**Task 3.15.6 Neighborhood/Subdivision/Area Names Mar 30, 2015**

Different areas of the city are referred to using a variety of name. Our goal is to capture the names used and try to maintain some consistency. The names are first compiled by city and then by polygon.

Because the fields are free form entry there is little consistency. The Subdivision names are preprocessed to remove excess text from the string.

The approach here is to count the number of time a subdivision name is used and if they are referenced more than x times the name is added to the lookup table.

Start Processing

* Write process start record to the Process\_Status\_Log table

Process name : Compile\_Subdivision\_Names

Replaces: Compile\_Area\_Names and getretsdata\_dictneighbourhood.php

(these two scripts are no longer required)

Process id: 1.25

* Select all Real\_Location records that have not been processed

SELECT l.Location\_id,

l.Sub\_id,

l.Subdivision ,

l.City\_id

FROM Real\_Location l, tb\_City c

WHERE l.City\_id = c.City\_id AND c.Include=1

AND l.dict\_Sub = 0

AND l.City\_id = 970 Note: City\_id = 970 just for testing

AND NOT ISNULL(l.Subdivision)

ORDER BY 2, 3

* For each location record
  + Perform “Process A” (See below)
* Once Real\_Location.Subdivision field has been processed

UPDATE Real\_Location

SET dict\_Sub = 1

WHERE Location\_id = <id of processed record>

* For each Location\_Subdivision record
  + Perform “Process A” (See below)

Note: This step is required to process any changes in the tb\_Subdivision\_Mapping against subdivision names that have already been processes.

* Add commonly used Subdivision names to the tb\_Subdivision\_City table

INSERT INTO tb\_Subdivision\_City

(Subdivision, City\_id, Count)

SELECT Subdivision, City\_id, Count(\*) AS Cnt

FROM Location\_Subdivision

WHERE Sub\_id=0

GROUP BY City\_id, Subdivision

HAVING Cnt >= n

n = SystemSettings where keyname=”AREANAME” and subkeyname=”THRESHOLD”)

UPDATE Location\_Subdivision l, tb\_Subdivision\_City t

SET l.Sub\_id = t.Sub\_id

WHERE l.Subdivision = t.Subdivision

AND l.City\_id = t.City\_id

AND l.Sub\_id=0

UPDATE tb\_Subdivision\_City

SET Accepted = 1

WHERE Count >= n2

n = SystemSettings where keyname=”AREANAME” and subkeyname=”THRESHOLD\_2”)

* Revisit the Location\_Subdivision records

In this step we use the subdivision names that are the most commonly used (ie. tb\_Subdivision\_City records where Accepted=1) and match them with the subdivision names in the Location\_Subdivision table.

SELECT Sub\_id, Subdivision, City\_id,

Length(Subdivision) AS Len

FROM tb\_Subdivision\_City

WHERE Accepted=1

ORDER BY 3, 4 DESC, 2

For each subdivision record

* Lookup the name string to find an alternate (tb\_Subdivision\_Mapping).
* Find matching subdivision names in tb\_Subdivision\_City table where City\_id=<City\_id> and Accepted=0

For each match found

* + Update the Location\_Subdivision records with the new tb\_Sub\_id

Note: Need to check if Sub\_id already used for a Location\_id

* + Delete the match tb record
* Find matching subdivision names in Location\_Subdivision table where City\_id = <City\_id> and Sub\_id = 0
  + Check if tb.Sub\_id already used for Location\_id

If Yes, Update counter and delete matched record

If No, Update Sub\_id = tb.Sub\_id

* Update all the tb\_Subdivision\_City table record count
* Process Polygon

SELECT c.Polygon\_id, s.Subdivision, SUM(Count)

FROM Location\_Subdivision s, Real\_Polygon p, Polygon\_Center c

WHERE s.Location\_id = p.Location\_id

AND p.Polygon\_id = c.Polygon\_id

GROUP BY c.Polygon\_id, s.Subdivision

ORDER BY c.Polygon\_id, s.Subdivision

For each Polygon\_id Subdivision

* + Update the Polygon\_Subdivision table

SELECT Polygon\_id

FROM Polygon\_Subdivision

WHERE Polygon\_id = <Polygon\_id>

AND Subdivision = <Subdivision>

If record exists

* + - Update Polygon\_ Subdivision .Count

else

* + - Add record
* Writeprocess end record to the Process\_Status\_Log table

**Process A**

* + Subdivision may contain multiple names in the same field separated by one of the following characters, comma (“,”), semicolon (“;”), backslash (“/”), asterisk (“\*”) or the word “or” (see SystemSettings where keyname=”AREANAME” and subkeyname=”PARSER”)
    - First lookup the full string to find an alternate (tb\_Subdivision\_Mapping).
    - Parse the string or the alternate string
  + Process each name or parsed name separately as follows
    - Lookup the name string to find an alternate (tb\_Subdivision\_Mapping).
    - Strip excess text from the names.

The following rules are used to remove the excess text

1. Perform the steps in the order STRIP\_A, STRIP\_B, STRIP\_C (see SystemSettings where keyname=”AREANAME” and subkeyname LIKE ”STRIP%”).
2. All the STRIP values contain a comma separated list. Parse these. If the last element references another SystemSettings record (ie. STRIP\_B), then the list of STRIP\_A elements must be followed by at least one of the STRIP\_B elements. If the combination of elements is found in the Subdivision name then delete from the beginning of the text to the end of the Subdivision string.
3. Continue processing the STRIP records in order (ie. STRIP\_C, STRIP\_D, etc). Parse the list, locate any one of the elements, remove to the end of the Subdivision string (unless the resulting name string will be less than 5 chars. Always search for the 1st occurrence of the element starting at the end of the Subdivision name. Continue with the next STRIP if one element found.
4. Replace Acronyms with full form. Query SystemSettings where keyname=”AREANAME” and subkeyname LIKE ”ACRONYM%”. The keyvalue is used as follows, the last element in the string is the full form, any elements preceeding the last element are the acronyms to replace. For example, HGTS,HTS,Heights, if either “HGTS” or “HTS” is found in Subdivision name, they are replaced with “Heights”. The acronym string will have 2 or more elements (it’s not restricted to 3).

Note: Do not replace acronyms found within a word. For example the “CK” in “Stock Heights” would not be replaced with “Creek”.

* Locate the Subdivision name in Location\_Subdivision table

SELECT Location\_id FROM Location\_Subdivision

WHERE Location\_id = <Location\_id>

AND Subdivision = <Subdivision name>

If Found

* Increment the count

else

* Search tb\_Subdivision\_City table for a matching Subdivision name

If Found

* + Insert new Location\_Subdivision record with Sub\_id = tb\_Subdivision\_City.Sub\_id

else

* + Insert new Location\_Subdivision record with Sub\_id =0