Farallon Geographics Homework Assignment

12/24/2018

Namjun Kim

Summary

PostgreSQL database was built from a flat data table of the graves of a couple of generations of a family. They are mapped using ArcGIS desktop from PostgreSQL database. A web map was successfully 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 haing a geometry column. (Appendix I for 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), a separate relationship may be useful.

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 were 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 the client if it were a real project.

To import the data to Postgres, a sql script was written to create tables, a script to insert 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. (Appendix II)

Since the size of the whole data is not too large, the visual inspection and hard codings the import script were viable. It may take more time to write codes to do semi-automatic inspection and 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 with actual names (the database tables have id numbers from the person table primary keys), for the further applications. (Appendix III)

A database connection to Postgres was created inside ArcGIS Pro. The grave table with spatial column was imported to ArcGIS Pro, and two relationship views (person-person, grave-person) were also imported. The relates were added in ArcGIS Pro between the grave table and the grave-person view, and the grave-person and the person-person views. 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 the layers from Postgres were connected and published. A webpage was created and Leaflet JavaScript code was 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 was also symbolized using a custom icon and also popup labels and links to related people were added. 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>. Also see a screen capture Appendix V)

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 paths were 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 of various reasons including the fact that the local server was not configure to satisfy so-called “same origin policy” for AJAX requests. 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.

Therefore, more researches on JSONP or ajax requests to GeoServer may be useful, or there may be 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 including this report are available in Gihub repository.

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

Appendix I

GRAVE

lon

lat

PERSON

first\_name

Alternate\_last\_name

RELATIONSHIP

1

grave\_id

relationship

INTERRED

1

N

N

person\_id

Alternate\_first\_name

last\_name

Appendix II

1. Create tables

CREATE EXTENSION postgis;

CREATE TABLE PERSON (

person\_id INTEGER NOT NULL,

first\_name VARCHAR(15) NOT NULL,

last\_name VARCHAR(15) NOT NULL,

alternate\_name VARCHAR(30),

PRIMARY KEY (person\_id)

);

CREATE TABLE GRAVE (

grave\_id INTEGER NOT NULL,

lat REAL NOT NULL,

lon REAL NOT NULL,

geom geometry(POINT,4326),

PRIMARY KEY (grave\_id),

CHECK (lat >= -90.0 AND lat <= 90.0),

CHECK (lon >= -180.0 AND lon <= 180.0)

);

CREATE TABLE INTERRED (

grave\_id INTEGER NOT NULL,

person\_id INTEGER NOT NULL,

PRIMARY KEY (grave\_id, person\_id),

FOREIGN KEY (grave\_id) REFERENCES grave(grave\_id),

FOREIGN KEY (person\_id) REFERENCES PERSON(person\_id)

);

CREATE TABLE RELATIONSHIP (

person1\_id INTEGER NOT NULL,

person2\_id INTEGER NOT NULL,

relationship VARCHAR(10),

PRIMARY KEY (person1\_id, person2\_id),

FOREIGN KEY (person1\_id) REFERENCES PERSON(person\_id),

FOREIGN KEY (person2\_id) REFERENCES PERSON(person\_id)

);

1. Import to PostgreSQL

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (1, 'Adam', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (2, 'Addison', 'Corrow');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (3, 'Almetria', 'Vaba');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (4, 'Betsy', 'Bremer', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (5, 'Bridgett', 'Corrow');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (6, 'Bud', 'Spence');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (7, 'Carl', 'Corrow');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (8, 'Carl', 'Corrow II');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (9, 'Cesar', 'Arteaga');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (10, 'David', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (11, 'Dorothy', 'Lodge', 'Thornton');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (12, 'Eric', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (13, 'Evelyn', 'Spence', 'Dalgity');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (14, 'Ginny', 'Spence');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (15, 'Hilmer', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (16, 'Jodi', 'Corrow', 'Pukle');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (17, 'John', 'Corrow');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (18, 'John', 'Smith');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (19, 'Julann', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (20, 'Kai', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (21, 'Odin', 'Arteaga');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (22, 'Pam', 'Corrow', 'Spence');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (23, 'Raymond', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (24, 'Rosemary', 'Otto', 'Lodge');

INSERT INTO PERSON (person\_id,first\_name,last\_name) VALUES (25, 'Sebastian', 'Arteaga');

INSERT INTO PERSON (person\_id,first\_name,last\_name,alternate\_name) VALUES (26, 'Suellen', 'Lodge', 'Laurs, Black');

INSERT INTO GRAVE VALUES (1, 42.2666, -71.8, ST\_GeomFromText('POINT(-71.8 42.2666)', 4326));

INSERT INTO GRAVE VALUES (2, 42.8666, -73.816, ST\_GeomFromText('POINT(-73.816 42.8666)', 4326));

INSERT INTO GRAVE VALUES (3, 33.3033, -116.981, ST\_GeomFromText('POINT(-116.981 33.3033)', 4326));

INSERT INTO GRAVE VALUES (4, 33.3034, -116.982, ST\_GeomFromText('POINT(-116.982 33.3034)', 4326));

INSERT INTO GRAVE VALUES (5, 32.9952, -117.2603, ST\_GeomFromText('POINT(-117.2603 32.9952)', 4326));

INSERT INTO GRAVE VALUES (6, 37.6688, -122.0808, ST\_GeomFromText('POINT(-122.0808 37.6688)', 4326));

INSERT INTO GRAVE VALUES (7, 20.914, -100.7436, ST\_GeomFromText('POINT(-100.7436 20.914)', 4326));

INSERT INTO GRAVE VALUES (8, 37.6688, -122.0808, ST\_GeomFromText('POINT(-122.0808 37.6688)', 4326));

INSERT INTO GRAVE VALUES (9, 21.3, -157.81666, ST\_GeomFromText('POINT(-157.81666 21.3)', 4326));

INSERT INTO GRAVE VALUES (10, 32.99, -117.2, ST\_GeomFromText('POINT(-117.2 32.99)', 4326));

INSERT INTO GRAVE VALUES (11, 47.659, -117.425, ST\_GeomFromText('POINT(-117.425 47.659)', 4326));

INSERT INTO GRAVE VALUES (12, 33.122, -117.297, ST\_GeomFromText('POINT(-117.297 33.122)', 4326));

INSERT INTO GRAVE VALUES (13, 37.6688, -122.0808, ST\_GeomFromText('POINT(-122.0808 37.6688)', 4326));

INSERT INTO GRAVE VALUES (14, 41.8, -71.422, ST\_GeomFromText('POINT(-71.422 41.8)', 4326));

INSERT INTO GRAVE VALUES (15, 37.6688, -122.0808, ST\_GeomFromText('POINT(-122.0808 37.6688)', 4326));

INSERT INTO GRAVE VALUES (16, 41.8, -71.422, ST\_GeomFromText('POINT(-71.422 41.8)', 4326));

INSERT INTO GRAVE VALUES (17, 27.988, 86.925, ST\_GeomFromText('POINT(86.925 27.988)', 4326));

INSERT INTO GRAVE VALUES (20, 37.6688, -122.0808, ST\_GeomFromText('POINT(-122.0808 37.6688)', 4326));

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (1, 13);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (1, 6);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (2, 22);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (3, 11);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (4, 15);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (5, 12);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (5, 26);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (6, 1);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (7, 9);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (8, 14);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (9, 4);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (10, 23);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (11, 24);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (12, 19);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (12, 25);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (12, 21);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (12, 18);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (13, 10);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (14, 17);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (15, 7);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (16, 8);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (16, 16);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (16, 5);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (16, 2);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (17, 20);

INSERT INTO INTERRED (grave\_id,person\_id) VALUES (20, 3);

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (1,3,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (1,20,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (3,20,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (6,14,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (6,22,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (7,8,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (7,17,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (8,2,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (8,5,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (8,16,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (11,24,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (11,15,NULL);

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (11,12,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (11,4,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (11,23,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (12,14,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (12,26,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (12,19,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (12,10,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (12,1,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (13,14,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (13,22,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (13,6,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (14,1,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (14,10,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (14,19,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (15,24,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (15,23,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (15,12,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (15,4,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (16,2,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (16,5,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (19,18,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (19,21,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (19,25,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (22,7,'Spouse');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (22,17,'Parent');

INSERT INTO RELATIONSHIP (person1\_id, person2\_id, relationship) VALUES (22,8,'Parent');

UPDATE relationship SET relationship = 'Spouse' WHERE person1\_id=11 AND person2\_id=15

1. Recreate the original flat table

CREATE TABLE grave\_flat AS

(SELECT a.grave\_id "GraveID",

a.lat "Lat",

a.lon "Lon",

concat(a.first\_name, ' ', a.last\_name) "Interred Name",

a.alternate\_name "Alternate Name",

concat(b.first\_name, ' ', b.last\_name) "Related Name",

b.relationship "Relationship"

FROM (SELECT g.grave\_id, g.lat, g.lon, p.person\_id, p.first\_name, p.last\_name, p.alternate\_name

FROM grave g, person p, interred i

WHERE i.person\_id = p.person\_id

AND g.grave\_id = i.grave\_id) a

LEFT JOIN

(SELECT q.last\_name, q.first\_name, r.person1\_id, r.relationship

FROM relationship r, person q

WHERE r.person2\_id = q.person\_id) b

ON a.person\_id = b.person1\_id);

ALTER TABLE grave\_flat ADD COLUMN id SERIAL;

Appendix III

CREATE VIEW interred\_view AS

(SELECT i.grave\_id, concat(p.first\_name, ' ', p.last\_name) as person

FROM interred i, person p, grave g

WHERE g.grave\_id = i.grave\_id

AND p.person\_id = i.person\_id);

CREATE VIEW relationship\_view AS

(SELECT person1, concat(q.first\_name, ' ', q.last\_name) as person2, a.relationship

FROM (SELECT concat(p.first\_name, ' ', p.last\_name) person1, r.person2\_id, r.relationship

FROM relationship r, person p

WHERE p.person\_id = r.person1\_id) a,

person q

WHERE q.person\_id = a.person2\_id);

CREATE VIEW relationship\_2way\_view AS

((SELECT \*

FROM relationship\_view)

UNION ALL

(SELECT person2, person1, relationship

FROM relationship\_view

WHERE relationship = 'Spouse' OR relationship IS NULL

)

UNION ALL

(SELECT person2, person1, 'Child'

FROM relationship\_view

WHERE relationship = 'Parent'

));

Appendix IV



Appendix V

