Linear GIS Data Iowa DOT

April 6

2011

The purpose of this document is to give instructional guide with respect to my current work on Linear GIS Data at Iowa Department of Transportation. Majority of the work done is primarily referenced from Dylan Roy's previous work and this is made possible through the help of Eric Abrams.

By Didum Abraham

Table of Contents

Environment Requirement

- Oracle SQL Client v10.1.1.8
- Oracle SQL*Plus, PL/SQL
- GeoMedia Professional
- Oracle Enterprise with Spatial v11.1 [2]
- Iowa MLLRS v2.6.4.5 [2]

Sample Project

1. Open Toad for Oracle or another Oracle SQL tool
2. Open a new SQL Style and create four tables for paint condition, salt, pavement material, and resilient modulus.
3. Example of table creation with paint condition, which can be seen in i80_tables.sql

```
CREATE TABLE IOWA_TRANSFORM.BPAINT
(
        USER_ID             INT NOT NULL PRIMARY KEY,
        ROUTE_TYPE          VARCHAR2(4),
        ROUTE_NAME          VARCHAR2(255),
        ROUTE_DIR           VARCHAR2(4),
        ST_POST             VARCHAR2(255),
        ED_POST             VARCHAR2(255),
        ST_OFFSET           NUMBER(5,3),
        ED_OFFSET           NUMBER(5,3),
        PAINT_VALUE         NUMBER(5,3),
        DATUM_REFERENCE         CLOB,
       GEOMETRY             MDSYS.SDO_GEOMETRY,
       GEO_EXTENT           VARCHAR2(255),
       SYSTEM_NAME          VARCHAR2(255)
);
```

4. Similar code lines can be used to create tables for pavement, salt and resilient modulus.
5. Tables and column names should be specified by the user's choice of naming convention, however, the column name designated for datum reference must be exactly **DATUM_REFERENCE** of typed **CLOB** and this is to avoid error messages when doing linear overlays.
6. The data type of each column name is to be determined by the user; however, column named GEOMETRY must be of data type **MDSYS.SDO_GEOMETRY**.
7. The following statements show how to put data into the table:
    - Insert into SchemaName.TableName(columnName1,columnName2,columnName3, …)
    - Values ('value1', 'value2', 'value3', …);
    - COMMIT;

8. Data segmentation shows how Figure 1, which can be found in Appendix, is dynamically segmented.
9. The assumed route and directions are I 80 East and I 80 West

Dynamic Segmentation

- If tables have not been created open i80_Tables.sql
    - Before executing file, first comment out all DROP TABLE and delete from statements
    - Now run the script
- **NOTE:** Metadata and Spatial Index is created for each table as in the case below for Paint Condition Table

Metadata and Spatial Index

Example with paint condition table [1]

Use delete statement if metadata already exists in schema otherwise comment it

```
DELETE FROM MDSYS.USER_SDO_GEOM_METADATA where table_name = 'BPAINT';
COMMIT;
```

Create Metadata for resulting Paint-Condition

```
INSERT INTO USER_SDO_GEOM_METADATA values('BPAINT', 'GEOMETRY',
   MDSYS.SDO_DIM_ARRAY( MDSYS.SDO_DIM_ELEMENT('X', -2147483648, 2147483647,
   0.000005), MDSYS.SDO_DIM_ELEMENT('Y',  -2147483648, 2147483647, 0.000005)),
   1050010);
COMMIT;
```

Create Spatial Index for resulting Paint-Condition

```
CREATE INDEX IOWA_TRANSFORM.BPAINT_SPIDX ON
IOWA_TRANSFORM.BPAINT(GEOMETRY) INDEXTYPE IS MDSYS.SPATIAL_INDEX
PARAMETERS('sdo_non_leaf_tbl=TRUE');
```

Cursor Loops

- Open i80_Cursor.sql
- Run this script after i80_Tables.sql has been successfully executed
- Observation
    - Notice that four cursors, namely: c_1, c_2, c_3, c_4, were declared
    - Each cursor could've been declared in a separate SQL-Style environment to ease error trouble shooting, but of course, the followed convention is of desired style.
    - For each individual cursor, there is a FOR-LOOP wrapped around it with a BEGIN-END statement inscribed in the FOR-LOOP
    - In BEGIN-END statement
        - A call is made to RefPostToDatumAL on testlrs and the tag placeholders have been replaced with each column name in the cursor. RefPostToDatumAL takes the reference post linear referencing method data and ties it to the LRS datum [2]
        - Another call is made to XmlObjectToWKT on testlrs where the xml datum object was obtained through RefPostToDatumAL. XmlObjectToWKT returns WKT geometry from the xml datum object.
        - Note that variable, v_datum, is of data type CLOB, hence, when this variable is initialized to get external url, it allows for matched data type, namely, dot get-CLOB function at the end of the statement
        - An if selection structure is used to check for error in RefPostToDatumAL (v_datum)
            - Output v_datum to the screen if error persists, otherwise,
            - Extract WKT from the XmlObject and
            - Update DATUM_REFERENCE and GEOMTRY columns in Paint Condition table
    - This process is repeated in each FOR-LOOP
- This observation remarks what is happening in the script

Linear Overlay

- Open i80_CursorQueries.sql
- Lastly, run this file

Linear Query Investigation

In building these script files, we are after these statements:

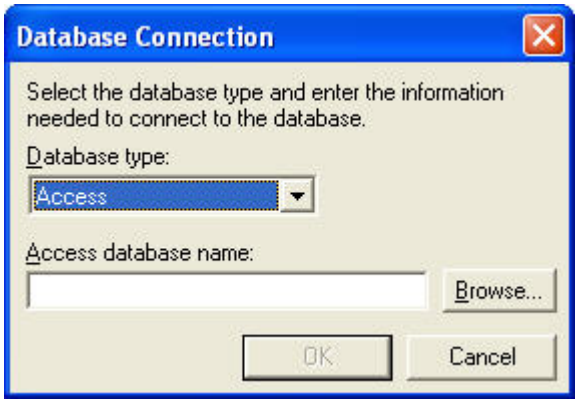
- Find where Paint Condition is greater than or equal 41,
- Where Salt is greater than 5,
- Where Pavement type is PCC and ACC, and
- Where Resilient Modulus is less than 4.1E3 MPa

Linear Query Output

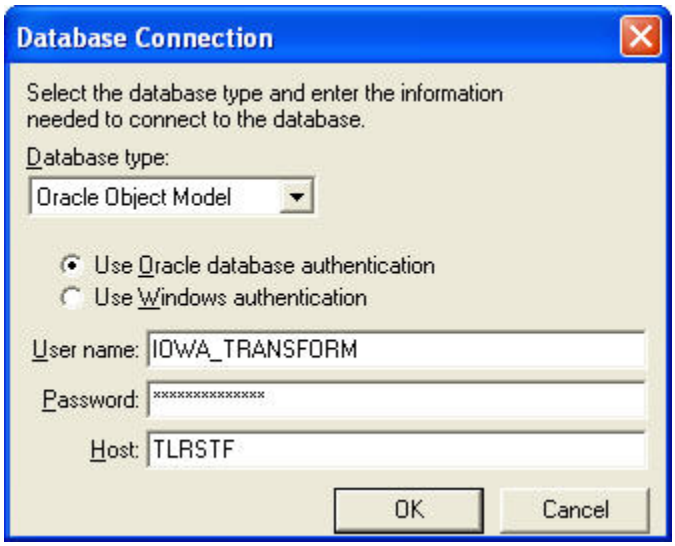
- Observation
    - The first BEGIN-END statement queries Paint Condition and Salt tables
    - Stores findings in a table called BOUTPUT1. This table is automatically created when the script is executed and can always be viewed in the schema.
    - The second BEGIN-END statement queries BOUTPUT1 and Pavement and stores results in BOUTPUT1 table.
    - The third BEGIN-END statement queries BOUTPUT1 and Resilient Modulus and stores results in BOUTPUT1 table.
    - This observation remarks what is happening in the script
- Final Result
    - BOUTPUT1 table now contains final solution to linearly queried data via Oracle
    - BOUTPUT1 table can now be registered in GeoMedia Professional to see the actual geometrical shape.

Register Linear Overlay Results in GeoMedia Professional

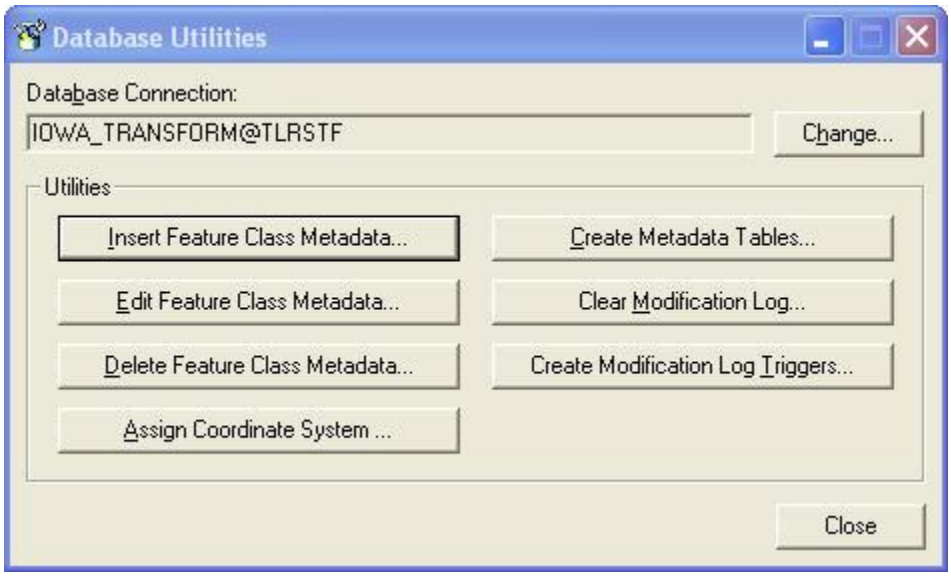
- Go to START→ALL Programs→GeoMedia Professional→Utilities→Database Utilities [1]
- A Database Connection window pops up as can be seen here; change Database Type to **Oracle Object Mode**.



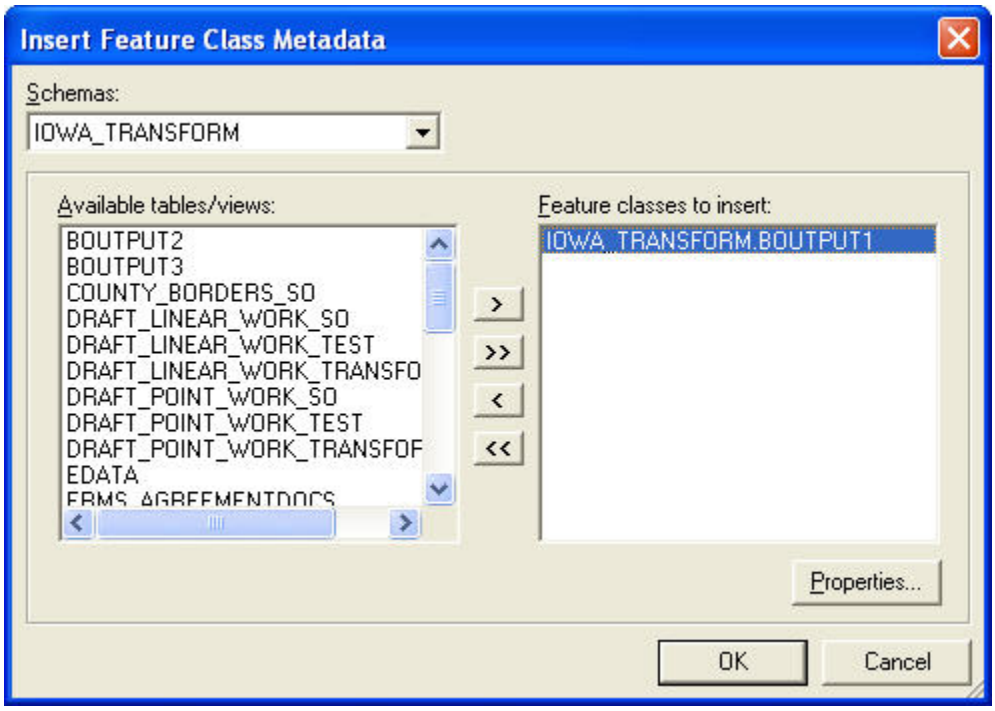
- Select **Use Oracle database authentication**, login, then click **OK**



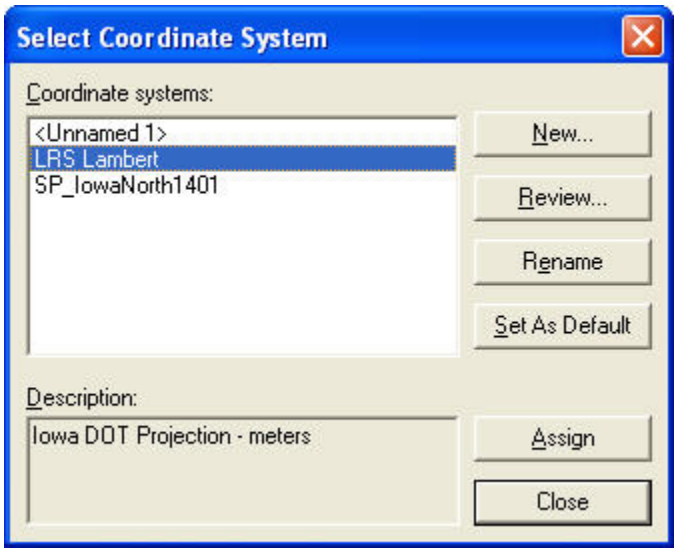
- Select **Insert Feature Class Metadata…**



- In Available tables/views, select **IOWA_TRANSFORM.BOUTPUT1** and click the add button, **>**, then click **OK**

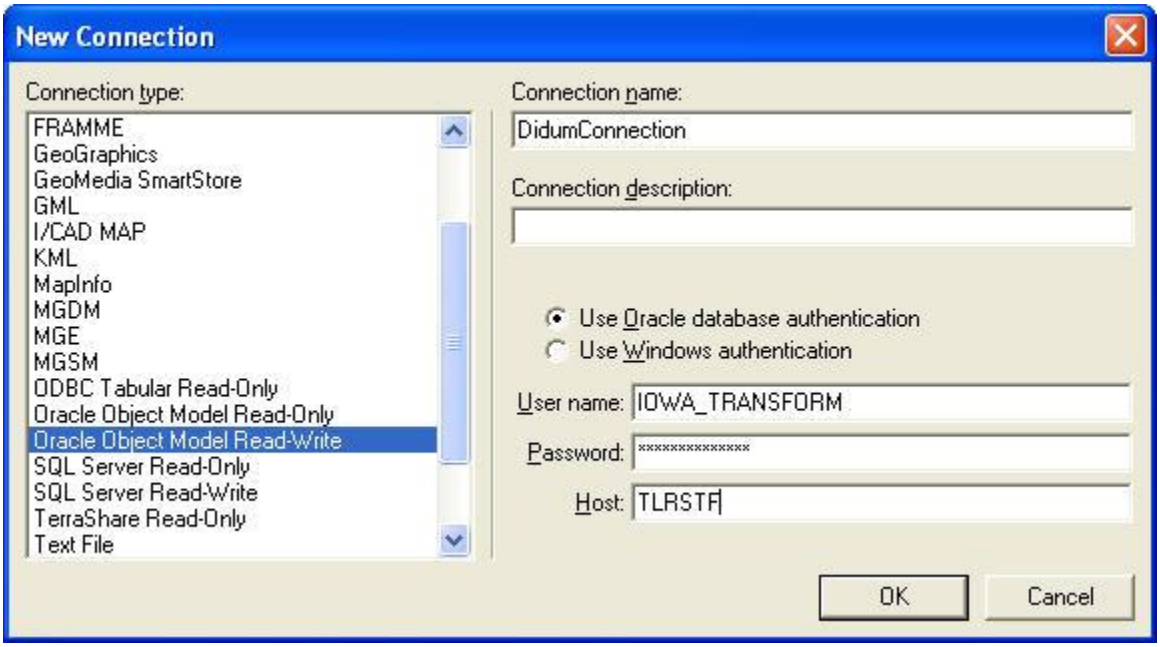


- A new window pops up called Select Coordinate System, choose **LRS Lambert** and click **Assign** and then close out of Database Utilities window



- Now open GeoMedia Professional→Blank GeoWorkspace→OK
    - In GeoMedia Pro, click **Warehouse**, then **New Connection…**
    - Connection type: select **Oracle Object Model Read-Write**
    - Connection name: write a short, easily to remember name in text field

- Select **Use Oracle database authentication**, login, then click **OK**



- Click Legend from the standard menu tab, **Add Legend Entries**, expand connection name in Features, select **IOWA_TRANSFORM.BOUTPUT1**, and click **OK**
- If all goes well, you should now see line segments, which is the result obtained from the queried data as can be seen in Appendix, Figure 2.

Appendix

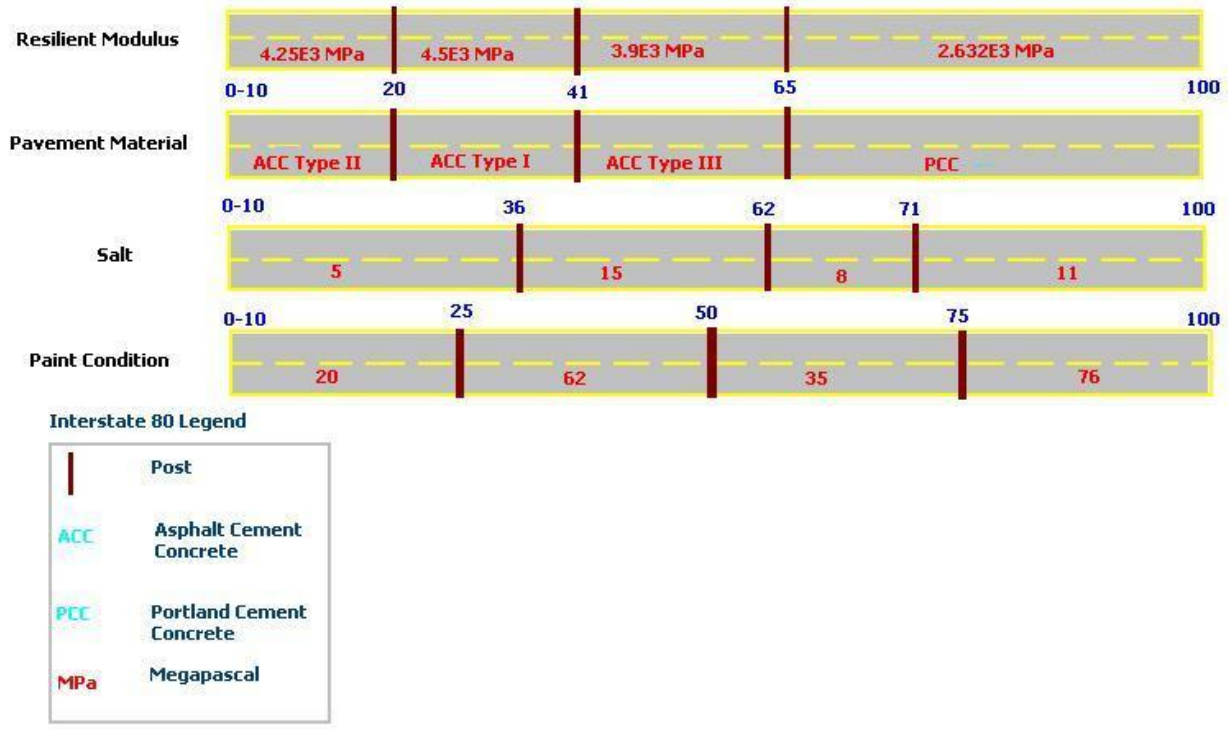


Figure 1: Sample project visualization

Figure 2: Final solution

Works Cited

1. Dylan, Roy. (2010, Jun. 23). In *Linear Overlay*. (chap. Sample Project-Linear Overlay Functions) Retrieved Apr. 12, 2011, from http://portal/IT/GIS/Shared%20Documents/LRS/Linear%20Overlay/Linear%20Overlay%20dynamic%20Segmentation%20Sample%201/Linear%20Overlay%20Documentation.pdf
2. Abrams, Eric. (2010, May. 20). In *How to Store and Transform Location Data*. (chap. Data Storage for LRMs) Retrieved Apr. 13, 2011, from http://portal/IT/GIS/Shared%20Documents/LRS/how%20store%20and%20transform%20data%20with%20LRS.pdf