[', ',[', 1, 1, '') + ')) AS p;'  
PRINT @sql;  
EXEC sp_executesql @sql;
```

Sub Query

- <https://www.simple-talk.com/sql/sql-training/subqueries-in-sql-server/>
- Very efficient method for querying the result(s) of a query
- For example, the previous PIVOT result, has all species. Let's limit results to only Rodents
- First, create a list of SpecCode that are rodents

```
SELECT SpecCode  
FROM luSpecies  
WHERE Taxon='ROD'
```

- This result is only a single column, we can use this to filter another query result

Sub Query, cont'd

- Now that you know which species you want in the result, make the PIVOT result into it's own source 'table' by placing the SQL statement within parentheses and giving it an alias and selecting all columns

```
SELECT *  
FROM (SELECT SpecCode, SVL, HF_mm, Tail_mm, Body_mm, Wt_g  
FROM (SELECT SpecCode, Parameter, AVG(Meas) AS Meas FROM  
tblFieldMeas GROUP BY SpecCode, Parameter) AS vert  
PIVOT (AVG(Meas) FOR Parameter IN (SVL, HF_mm, Tail_mm, Body_mm,  
Wt_g)) AS pvt) AS Src
```

Sub Query, cont'd

- Now we have a 'new' result that we can then filter with the result of our rodent query

```
SELECT *
FROM (SELECT SpecCode, SVL, HF_mm, Tail_mm, Body_mm, Wt_g
FROM (SELECT SpecCode, Parameter, AVG(Meas) AS Meas FROM
tblFieldMeas GROUP BY SpecCode, Parameter) AS vert
PIVOT (AVG(Meas) FOR Parameter IN (SVL, HF_mm, Tail_mm, Body_mm,
Wt_g)) AS pvt) AS Src
WHERE
SpecCode IN(SELECT SpecCode FROM luSpecies WHERE Taxon='ROD')
```


DATEPART Function

- We know to always store date/time in one column. However sometimes we still want some portion of the date/time as a grouping column. Use the DATEPART Function to assist (e.g. ... <http://msdn.microsoft.com/en-us/library/ms174420.aspx>)

```
SELECT SampDate, DATEPART(yy, SampDate) AS SampYear
FROM tblFieldMeas
GROUP BY SampDate
```

DATEDIFF Function

- You can calculate the length of time between two values, very useful. Use the DATEDIFF Function to assist
(e.g. ... <http://msdn.microsoft.com/en-us/library/ms189794.aspx>)

```
SELECT DATEDIFF(yy, CAST('1/1/1907' AS date), CAST('1/1/2014' AS date))
```

Intro to Spatial Objects

- Two different data types
 - geography = spheroid based so lon/lat
 - geometry = projected (e.g. UTM)
- Require an EPSG code:
 - <http://spatialreference.org/>
 - SREL data (our data) in NAD83, Zone 17
 - **EPSG:26917**

Extract Point Information

- SQL Can act as a GIS....HOORAY for us!
- If you have a spatial column, you can obtain the stored coordinates from the 'geometry' column through the use of 'ST' commands
- Get the X and Y from a point feature class

```
SELECT Shape, Point_ID, Shape.STX AS UTMX, Shape.STY AS UTMY  
FROM fcTrapArrays
```

Spatial Methods

- Here is a list of the possible Spatial Methods/Queries

<http://msdn.microsoft.com/en-us/library/bb933960.aspx>

- Please see the SpatialQueriesIntro.sql file for the in class examples