Homework Assignment

Namjun Kim

Summary

PostgreSQL database was built from a flattened grave data of a couple of generations of a family and the graves are mapped using ArcGIS desktop from PostgreSQL database. A web map was created using GeoServer WMS service and Leaflet JavaScript framework.

Procedure

1. Data model

The data model was designed by normalizing the flat table. Two entity tables (grave and person) and two relationship tables (person-person and person-grave) were created with only grave table has a geometry column. (See Appendix 1, Entity Relationship Diagram)

Note that, the alternate name was treated as an attribute of a person since a person have only one alternate name except for one (Suellent Laurs / Suellen Black, which is to confirmed). But if the alternate name becomes important in the relations (since they are mostly related to marriages), it may have be a separate relationship.

1. Data import to Postgres

Before the data was imported to PostgreSQL database, the data was cleaned since there are some typos and errors in the table. The data was cleaned by visual inspection with the help of Open Refine (Google Refine). E.g. Adam Lodge / Adam Loge, Eric / Erik Lodge (Eric Lodge), John Smith / JohnSmith, Suellen Lodge/ Suellent Laurs / Suellen Black (extra commas which make a row incompatible). Some are obvious but some require confirming with client if it were a real project.

To import the data to Postgres, a sql script to create tables (see Appendix II), a script to insert all the data, and a query to recreate the original flat table to confirm that the import were successful. Postgres 9.5 and PostGIS 2.3 with pgAdmin III were used.

Since the size of the whole data is not too large, the visual inspection and hard code the import script were viable. It may take more time to write some codes to do semi-automatic import. However, if the data size were very large writing some code to inspect the data and import the data to database may be necessary. I did not try to write one for this project.

1. Desktop GIS

ArcGIS Pro was used to map the data in a desktop environment.

First, sql scripts were written to create two relationship views, which have actual names (database tables have id number from person table primary key), for the desktop application. (Appendix III)

A database connection to Postgres were created inside ArcGIS Pro, the grave table with spatial column were imported to ArcGIS Pro, and two relationship views (person-person, grave-person) were also imported. The relates were established in ArcGIS Pro between the grave table and grave-person view, and grave-person and person-person. Then if you select a grave, you can find who were interred and who are related through relate tables.

Although importing two views ArcGIS Pro was successful, there was warning message because the views do not have integer-type OID. The permanent OID can be given by creating two tables instead of two views. But then two additional tables have to be maintained which are duplicate. Therefore, the views were used in desktop ArcGIS as well as in web maps later.

1. Web map using GeoServer/Leaflet

The GeoServer was installed and layers from Postgres were connected and published. Leaflet JavaScript were written to request WMS (web map service) from GeoServer. SLD (Style Layer Descriptor) was used to symbolize the point using a custom icon. (a screen capture in Appendix IV)

However, the web map is only available at localhost at the moment. So, a web map was created using exported geoJSON type data sets inside the html document. It also symbolized using a custom icon and also popup label were added and links to related people. If you click a grave, a popup will give the names of people interred as well as links to the people related to that person. (It is available at <http://njkim.github.io/grave_map.html>)

Further direction

WMS from GeoServer is useful to provide simple web map quickly but rather limited in terms of accessing the attributes. However, Leaflet does not seem to support WFS (web feature service) natively. So, the second web map described above cannot be created using GeoServer WMS service.

So, some alternative path was tried. One obvious possibility was using GeoServer’s JSON format. That was part of the reason why the second web map was written based on geoJSON file type. Unfortunately, requesting a JSON from GeoServer was not successful because the local server was not configure to satisfy so-called “same origin policy” for AJAX request. A possible work-around by using padded JSON (JSONP) was also not successful. The request got an error even though the JSONP was enabled in GeoServer.

More research on JSONP or ajax may be useful, or there is other possible solution including using OpenLayer instead of LeafLet. OpenLayer is known to have much richer support on WFS.

All the related documents and scripts are available in Gihub repository including this report.

<https://github.com/njkim/grave-project>

Appendix I

Appendix II

Appendix III

Appendix IV