USDA NRCS

Statewide Aggregated Soil Attributes For Wisconsin

Version 1.0

Jesse Turk

9/11/2011

The following document is designed as a tool to create mapunit level aggregated tables of SSURGO certified soils data for use in GIS systems. It should be considered as a work in progress and updates should be sought out.

This document contains SQL queries designed to be used in the Soil Data Mart Web Application Interface. With the exception of the query to create the Appendix 1 of the NRCS Conservation Planning Tech note 1, the results of the queries are designed to produce complete aggregated tables for SSURGO soil properties, interpretations and general information. By complete, we mean that one row of data will be provided for each soil map unit in the state, regardless if that map unit is populated with the attribute, properties or interpretation (8116 rows as of 2/22/2010).

Care has been taken to document what the actual soil property, interpretation or attribute actually is and the aggregation method that has been used to derive it. However, it is strongly recommended that you contact a soil scientist from the Natural Resources Conservation Service before using any attribute in this table in order to make sure the attribute is being used correctly. For contact information, please visit http://www.wi.nrcs.usda.gov/technical/soil/contact.html or contact:

Jesse Turk
715-343-6200
1462 Strongs Ave
Stevens Point, WI 54481
mailto:jesse.turk@wi.usda.gov

Usage:

Since the intent of all the queries, with the exception of the of Appendix 1 of the NRCS Conservation Planning Tech note 1 query, are intended to be used in a GIS, the best method to retrieve the results is to download the table as a text file. Steps to retrieve a text file of the table are:

1. Identify the attribute that you are interested and find its associated table.
2. Read the documentation and be sure you understand the attribute and aggregation method.
3. If the attribute and aggregation method is acceptable, copy the entire sql query, from one ------ break to the other ----- break.
4. Paste the copied query into the Soil Data Mart Web Application Interface located at this web address http://sdmdataaccess.nrcs.usda.gov/Query.aspx.
5. Select Queued/Text
6. Check “First row contains column names”
7. Choose Field Delimiter: Comma
8. Choose Text Delimiter: Double Quote
9. Enter and confirm your e-mail address.

Once the query has processed, you will receive an email with a link to a zip file containing the table. Unfortunately, the table is named table.txt by default. You will need to open the zip file and extract the table.txt file and rename. If you attempt to import the table into ArcGIS named as table.txt, you will get an error. So the table must be renamed.

Definitions:

Aggregation: Aggregation is the process by which a set of component attribute values is reduced to a single value to represent the map unit as a whole.
A map unit is typically composed of one or more "components". A component is either some type of soil or some nonsoil entity, e.g., rock outcrop. The components in the map unit name represent the major soils within a map delineation. Minor components make up the balance of the map unit. Great differences in soil properties can occur between map unit components and within short distances. Minor components may be very different from the major components. Such differences could significantly affect use and management of the map unit. Minor components may or may not be documented in the database. The results of aggregations do not reflect the presences or absence of limitations of the components which are not listed in the database. An on-site investigation is required to identify the location of individual map unit components.
For each of a map unit's components, a corresponding percent composition is recorded. A percent composition of 60 indicates that the corresponding component typically makes up approximately 60% of the map unit. Percent composition is a critical factor in some, but not all, aggregation methods.
For the attribute being aggregated, the first step of the aggregation process is to derive one attribute value for each of a map unit's components. From this set of component attributes, the next step of the aggregation process derives a single value that represents the map unit as a whole. Once a single value for each map unit is derived, a thematic map for soil map units can be generated. Aggregation must be done because, on any soil map, map units are delineated but components are not.

Dominant Condition: The aggregation method "Dominant Condition" first groups like attributed values for the components in a map unit. For each group, percent composition is set to the sum of the percent composition of all components participating in that group. These groups not represent "conditions" rather than components. The attribute value associated with the group with the highest cumulative percent composition is returned. If more than one group shares the highest cumulative percent composition, the "most limiting" result is typically returned.

Dominant Component: The aggregation method "Dominant Component" returns the attribute value associated with the component with the highest percent composition in the map unit. If more than one component shares the highest percent composition, the "most limiting" result is typically returned.

Most Limiting: This aggregation method returns the most restrictive result. For this aggregation method, the most limited result among all components of the map unit is returned. If one were making a decision based on this result, that decision would be based on the most conservative, or most pessimistic, result.

Least Limiting: This aggregation method returns the least restrictive result. For this aggregation method, the least limiting result among all components of the map unit is returned. If one were making a decision based on this result, that decision would be based on the least conservative, or most optimistic, result.

Weighted Average: The aggregation method "Weighted Average" computes a weighted average value for all components in the map unit. Percent composition is the weighting factor.
The result returned by this aggregation method represents a weighted average value of the corresponding attribute throughout the map unit.

Queries

****

Mapunit Table

Field Name	Definition
musym	*The symbol used to uniquely identify the soil mapunit in the soil survey.*
muname	*Correlated name of the mapunit*
mukind	*Code identifying the kind of mapunit.*
farmlndcl	*Identification of map units as prime farmland, farmland of statewide importance*
mukey	*A non-connotative string of characters used to uniquely identify a record in the Mapunit table.*
mukey_num	*A non-connotative integer used to uniquely identify a record in the Mapunit table*

```
SELECT  mu.musym, mu.muname, mu.mukind, mu.farmlndcl, mu.mukey, cast (mu.mukey as int) as mukey_num
FROM sacatalog sac
      INNER JOIN legend l ON l.areasymbol = sac.areasymbol and l.areasymbol LIKE 'WI%'
      INNER JOIN mapunit mu ON mu.lkey = l.lkey
```

****

Properties Table

Field Name	Definition
mukey_num	*A non-connotative integer used to uniquely identify a record in the Mapunit table*
mukey	*A non-connotative string of characters used to uniquely identify a record in the Mapunit table.*
drclassdcd	*The natural drainage condition of the soil refers to the frequency and duration of wet periods. This column displays the dominant drainage class for the map unit, based on composition percentage of each map unit component.*
drclasswettest	*The natural drainage condition of the soil refers to the frequency and duration of wet periods. This column displays the wettest drainage class assigned to an individual component of the map unit whose composition in the map unit is equal to or exceeds 15%.*
brockdepmin	*The distance from the soil surface to the top of a bedrock layer, expressed as a shallowest depth of components whose composition in the map unit is equal to or exceeds 15%. Map units that do not have bedrock contact within 203 cm are reported as null.*
wtdepannmin	*The shallowest depth to a wet soil layer (perched or apparent water table) at any time during the year expressed as centimeters from the soil surface, for components whose composition in the map unit is equal to or exceeds 15%. Map units that do not have any wet soil layers within 203cm are reported as null*
wtdepaprjunmin	*The shallowest depth to a wet soil layer (perched or apparent water table) during the months of April through June expressed as centimeters from the soil surface, for components whose composition in the map unit is equal to or exceeds 15%. Map units that do not have any wet soil layers within 203cm are reported as null*
flodfreqdcd	*The annual probability of a flood event expressed as a class. This column displays the dominant flood frequency class for the map unit, based on composition percentage of map unit components whose composition in the map unit is equal to or exceeds 15%.*
flodfreqmax	*The annual probability of a flood event expressed as a class. This column displays the highest probability class assigned to an individual component of the map unit whose composition in the map unit is equal to or exceeds 15%.*
pondfreqprs	*The percentage of the map unit that is subject to water being ponded on the soil surface, expressed as one of four classes; 0-14%, 15-49%, 50-74% or 75-100%.*
hydgrpdcd	*Hydrologic Group is a grouping of soils that have similar runoff potential under similar storm and cover conditions. This column displays the dominant hydrologic group for the map unit, based on composition percentage of each map unit component.*
NoDuffSufTex	*Surface Texture of the dominant condition, with the duff layer texture removed, where there is a duff layer.*
SufTex	*Surface Texture of the dominant condition*

```
SELECT
cast (mu.mukey as int) as mukey_num, mu.mukey, muag.drclassdcd, muag.drclasswettest, muag.brockdepmin,
muag.wtdepannmin, muag.wtdepaprjunmin, muag.flodfreqdcd, muag.flodfreqmax, muag.pondfreqprs, muag.hydgrpdcd
```

INTO #muagTemp
FROM sacatalog sac
INNER JOIN legend l ON l.areasymbol = sac.areasymbol and l.areasymbol LIKE 'WI%'
INNER JOIN mapunit mu ON mu.lkey = l.lkey
INNER JOiN muaggatt muag on muag.mukey = mu.mukey

--Texture routine. This is incomplete as of now. Mapunits with 2 domcond are represented with two rows. The problem lies with T-sql having no FIRST function, need to look into Top.

--Fixed. 5/7/2010 Used row_number.

SELECT mapunit.mukey, Sum(component.comppct_r) AS SumOfcomppct_r, chorizon.hzdept_r, chtexturegrp.texture, chtexturegrp.rvindicator
INTO #TempTex1
FROM sacatalog sac
INNER JOIN legend l ON l.areasymbol = sac.areasymbol and l.areasymbol LIKE 'WI%'
INNER JOIN mapunit ON mapunit.lkey = l.lkey
INNER JOIN (component INNER JOIN (chorizon INNER JOIN chtexturegrp ON chorizon.chkey = chtexturegrp.chkey) ON component.cokey = chorizon.cokey) ON mapunit.mukey = component.mukey
GROUP BY mapunit.musym, mapunit.muname, mapunit.mukey, chorizon.hzdept_r, chtexturegrp.texture, chtexturegrp.rvindicator
HAVING (((chorizon.hzdept_r)=0) AND ((chtexturegrp.rvindicator)='yes'))

SELECT Max(#TempTex1.SumOfcomppct_r) AS MaxOfSumOfcomppct_r, #TempTex1.mukey
INTO #TempTex2
FROM #TempTex1 GROUP BY #TempTex1.mukey;

SELECT #TempTex1.texture, #TempTex1.mukey
INTO #TempTex3
FROM #TempTex1 INNER JOIN #TempTex2 ON (#TempTex1.mukey=#TempTex2.mukey) AND (#TempTex1.SumOfcomppct_r=#TempTex2.MaxOfSumOfcomppct_r);

SELECT mapunit.musym, #TempTex3.texture, mapunit.muname, mapunit.mukey
INTO #tex
FROM legend INNER JOIN (#TempTex3 RIGHT JOIN mapunit ON #TempTex3.mukey = mapunit.mukey) ON legend.lkey = mapunit.lkey
GROUP BY mapunit.musym, mapunit.muname, mapunit.mukey, legend.areasymbol, #TempTex3.texture
HAVING legend.areasymbol Like 'WI%';

WITH #Firstoftex1 AS (Select mukey, texture, rn = row_number() OVER (PARTITION BY mukey ORDER BY texture) From #tex)

Select texture, mukey
INTO #Firstoftex
From #Firstoftex1
Where rn=1

--Texture W\O Duff

--Forested Soils are often described with thin duff layers. In Wisconsin, approximately 570 major components are described with duff layers ranging from 2 to 10 cm. thick. Often, the duff layer is destroyed; therefore knowing the first mineral layer is beneficial. This field provides the texture of the first mineral layer below the duff layer.

SELECT mapunit.mukey, component.comppct_r, chorizon.hzdept_r, component.cokey, chorizon.chkey, chtexturegrp.texture, chtexturegrp.rvindicator, component.majcompflag
INTO #NoDuffTemp1
FROM (legend INNER JOIN (mapunit LEFT JOIN component ON mapunit.mukey = component.mukey) ON legend.lkey = mapunit.lkey) LEFT JOIN (chorizon LEFT JOIN chtexturegrp ON chorizon.chkey = chtexturegrp.chkey) ON component.cokey = chorizon.cokey
WHERE (((chorizon.hzdept_r)=(SELECT Min(chorizon.hzdept_r) AS MinOfhzdept_r
FROM chorizon LEFT JOIN chtexturegrp ON chorizon.chkey = chtexturegrp.chkey
Where chtexturegrp.texture Not In ('SPM','HPM', 'MPM') AND chtexturegrp.rvindicator='Yes' AND component.cokey = chorizon.cokey)) AND ((chtexturegrp.rvindicator)='Yes') AND ((legend.areasymbol) Like 'WI%') AND ((component.majcompflag)='Yes'))
ORDER BY legend.areasymbol, mapunit.musym, chorizon.hzdept_r

SELECT #NoDuffTemp1.mukey, Sum(#NoDuffTemp1.comppct_r) AS SumOfcomppct_r, #NoDuffTemp1.texture

```
INTO #NoDuffTemp2
FROM #NoDuffTemp1
GROUP BY #NoDuffTemp1.mukey, #NoDuffTemp1.texture

SELECT #NoDuffTemp2.mukey, Max(#NoDuffTemp2.SumOfcomppct_r) AS MaxOfSumOfcomppct_r
INTO #NoDuffTemp3
FROM #NoDuffTemp2
GROUP BY #NoDuffTemp2.mukey

SELECT #NoDuffTemp3.mukey, #NoDuffTemp2.texture
INTO  #NoDufftex
FROM #NoDuffTemp2 INNER JOIN #NoDuffTemp3 ON (#NoDuffTemp2.SumOfcomppct_r =
#NoDuffTemp3.MaxOfSumOfcomppct_r) AND (#NoDuffTemp2.mukey = #NoDuffTemp3.mukey);

WITH #FirstofNoDufftex1 AS (Select mukey, texture, rn = row_number() OVER (PARTITION BY mukey ORDER BY texture) From
#NoDufftex)

Select texture, mukey
INTO #FirstofNoDufftex
From #FirstofNoDufftex1
Where rn=1


SELECT #muagTemp.*, #FirstofNoDufftex.texture as NoDuffSufTex, #Firstoftex.texture as SufTex
FROM #muagTemp
LEFT JOIN #FirstofNoDufftex ON #muagTemp.mukey = #FirstofNoDufftex.mukey
LEFT JOIN #Firstoftex ON #muagTemp.mukey = #Firstoftex.mukey
```

****

Table Name: AWC

Available water storage (AWS) is the volume of water that the soil, to a specified depth, can store that is available to plants. AWS is calculated from AWC (available water capacity) which is commonly estimated as the difference between the water contents at 1/10 or 1/3 bar (field capacity) and 15 bars (permanent wilting point) tension, and adjusted for salinity and fragments.

For the purpose of this table, it is reported as the weighted average of all components in the map unit (really only the major components, since those are the ones that have AWC populated (as of 02/2010)), and is expressed as centimeters of water to a depth of 25, 50, 100 and 150 cm.

In order to interpret the AWS, groupings were created by Carl Wacker in the early 90's. The original groupings were based on inches of water and have been converted to cm of water and are outlined in the table below.

Centimeters of soil	Centimeters of Available Water				
	v_low	Low	Moderate	High	V_high
25	<1.27	>=.1.27	>= 2.54	>=3.81	>=5.08
50	<2.54	>=2.54	>=5.08	>=7.62	>=10.16
100	<5.08	>=5.08	>=10.16	>=15.24	>=20.32
150	<7.62	>=7.62	>=15.24	>=22.86	>=30.48

```
SELECT muaggatt.aws025wta,
CASE    WheN muaggatt.aws025wta is NULL THEN NULL
        WHEN muaggatt.aws025wta < 1.27 THEN 'v_low'
        WHEN muaggatt.aws025wta >= 1.27 AND muaggatt.aws025wta < 2.54 THEN 'low'
        WHEN muaggatt.aws025wta >= 2.54 AND muaggatt.aws025wta  < 3.81 THEN 'moderate'
        WHEN muaggatt.aws025wta >= 3.81 AND muaggatt.aws025wta  < 5.08 THEN 'high'
        WHEN muaggatt.aws025wta >= 5.08 THEN 'v_high'
        ELSE 'error'
        END AS aws025g,
muaggatt.aws050wta,
CASE    WHEN muaggatt.aws050wta is NULL THEN NULL
        WHEN muaggatt.aws050wta < 2.54 THEN 'v_low'
        WHEN muaggatt.aws050wta >= 2.54 AND muaggatt.aws050wta < 5.08 THEN 'low'
        WHEN muaggatt.aws050wta >= 5.08 AND muaggatt.aws050wta  < 7.62 THEN 'moderate'
        WHEN muaggatt.aws050wta >= 7.62 AND muaggatt.aws050wta  < 10.16 THEN 'high'
        WHEN muaggatt.aws050wta >= 10.16 THEN 'v_high'
        ELSE 'error'
        END AS aws050g,
```

```
muaggatt.aws0100wta,
CASE    WheN muaggatt.aws0100wta is NULL THEN NULL
        WHEN muaggatt.aws0100wta < 5.08 THEN 'v_low'
        WHEN muaggatt.aws0100wta >= 5.08 AND muaggatt.aws0100wta < 10.16 THEN 'low'
        WHEN muaggatt.aws0100wta >= 10.16 AND muaggatt.aws0100wta  < 15.24 THEN 'moderate'
        WHEN muaggatt.aws0100wta >= 15.24 AND muaggatt.aws0100wta  < 20.32 THEN 'high'
        WHEN muaggatt.aws0100wta >= 20.32 THEN 'v_high'
        ELSE 'error'
        END AS aws0100g,
muaggatt.aws0150wta,
CASE    WheN muaggatt.aws0150wta is NULL THEN NULL
        WHEN muaggatt.aws0150wta < 7.62 THEN 'v_low'
        WHEN muaggatt.aws0150wta >= 7.62 AND muaggatt.aws0150wta < 15.24 THEN 'low'
        WHEN muaggatt.aws0150wta >= 15.24 AND muaggatt.aws0150wta  < 22.86 THEN 'moderate'
        WHEN muaggatt.aws0150wta >= 22.86 AND muaggatt.aws0150wta  < 30.48 THEN 'high'
        WHEN muaggatt.aws0150wta >= 30.48 THEN 'v_high'
        ELSE 'error'
        END AS aws0150g,
muaggatt.mukey
FROM legend
INNER JOIN mapunit  ON legend.lkey = mapunit.lkey AND legend.areasymbol LIKE 'WI%'
INNER JOIN muaggatt ON mapunit.mukey = muaggatt.mukey
```

--Concerns: Worried about how root restrictive layers are handled in the MUAGGATT table. As of now, it looks like densic materials qualify as root restrictive but fragipans are not. However, it needs to be determined if this is the case. Update: Looks like hardness is a issue with fragipans. If hardness is null, it looks like it is considered root restrictive. There could be issues with how hardness is populated for fragipans. Also, Should look into the wtg Ave calc for mapunits that do not equal 100%.

****

Table Name: hydric

This query is designed to create a table with the percent of the mapunit that is designated as Hydric along with a categorical designation. Some of Wisconsin's surveys have minor components that do not have the component percent fields populated. Because of the unpopulated component percentages, the muaggatt table's hydclprs field does not accurately reflect the Hydric classification of some mapunits. In order to deal with the null values in the component percentages, certain assumptions about mapunit composition are made. The main assumption that is made is that in a consociation the major component comprises 85% of a mapunit. Therefore, if the mapunit has one 1 minor component and the component percentage is null, and the major is set to 100, then the major is adjusted to 85 percent and the minor is assumed to comprise 15 percent of the unit. Of course, if the major component has a percentage that is not equal to 100, that percentage is used and the remaining percentage is assigned to the minor component or components.

The current national grouping of Hydric classification does not meet the needs for most Wisconsin users. Therefore, WI has developed its own criteria. The groupings are as follows:

- The map unit is not rated for hydric classification --- For miscellaneous areas
- The map unit is not hydric --- (0%)
- The map unit is all Hydric -- (100%)
- The map unit is predominantly hydric -- (>=85%)
- The map unit is partially hydric --15-85% or any percent if the hydric component is a major component.
- The map unit has hydric inclusions -- <85% and no Hydric major components.

```
SELECT mapunit.mukey, component.majcompflag, component.cokey,  component.hydricrating, CAST(component.comppct_r AS
decimal(6,3)) AS comppct_r,
CASE    WHEN component.hydricrating =  'yes' THEN 1
        WHEN component.hydricrating = 'no' THEN 2
        WHEN component.hydricrating = 'unranked' THEN 3
        ELSE NULL
        END AS hydcode
INTO #TempHyd1
FROM legend
        INNER JOIN mapunit ON legend.lkey=mapunit.lkey AND legend.areasymbol Like 'WI%'
        LEFT JOIN component ON mapunit.mukey=component.mukey
```

```
SELECT component.mukey, CAST (Sum(component.comppct_r) AS DECIMAL(6,3)) AS notzipttl,
CAST(Count(component.comppct_r) AS DECIMAL(6,3)) AS notzipcnt, legend.areasymbol
INTO #TempHydric2
FROM legend INNER JOIN (mapunit INNER JOIN component ON mapunit.mukey = component.mukey) ON legend.lkey =
mapunit.lkey
GROUP BY component.mukey, legend.areasymbol
HAVING (((Sum(component.comppct_r)) Is Not Null) AND ((legend.areasymbol) Like 'WI%'))

SELECT component.mukey, CAST (Sum(component.comppct_r) AS DECIMAL(6,3)) AS notzipttl,
CAST(Count(component.comppct_r) AS DECIMAL(6,3)) AS notzipcnt, legend.areasymbol
INTO #TempHydricMaj
FROM legend INNER JOIN (mapunit INNER JOIN component ON mapunit.mukey = component.mukey) ON legend.lkey =
mapunit.lkey
GROUP BY component.mukey, legend.areasymbol, component.majcompflag
HAVING (((Sum(component.comppct_r)) Is Not Null) AND (component.majcompflag = 'Yes') AND ((legend.areasymbol) Like 'WI%'))


SELECT mapunit.mukey,  CAST (Count(component.mukey) AS decimal(6,3)) As nullct
INTO #TempHydric3
FROM legend INNER JOIN (mapunit INNER JOIN component ON mapunit.mukey = component.mukey) ON legend.lkey =
mapunit.lkey
Where component.comppct_r Is Null and legend.areasymbol like 'WI%'
GROUP BY mapunit.mukey

SELECT MIN(hydcode) As hydcodeMin, mukey
INTO #TempHydric4
FROM #TempHyd1
GROUP BY mukey



SELECT CAST ((CASE          WHEN #TempHydric2.notzipttl < 100 and #TempHydric3.nullct >=1 and #TempHyd1.comppct_r IS
NOT NULL THEN #TempHyd1.comppct_r
                    WHEN (#TempHyd1.comppct_r) IS NOT NULL and #TempHydric2.notzipttl = 100.000 and
#TempHydric2.notzipcnt = 1 and #TempHydric3.nullct >= 1 THEN 85.000
                    WHEN (#TempHyd1.comppct_r) IS NOT NULL and #TempHydric2.notzipttl = 100.000 and
#TempHydric2.notzipcnt > 1 and #TempHyd1.majcompflag = 'Yes' and #TempHydric3.nullct >= 1 THEN (#TempHyd1.comppct_r -
5.000)
                    WHEN ((#TempHyd1.comppct_r)is null and #TempHydric2.notzipttl < 100.000) THEN (100.000 -
#TempHydric2.notzipttl)/#TempHydric3.nullct
                    WHEN (#TempHyd1.comppct_r) IS NULL and #TempHydric2.notzipttl = 100.000 and #TempHydric2.notzipcnt
= 1 THEN 15.000/#TempHydric3.nullct
                    WHEN (#TempHyd1.comppct_r) IS NULL and #TempHydric2.notzipttl = 100.000 and #TempHydric2.notzipcnt
> 1 THEN (100.000 -(100.000 - (#TempHydricMaj.notzipcnt * 5.000)))/#TempHydric3.nullct
                    ELSE #TempHyd1.comppct_r
                    end) AS DECIMAL(6,3)) AS adjstpct,

          Case      WHEN #TempHyd1.majcompflag = 'yes' and #TempHyd1.hydcode = 3  THEN 1
                    WHEN #TempHyd1.majcompflag = 'yes' and #TempHyd1.hydricrating IS NULL and #TempHydric4.hydcodeMin
> 1 THEN 1
                    ELSE 0
                    END AS major,
          CASE    WHEN #TempHyd1.majcompflag = 'yes' and #TempHyd1.hydcode = 1  THEN 1
                    ELSE 0
                    END as minormajor1,

          CASE      WHEN #TempHyd1.hydricrating IS NULL AND #TempHydric4.hydcodeMin > 1 THEN 0
                    ELSE 1
                    END as null1,
#TempHyd1.mukey, #TempHyd1.cokey
INTO #TempHydric5

FROM (#TempHyd1 LEFT JOIN #TempHydric2 ON #TempHyd1.mukey = #TempHydric2.mukey) LEFT JOIN #TempHydric3 ON
#TempHyd1.mukey = #TempHydric3.mukey LEFT JOIN #TempHydric4 ON #TempHyd1.mukey = #TempHydric4.mukey LEFT JOIN
#TempHydricMaj ON #TempHyd1.mukey = #TempHydricMaj.mukey


SELECT MIN(null1) AS nullz, mukey, MAX(minormajor1) as minormajor
```

```
INTO #TempHydric6
FROM #TempHydric5
GROUP BY mukey

SELECT CASE     WHEN #TempHyd1.hydcode = 2 THEN 0
                WHEN #TempHyd1.hydcode = 1 THEN #TempHydric5.adjstpct
                WHEN #TempHydric5.major = 1 THEN 9999
                WHEN #TempHydric6.nullz = 0 THEN 9999
                ELSE 0
                END AS hydpct,
#TempHyd1.mukey, #TempHyd1.cokey
INTO #TempHydric7
FROM #TempHyd1
        LEFT JOIN #TempHydric5 ON #TempHyd1.cokey = #TempHydric5.cokey
        LEFT JOIN #TempHydric6 ON #TempHyd1.mukey = #TempHydric6.mukey

SELECT hydpct, mukey, cokey
INTO #TempHydric8
FROM #TempHydric7
GROUP BY hydpct, mukey, cokey

SELECT mukey, MAX(hydpct) as Maxhydpct, SUM(hydpct) AS Sumhydpct
INTO #TempHydric9
FROM #TempHydric8
GROUP BY mukey

SELECT CASE     WHEN MaXhydpct = 9999 THEN 9999
                ELSE CAST ((ROUND (Sumhydpct,0)) AS int)
                END AS hydpctsum,
        CASE    WHEN nullz = 0 THEN 'The map unit is not rated for hydric classification'
                WHEN Maxhydpct = 9999 THEN 'The map unit is not rated for hydric classification'
                WHEN Sumhydpct = 0 THEN 'The map unit is not hydric'
                WHEN Sumhydpct = 100 THEN 'The map unit is all hydric'
                WHEN Sumhydpct >= 85 THEN 'The map unit is predominantly  hydric'
                WHEN Sumhydpct <= 15 and minormajor > 0 THEN 'The map unit is partially hydric'
                WHEN Sumhydpct <= 15 THEN 'The map unit has hydric inclusions'
                ELSE 'The map unit is partially hydric'
                END AS hydrating,
#TempHydric9.mukey
FROM #TempHydric9
INNER JOIN #TempHydric7 on #TempHydric7.mukey = #TempHydric9.mukey
INNER JOIN #TempHydric6 on #TempHydric6.mukey = #TempHydric9.mukey
GROUP BY #TempHydric9.mukey, Maxhydpct, Sumhydpct, nullz, minormajor
```

****

K and T

This query returns properties associated with soil erosion. The T factor is the maximum amount of erosion at which the quality of a soil as a medium for plant growth can be maintained. This table produces the T factor aggregated in two ways, by the dominant condition method and by a most limiting (lowest T factor of the major components) method.

The K factor, in this case the Kf factor, is an erodibility factors which quantify the soil detachment by water. The Kf factor only considers the fine-earth fraction of the soil. Because a portion of the state is dominated by forested land use, many soils in Wisconsin have been populated with a “duff” layer. This duff does not have a Kf factor assigned. Since the thin duff layer is almost always destroyed by plowing, we report the Kfactor of the first mineral horizon for these soils that have a thin duff layer. Soils without a duff layer, the Kf factor is reported as the surface horizon. In this table, the Kf factor is aggregated in two ways, by the dominant condition method and by a a most limiting (highest surface (as defined abouve) Kf factor of the major components) method.

Fields:

Mukey:	A non-connotative string of characters used to uniquely identify a record in the Mapunit table.
tfactDomCond	T factor aggregated using the Dominant Condition Aggregation Method
TfactMostLmt	T factor aggregated using the Most Limiting Aggregation Method
kfffactDomCond	Kf factor of the first mineral layer, Dominant Condition
KffactMostLmt	Kf factor of the first mineral layer, Most limiting

---

```
SELECT mapunit.mukey, component.comppct_r, chorizon.hzdept_r, component.cokey, chorizon.chkey,  component.majcompflag, chorizon.kffact
INTO #NoDuffK1
FROM (legend INNER JOIN (mapunit LEFT JOIN component ON mapunit.mukey = component.mukey) ON legend.lkey = mapunit.lkey) LEFT JOIN chorizon  ON component.cokey = chorizon.cokey
WHERE (((chorizon.hzdept_r)=(SELECT Min(chorizon.hzdept_r) AS MinOfhzdept_r
FROM chorizon LEFT JOIN chtexturegrp ON chorizon.chkey = chtexturegrp.chkey
Where chtexturegrp.texture Not In ('SPM','HPM', 'MPM')  AND chtexturegrp.rvindicator='Yes' AND component.cokey = chorizon.cokey )) AND ((legend.areasymbol) Like 'WI%') AND ((component.majcompflag)='Yes'))
ORDER BY legend.areasymbol, mapunit.musym, chorizon.hzdept_r


SELECT #NoDuffK1.mukey, Sum(#NoDuffK1.comppct_r) AS SumOfcomppct_r, #NoDuffK1.kffact
INTO #NoDuffK2
FROM #NoDuffK1
GROUP BY #NoDuffK1.mukey, #NoDuffK1.kffact

SELECT #NoDuffK2.mukey, Max(#NoDuffK2.SumOfcomppct_r) AS MaxOfSumOfcomppct_r
INTO #NoDuffK3
FROM #NoDuffK2
GROUP BY #NoDuffK2.mukey

SELECT #NoDuffK3.mukey, #NoDuffK2.kffact
INTO  #NoDuffk
FROM #NoDuffK2 INNER JOIN #NoDuffK3 ON (#NoDuffK2.SumOfcomppct_r = #NoDuffK3.MaxOfSumOfcomppct_r) AND (#NoDuffK2.mukey = #NoDuffK3.mukey);

WITH #FirstofNoDuffk1 AS (Select mukey, kffact, rn = row_number() OVER (PARTITION BY mukey ORDER BY kffact desc) From #NoDuffk)

Select kffact as KffactDomCond, mukey
INTO #FirstofNoDuffk
From #FirstofNoDuffk1
Where rn=1

SELECT #NoDuffK1.mukey, MAX( #NoDuffK1.kffact) as KffactMostLmt
INTO #NoDuffMLK
FROM #NoDuffK1
GROUP BY #NoDuffK1.mukey



SELECT Min(component.tfact) AS MinTfact, component.mukey
INTO #Tfact
FROM legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN component ON mapunit.mukey = component.mukey AND component.majcompflag='yes'
GROUP BY component.mukey

SELECT component.mukey, Sum(component.comppct_r) AS SumOfcomppct_r, component.tfact
INTO #TfactDom1
FROM legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN component ON mapunit.mukey = component.mukey AND component.majcompflag='yes'
GROUP BY component.mukey, component.tfact

SELECT #TfactDom1.mukey, Max(#TfactDom1.SumOfcomppct_r) AS MaxOfSumOfcomppct_r
INTO #TfactDom2
FROM #TfactDom1
GROUP BY #TfactDom1.mukey

SELECT #TfactDom2.mukey, #TfactDom1.tfact
INTO  #TfactDom
FROM #TfactDom1 INNER JOIN #TfactDom2 ON (#TfactDom1.SumOfcomppct_r = #TfactDom2.MaxOfSumOfcomppct_r) AND (#TfactDom1.mukey = #TfactDom2.mukey);

WITH #FirstofTfactDom1 AS (Select mukey, tfact, rn = row_number() OVER (PARTITION BY mukey ORDER BY tfact ASC) From #TfactDom)

Select Tfact as tfactDomCond, mukey
```

INTO #FirstofTfact
From #FirstofTfactDom1
Where rn=1

SELECT #Tfact.mukey, #FirstofTfact.tfactDomCond, #Tfact.MinTfact as TfactMostLmt, #FirstofNoDuffk.kffactDomCond, #NoDuffMLK.KffactMostLmt
FROM #Tfact
LEFT JOIN #FirstofNoDuffk ON #Tfact.mukey = #FirstofNoDuffk.mukey
LEFT JOIN #NoDuffMLK ON #Tfact.mukey = #NoDuffMLK.mukey
LEFT JOIN #FirstofTfact ON #Tfact.mukey = #FirstofTfact.mukey
-- No duff not working for beaverbay.

Nutrient Management Planning

****

This Script is designed to create a state wide version of Appendix 1 of the NRCS Conservation Planning Tech note 1. The result will list Wisconsin soils that are more susceptible to groundwater contamination from leaching of nitrogen based on NRCS soil survey data. The soils list covers all counties. The 72 counties listed are arranged alphabetically beginning with Adams County. The soils within each county are arranged alphabetically or numerically by soil map unit symbol. Only Mapunits that are restricted in some fashion are included in the list. To constrain the results to a specific county replace all 'WI%' with the appropriate fips code for the county. An example would be "WI007' for Bayfield County.

Use the following information to interpret components of the table:

Field Name	Description	Attributes
County	The name of one of 72 Wisconsin Counties	Ex. Bayfield
Symbol	Soil Map unit symbol found in Web Soil Survey	Ex. 713B or DeC2
Restriction	Soil characteristics that create a high potential for nitrate leaching to groundwater	P - indicates high permeability R - indicates any type of bedrock less than 20 inches from the surface W - indicates an **apparent water table** less than 12 inches from the surface + - indicates the map unit may have multiple restrictions. An on-site investigation is needed to identify which restrictions may actually be present.
MapUnitName	The name of the Mapunit	Ex. Allendale loamy fine sand, 0 to 3 percent slopes

Soils with no entry in the restrictions column are less likely to have NRCS 590 restrictions for wetness, bedrock, or permeability (leaching). Other soil properties affecting NRCS 590 suitability, such as % slope and surface texture were not evaluated.

```
SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, mapunit.mukey
INTO #Temp
FROM (mapunit INNER JOIN component ON mapunit.mukey = component.mukey) Inner JOIN cointerp ON component.cokey =
cointerp.cokey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=0) AND
((component.majcompflag)='yes'))

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, cointerp.rulename, mapunit.mukey
INTO #TempPerm
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey=cointerp.cokey) ON
mapunit.mukey=component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, cointerp.rulename, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=1) AND
((component.majcompflag)='yes') AND ((cointerp.rulename)='Permeable Soils Subrule (WI)'));

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, cointerp.rulename, mapunit.mukey
INTO #TempRock
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey=cointerp.cokey) ON
mapunit.mukey=component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, cointerp.rulename, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=1) AND
((component.majcompflag)='yes') AND ((cointerp.rulename)='Bedrock Subrule  (WI)'));

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, cointerp.rulename, mapunit.mukey
INTO #TempWater
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey = cointerp.cokey) ON mapunit.mukey =
component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, cointerp.rulename, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=1) AND
((component.majcompflag)='yes') AND ((cointerp.rulename)='Apparent H20 Subrule (WI)'));

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, mapunit.mukey, cointerp.interphrc
INTO #TempNotRated
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey = cointerp.cokey) ON mapunit.mukey =
component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, mapunit.mukey, cointerp.interphrc
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=0) AND
((component.majcompflag)='yes') AND ((cointerp.interphrc)='Not Rated'));

SELECT #Temp.musym, #Temp.mukey,
CASE
        WHEN #TempWater.MaxOfinterplr = 1
        THEN 'w'
        ELSE '' END AS W,
CASE
        WHEN #TempRock.MaxOfinterplr = 1
        THEN 'r'
        ELSE ''
        END AS R,
CASE
        WHEN #TempPerm.MaxOfinterplr = 1
        THEN 'p'
        ELSE ''
        END AS P,
CASE
        WHEN #TempNotRated.interphrc = 'Not Rated'
        THEN '+'
        ELSE ''
        END AS NotRated
INTO #TempLetter
FROM #Temp
        LEFT JOIN #TempPerm ON #Temp.mukey = #TempPerm.mukey
        LEFT JOIN #TempRock ON #Temp.mukey = #TempRock.mukey
        LEFT JOIN #TempWater ON #Temp.mukey = #TempWater.mukey
        LEFT JOIN #TempNotRated ON #Temp.mukey = #TempNotRated.mukey
```

```
ORDER by #Temp.musym

SELECT laoverlap.areaname AS County, mapunit.musym AS Symbol, #TempLetter.W + #TempLetter.P + #TempLetter.r +
#TempLetter.NotRated as Restriction, mapunit.muname as MapUnitName
FROM legend
        Inner JOIN laoverlap ON legend.lkey = laoverlap.lkey
        LEFT JOIN mapunit ON legend.lkey = mapunit.lkey
        LEFT JOIN #TempLetter ON mapunit.mukey = #TempLetter.mukey
WHERE  laoverlap.areasymbol LIKE 'WI%' AND laoverlap.areatypename LIKE 'County%' AND #TempLetter.W <> ''
        OR laoverlap.areasymbol LIKE 'WI%' AND laoverlap.areatypename LIKE 'County%' AND #TempLetter.P <> ''
                OR laoverlap.areasymbol LIKE 'WI%' AND laoverlap.areatypename LIKE 'County%' AND #TempLetter.R <> ''
                        OR laoverlap.areasymbol LIKE 'WI%' AND laoverlap.areatypename LIKE 'County%' AND
#TempLetter.NotRated <> ''
Order BY County, legend.areasymbol, mapunit.museq
```

****

Use this script to generate a statewide table compatible with GIS for interpreting the 590 Standard.

```
SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, mapunit.mukey
INTO #Temp
FROM (mapunit INNER JOIN component ON mapunit.mukey = component.mukey) Inner JOIN cointerp ON component.cokey =
cointerp.cokey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=0) AND
((component.majcompflag)='yes'))

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, cointerp.rulename, mapunit.mukey
INTO #TempPerm
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey=cointerp.cokey) ON
mapunit.mukey=component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, cointerp.rulename, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=1) AND
((component.majcompflag)='yes') AND ((cointerp.rulename)='Permeable Soils Subrule (WI)'));

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, cointerp.rulename, mapunit.mukey
INTO #TempRock
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey=cointerp.cokey) ON
mapunit.mukey=component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, cointerp.rulename, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=1) AND
((component.majcompflag)='yes') AND ((cointerp.rulename)='Bedrock Subrule  (WI)'));

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, Max(cointerp.interplr) AS MaxOfinterplr,
component.majcompflag, cointerp.rulename, mapunit.mukey
INTO #TempWater
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey = cointerp.cokey) ON mapunit.mukey =
component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, cointerp.rulename, mapunit.mukey
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=1) AND
((component.majcompflag)='yes') AND ((cointerp.rulename)='Apparent H20 Subrule (WI)'));

SELECT mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, mapunit.mukey, cointerp.interphrc
INTO #TempNotRated
FROM mapunit INNER JOIN (component INNER JOIN cointerp ON component.cokey = cointerp.cokey) ON mapunit.mukey =
component.mukey
GROUP BY mapunit.musym, cointerp.mrulename, cointerp.ruledepth, component.majcompflag, mapunit.mukey, cointerp.interphrc
HAVING (((cointerp.mrulename)='AWM - Sensitive Soil Features (WI)') AND ((cointerp.ruledepth)=0) AND
        ((component.majcompflag)='yes') AND ((cointerp.interphrc)='Not Rated'));

SELECT #Temp.musym, #Temp.mukey,
CASE
        WHEN #TempWater.MaxOfinterplr = 1
        THEN 'w'
        ELSE '' END AS W,
CASE
        WHEN #TempRock.MaxOfinterplr = 1
        THEN 'r'
        ELSE ''
        END AS R,
CASE
        WHEN #TempPerm.MaxOfinterplr = 1
        THEN 'p'
        ELSE ''
        END AS P,
CASE
        WHEN #TempNotRated.interphrc = 'Not Rated'
        THEN '+'
        ELSE ''
        END AS NotRated
INTO #TempLetter
FROM #Temp
```

```
        LEFT JOIN #TempPerm ON #Temp.mukey = #TempPerm.mukey
        LEFT JOIN #TempRock ON #Temp.mukey = #TempRock.mukey
        LEFT JOIN #TempWater ON #Temp.mukey = #TempWater.mukey
        LEFT JOIN #TempNotRated ON #Temp.mukey = #TempNotRated.mukey
ORDER by #Temp.musym

SELECT  mapunit.musym AS Symbol, #TempLetter.W + #TempLetter.P + #TempLetter.r + #TempLetter.NotRated as Restriction,
        mapunit.muname as MapUnitName , mapunit.mukey
FROM legend
        LEFT JOIN mapunit ON legend.lkey = mapunit.lkey
        LEFT JOIN #TempLetter ON mapunit.mukey = #TempLetter.mukey
WHERE  legend.areasymbol LIKE 'WI%'
ORDER BY  legend.areasymbol, mapunit.museq
```

Land Evaluation

****

The Land Evaluation (LE) system was developed by the USDA in 1980 and is now widely used throughout the U.S. LE provides a systematic and objective way to evaluate and numerically rank soils for their relative value for a specific land use.

LEag is a default LE score for use in Wisconsin. Higher numbers mean greater value for common cultivated row crops. LEag scores reflect the total yield and value of suitable crops, and the economic and environmental costs of producing a crop. Possible LEag values range from 0 to 100

Three other soil data elements are combined to produce the LEag score. This produces a rating that reflects the most important soil considerations for agricultural use:

Prime farmland class - rates the major physical & chemical soil properties affecting agricultural use.

Improved land capability class - rates the risk of environmental damage (eg: soil erosion, off-site damage from sediment, nutrient, and pesticide runoff or leaching) and the degree of management concerns and limitations for agricultural use.

Agricultural Productivity Index - rates the potential yield and value of agricultural crops.

Each of these data elements is assigned a point score from 0 to 100:

prime farmland	100
prime where drained	70
prime where not flooded	40
prime where drained and not flooded	30

capability class 1 = 100 ; 2 = 90 ; 3 = 70 ; 4 = 50 ; 5 or 6 = 20 ; 7 = 10 ; 8 = 0

Productivity Index (already on a 0 to 100 scale)

A weighting factor is then applied to each of the 3 data element scores to reflect their relative importance. The default Wisconsin LEag score is calculated using the formula:

LEag = (prime score x 0.15) + (capability score x 0.30) + (productivity index x 0.55)

With the assistance of a soil scientist, the data elements used, the point scores assigned to the point scores, and the weighting factor for each point score can be adjusted to reflect local priorities and conditions.

Sample calculation:

soil property	score		weight		LE score
Prime if drained	70	x	0.15	=	10.5
Capability class = 2	90	x	0.30	=	27.0
Prod. Index = 86	86	x	0.55	=	47.3
					84.8
TOTAL LEag score for the map unit =					= **85**

Fields:

Areasymbol	*The symbol used to identify a unique Soil Survey Area. Most commonly it is a concatenation of WI and the FIPS code eg: WI001 = Adams County. However, Wisconsin has 3 dual county soil surveys. WI600 is the symbol for Calumet and Manatowoc Counties, WI601 is Kenosha and Racine, and WI602 is Milwaukee and Waukasha.*
musym	*The symbol used to uniquely identify the soil mapunit in the soil survey.*
muname	*Correlated name of the mapunit*
StateNormLEag	*This field reports the LE score, with the best mapunit in the **STATE** receiving a score of 100*
CountyNormLEag	*This field reports the LE score, with the best mapunit in the **SOIL SURVEY AREA** receiving a score of 100*
mukey	*A non-connotative string of characters used to uniquely identify a record in the Mapunit table.*
mukey_num	*A non-connotative integer used to uniquely identify a record in the Mapunit table*

```
--LE Script
-------------------------------------------------------------------------------
--PI Portion
-------------------------------------------------------------------------------
--Determine the Max yeild for the 9 indicator crops

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxAlfalfaYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Alfalfa hay') AND ((mucropyld.yldunits)='Tons'))

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxCornYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='corn') AND ((mucropyld.yldunits)='bu'))

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxGrassCloverPasYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
WHERE (((mucropyld.mukey)<>'422324'))
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Grass-clover') AND ((mucropyld.yldunits)='AUM'))

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxKyBlueYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Kentucky bluegrass') AND ((mucropyld.yldunits)='AUM'))

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxOatYld
```

```
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='oats') AND ((mucropyld.yldunits)='bu'))

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxRCloverYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
WHERE (((mucropyld.mukey)<>'422324'))
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Red clover hay') AND ((mucropyld.yldunits)='Tons'))

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxReedCanaryYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Reed canarygrass') AND ((mucropyld.yldunits)='AUM'));

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxSmoothBromYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
WHERE (((mapunit.mukey)<>'423320')) and legend.areasymbol LIKE 'WI%'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Smooth bromegrass') AND ((mucropyld.yldunits)='AUM'));

SELECT mucropyld.cropname, mucropyld.yldunits, Max(mucropyld.nonirryield_r) AS MaxOfnonirryield_r
INTO #MaxWheatYld
FROM legend INNER JOIN (mapunit INNER JOIN mucropyld ON mapunit.mukey = mucropyld.mukey) ON legend.lkey =
mapunit.lkey AND Legend.areasymbol LIKE 'WI%'
Where mapunit.mukey <> '423777'
GROUP BY mucropyld.cropname, mucropyld.yldunits
HAVING (((mucropyld.cropname)='Winter Wheat') AND ((mucropyld.yldunits)='bu'));
--------------------------------------------------------------------------------
SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxAlfalfaYld.MaxOfnonirryield_r, mucropyld.nonirryield_r,
Round(([mucropyld].[nonirryield_r]/[#MaxAlfalfaYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey
INTO #AlfalfaNorm
FROM  #MaxAlfalfaYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
--GROUP BY mapunit.musym, mucropyld.cropname, mucropyld.yldunits,
Round(([mucropyld].[nonirryield_r]/[#MaxAlfalfaYld].[MaxOfnonirryield_r]*100),0), mapunit.mukey
where (((mucropyld.cropname)='Alfalfa hay') AND ((mucropyld.yldunits)='tons'))


SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxCornYld.MaxOfnonirryield_r, mucropyld.nonirryield_r,
Round(([mucropyld].[nonirryield_r]/[#MaxCornYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey
INTO #CornNorm
FROM  #MaxCornYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='corn') AND ((mucropyld.yldunits)='bu'))

SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxGrassCloverPasYld.MaxOfnonirryield_r,
mucropyld.nonirryield_r, Round(([mucropyld].[nonirryield_r]/[#MaxGrassCloverPasYld].[MaxOfnonirryield_r]*100),0) AS Expr1,
mapunit.mukey
INTO #GrassCloverNorm
FROM  #MaxGrassCloverPasYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='Grass-clover') AND ((mucropyld.yldunits)='AUM'))

SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxKyBlueYld.MaxOfnonirryield_r, mucropyld.nonirryield_r,
Round(([mucropyld].[nonirryield_r]/[#MaxKyBlueYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey
INTO #KyBlueNorm
FROM  #MaxKyBlueYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
```

where (((mucropyld.cropname)='Kentucky bluegrass') AND ((mucropyld.yldunits)='AUM'))

*SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxOatYld.MaxOfnonirryield_r, mucropyld.nonirryield_r, Round(([mucropyld].[nonirryield_r]/[#MaxOatYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey*
INTO #OatsNorm
FROM #MaxOatYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='oats') AND ((mucropyld.yldunits)='bu'))

*SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxRCloverYld.MaxOfnonirryield_r, mucropyld.nonirryield_r, Round(([mucropyld].[nonirryield_r]/[#MaxRCloverYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey*
INTO #RcloverNorm
FROM #MaxRCloverYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='Red clover hay') AND ((mucropyld.yldunits)='Tons'))

*SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxReedCanaryYld.MaxOfnonirryield_r, mucropyld.nonirryield_r, Round(([mucropyld].[nonirryield_r]/[#MaxReedCanaryYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey*
INTO #ReedNorm
FROM #MaxReedCanaryYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='Reed canarygrass') AND ((mucropyld.yldunits)='AUM'))

*SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxSmoothBromYld.MaxOfnonirryield_r, mucropyld.nonirryield_r, Round(([mucropyld].[nonirryield_r]/[#MaxSmoothBromYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey*
INTO #BromeNorm
FROM #MaxSmoothBromYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='Smooth bromegrass') AND ((mucropyld.yldunits)='AUM'))

*SELECT mapunit.musym, mucropyld.cropname, mucropyld.yldunits, #MaxWheatYld.MaxOfnonirryield_r, mucropyld.nonirryield_r, Round(([mucropyld].[nonirryield_r]/[#MaxWheatYld].[MaxOfnonirryield_r]*100),0) AS Expr1, mapunit.mukey*
INTO #WheatNorm
FROM #MaxWheatYld, legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
INNER JOIN mucropyld ON mapunit.mukey=mucropyld.mukey
where (((mucropyld.cropname)='Winter Wheat') AND ((mucropyld.yldunits)='bu'))

SELECT mapunit.museq, mapunit.mukey, legend.areasymbol, mapunit.muname, mapunit.musym, #AlfalfaNorm.Expr1 as AlfalfaNorm,#CornNorm.Expr1 as CornNorm,#GrassCloverNorm.Expr1 as GrassCloverNorm,#KyBlueNorm.Expr1 as KyBlueNorm,#OatsNorm.Expr1 as OatsNorm,#RcloverNorm.Expr1 as RcloverNorm,#ReedNorm.Expr1 as ReedNorm,#BromeNorm.Expr1 as BromeNorm,#WheatNorm.Expr1 as WheatNorm
INTO #NormalMaster
From legend
INNER JOIN mapunit ON legend.lkey = mapunit.lkey AND legend.areasymbol Like 'WI%'
LEFT JOIN #AlfalfaNorm on mapunit.mukey=#AlfalfaNorm.mukey
LEFT JOIN #CornNorm on mapunit.mukey=#CornNorm.mukey
LEFT JOIN #GrassCloverNorm on mapunit.mukey=#GrassCloverNorm.mukey
LEFT JOIN #KyBlueNorm on mapunit.mukey=#KyBlueNorm.mukey
LEFT JOIN #OatsNorm on mapunit.mukey=#OatsNorm.mukey
LEFT JOIN #RcloverNorm on mapunit.mukey=#RcloverNorm.mukey
LEFT JOIN #ReedNorm on mapunit.mukey= #ReedNorm.mukey
LEFT JOIN #BromeNorm on mapunit.mukey=#BromeNorm.mukey
LEFT JOIN #WheatNorm on mapunit.mukey=#WheatNorm.mukey

SELECT mapunit.mukey, count(cropname) as cntsrps
INTO #crops
From legend
INNER JOIN mapunit on legend.lkey = mapunit.lkey AND areasymbol like 'WI%'
INNER JOIN mucropyld on mapunit.mukey = mucropyld.mukey
group by mapunit.mukey

```
--------------------------------------------------------------------------------

SELECT  areasymbol, musym, muname,
        CASE    WHEN NM.CornNorm IS NOT NULL AND NM.AlfalfaNorm IS NOT NULL AND NM.OatsNorm IS NOT NULL
THEN (NM.CornNorm + NM.AlfalfaNorm + NM.OatsNorm)/3
                WHEN NM.CornNorm IS NOT NULL AND NM.OatsNorm IS NOT NULL THEN ((NM.CornNorm +
NM.OatsNorm) * .95)/2
                WHEN NM.CornNorm IS NOT NULL AND NM.AlfalfaNorm IS NOT NULL THEN ((NM.CornNorm +
NM.AlfalfaNorm) * .95)/2
                WHEN NM.CornNorm IS NOT NULL THEN (NM.CornNorm * .90)

                WHEN NM.AlfalfaNorm IS NOT NULL AND NM.OatsNorm IS NOT NULL THEN ((NM.AlfalfaNorm +
NM.OatsNorm) * .8)/2
                WHEN NM.OatsNorm IS NOT NULL THEN NM.OatsNorm * .8
                WHEN NM.WheatNorm IS NOT NULL THEN NM.WheatNorm * .8

                WHEN NM.AlfalfaNorm IS NOT NULL THEN NM.AlfalfaNorm * .7
                WHEN NM.RcloverNorm IS NOT NULL THEN NM.RcloverNorm * .6

                WHEN NM.GrassCloverNorm IS NOT NULL THEN NM.GrassCloverNorm * .4
                WHEN NM.BromeNorm IS NOT NULL THEN NM.BromeNorm * .4
                WHEN NM.ReedNorm IS NOT NULL THEN NM.ReedNorm * .4
                WHEN NM.KyBlueNorm IS NOT NULL THEN NM.KyBlueNorm * .2
                WHEN cntsrps > 0 THEN 15
        ELSE 0
        END AS PI, nm.mukey, museq
INTO #Pi
FROM #NormalMaster NM
Left JOIN #crops crops on crops.mukey = NM.mukey
Order By areasymbol, museq

SELECT Max(PI) As StateMax
INTO #StateMax
FROM #Pi

SELECT areasymbol, musym, muname,ROUND((PI/StateMax * 100),0) AS StateNormPI, mukey
INTO #StateNormal
FROM #Pi, #StateMax


SELECT Max(PI) As AreaMax, areasymbol
INTO #AreasymMax
FROM #Pi
GROUP BY areasymbol

SELECT Pi.areasymbol, Pi.musym, Pi.muname, ROUND((Pi.PI/AreaMax *100),0) AS CountyNormPI, sn.StateNormPI, Pi.mukey,
pi.museq
INTO #PIForLE
FROM #Pi Pi
        INNER JOIN #StateNormal sn on Pi.mukey = sn.mukey
        INNER JOIN #AreasymMax On Pi.areasymbol = #AreasymMax.areasymbol
ORDER by Pi.areasymbol, Pi.museq

--------------------------------------------------------------------------------
--LCC Score
--------------------------------------------------------------------------------
SELECT l.areasymbol, m.musym, m.muname, m.mukey, CAST(c.comppct_r AS decimal(6,3)) AS comppct_r, c.majcompflag,
CAST(c.nirrcapcl AS decimal(6,3)) AS nirrcapcl
INTO #Cap1
FROM legend l
INNER JOIN mapunit m ON l.lkey = m.lkey
INNER JOIN component c ON m.mukey = c.mukey
WHERE l.areasymbol Like 'wi%' AND c.majcompflag='yes'

Select SUM(comppct_r) as SumPct, mukey
INTO #Cap2
FROM #Cap1
GROUP by mukey
```

```
SELECT CAST((comppct_r/SumPct * Nirrcapcl) AS decimal(6,3)) As ave, #Cap1.mukey
INTO #Cap3
FROM #Cap1
INNER JOIN #Cap2 on #cap1.mukey = #Cap2.mukey

SELECT CAST (Round(SUM(ave),0) AS INT) as WtAveLCC, mukey
INTO #Cap4
FROm #Cap3
Group by mukey

SELECT  CASE      WHEN WtAveLCC = 1 THEN 100
                  WHEN WtAveLCC = 2 THEN 90
                  WHEN WtAveLCC = 3 THEN 70
                  WHEN WtAveLCC = 4 THEN 50
                  WHEN WtAveLCC <= 6 THEN 20
                  WHEN WtAveLCC = 7 THEN 10
                  WHEN WtAveLCC = 8 THEN 0
          ELSE 0
          END AS LCCScore, mukey, WtAveLCC
INTO #LCCforLE
FROM #Cap4
--nulls in lcc will be given a class of 0
-------------------------------------------------------------------------------
--FarmClass
-------------------------------------------------------------------------------
SELECT  CASE      WHEN m.farmlndcl = 'All areas are prime farmland' THEN 100
                  WHEN m.farmlndcl = 'Prime farmland if drained' THEN 70
                  WHEN m.farmlndcl = 'Prime farmland if protected from flooding or not frequently flooded during the growing
season' THEN 40
                  WHEN m.farmlndcl = 'Prime farmland if drained and either protected from flooding or not frequently flooded
during the growing season' THEN 30
          ELSE 0
          END AS pflScore, mukey
INTO #PflForLE

FROM legend l
INNER JOIN mapunit m ON l.lkey = m.lkey and l.areasymbol Like 'WI%'

-------------------------------------------------------------------------------
--Final LE shebang
-------------------------------------------------------------------------------

SELECT Pi.areasymbol, Pi.musym, Pi.muname, ((pfl.pflScore * .15) + (lcc.LCCScore * .30) + (pi.CountyNormPI * .55)) AS
CountyLE, ((pfl.pflScore * .15) + (lcc.LCCScore * .30) + (pi.StateNormPI * .55)) AS StateLE, pi.mukey, pi.museq
INTO #LE1
FROM #PflForLE pfl
          INNER JOIN #LCCforLE lcc ON lcc.mukey = pfl.mukey
          INNER JOIN #PIForLE pi On pi.mukey = pfl.mukey

oRDER by Pi.areasymbol, Pi.museq

SELECT MAX(StateLE) as StateMaxLE
INTO #StateLEMax
From #LE1

SELECT MAX(CountyLE) as CountyMaxLE, areasymbol
INTO #CountyLEMax
FROM #LE1
Group by Areasymbol

SELECT LE1.areasymbol, LE1.musym, LE1.muname, Round((StateLE/StateMaxLE *100),0) AS StateNormLEag,
Round((CountyLE/CountyMaxLE * 100),0) AS CountyNormLEag, LE1.mukey, cast (le1.mukey as int) as mukey_num
FROM  #StateLEMax, #LE1 LE1
INNER JOIN #CountyLEMax cm ON cm.areasymbol = LE1.areasymbol
oRDER by LE1.areasymbol, LE1.museq
```

WI DNR Tables

****

Table Name: WIDnrDrainRef

****

SELECT
cast (mu.mukey as int) as MUKEY_NUM_CODE, drclassdcd as DRCLASSDCD, drclasswettest as DRCLASSWET
FROM sacatalog sac
INNER JOIN legend l ON l.areasymbol = sac.areasymbol and l.areasymbol LIKE 'WI%'
INNER JOIN mapunit mu ON mu.lkey = l.lkey
INNER JOIN muaggatt muag on muag.mukey = mu.mukey

****

Table Name: WiMapunitDNR.txt

****

SELECT
l.areasymbol, mu.musym as MUSYM, mu.muname as MUNAME, mu.mukind as MUKIND, mu.mukey as MUKEY,mu.farmlndcl as FARMLNDCL, cast (mu.mukey as INT) as MUKEY_NUM_CODE -- attributes from table "mapunit"
FROM sacatalog sac
INNER JOIN legend l ON l.areasymbol = sac.areasymbol and l.areasymbol LIKE 'WI%'
INNER JOIN mapunit mu ON mu.lkey = l.lkey
INNER JOiN muaggatt muag on muag.mukey = mu.mukey

****

Table Name: WiDnrAWC

****

SELECT
cast (mu.mukey as int) as MUKEY_NUM_CODE, aws025wta, aws050wta, aws0100wta,aws0150wta
FROM sacatalog sac
INNER JOIN legend l ON l.areasymbol = sac.areasymbol and l.areasymbol LIKE 'WI%'
INNER JOIN mapunit mu ON mu.lkey = l.lkey
INNER JOiN muaggatt muag on muag.mukey = mu.mukey

Action Items:
Finish Hydric table.
Fix Texture
Review AWC calculations to determine the proper interpretation of Root Restricive and the wtg Ave calc.
Revise Mapunit table to include short soils descriptions
Create other hyd group aggregations
Create other texture aggregation and create a separate texture table
Create LE calc and erosion factors
Create engineering interp table.
Improve metadata for bed, wt.
Define aggregation methods further.
Water Table Type
RUSLE2 Attributes

Revision Notes .7

Fixed 590 Slopes.
Added WI to the constraint section of the no duff query
Fixed Texture, notes included.