Applied US Census Spatial Reference PostGIS Database

Nicholas Owen Tapia

Spatial Sciences Institute, December 13th, 2013

tapia.nicholas@gmail.com

ABSTRACT

Mountains of wonderful data are now publicly available and downloadable from geoportals in both vector and raster format. There are two main options for downloading them: bulk download [2] or select by area [1]. For many use cases this can be terribly restrictive and inefficient: it forces users to download more than they need and does not allow them to run custom queries that can spatially relate geometries. This paper presents an efficient way to offer highly customizable vector data downloads. Taking advantage of the highly advanced and open source Postgresql and its spatial component PostGIS, the proposed solution stores vector data in a PostGIS database, is queried in sql, and easily written out as a shapefile. Because the data is stored in a spatial database joining is not just limited table fields but enabled for spatial joins as well. The result is a significant reduction in data download and preparation time.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *Spatial Databases and GIS*

Keywords

PostGIS, Postgresql, postgres, Census, Shapefile, Custom Shapefiles, Custom Download

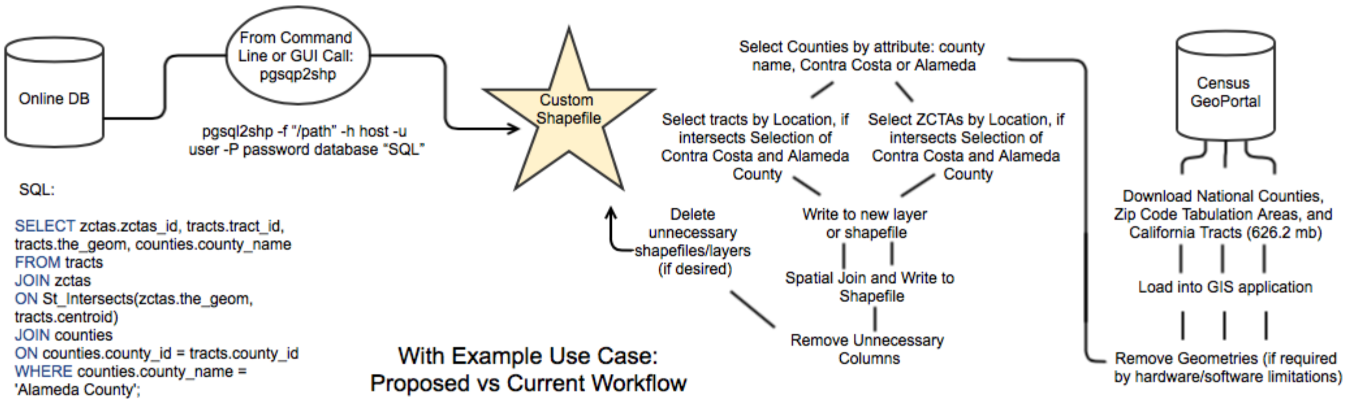
1. INTRODUCTION

1.1 The Problem

The problem addressed here is the inefficient work that data users must do to relate (spatially or otherwise) data. One example where much this work can be saved is with the US Census Tiger [2] Lines geoportal where the geometries such as zip code tabulation areas, census tracts, and county boundaries are offered. The US Census offers Tiger Line geometries as whole, isolated shapefiles. To exemplify how this might be inefficient, take his use case: A user needs a file with census tract geometries from Alameda and Contra Costa County with an additional column identifying within which zip code those tracts fall, as well as a column identifying their respective county name. First, a user must download the national zip code tabulation file, the national counties file, and the tracts for California, the zipped file sizes being 525.5 mb, 73.5 mb, and 27.2 mb respectively. Some GIS users will have to par down the size of the data by deleting geometries and columns first due to application and hardware restrictions. There are multiple paths to reach the final destination, but they will all have a lot of mouse clicks that probably involve running multiple select queries, a couple spatial joins, and many column deletions to end up with the appropriate shapefile.

1.2 The Proposed Solution

The proposed solution is an online queriable database that allows for users to input a sql statement through code or graphical user interface (GUI) and download the resulting file (see Figure 1). The main benefits to the user are saving time and ease of use.


Online DB
From Command Line or GUI Call: pgsqp2shp
Custom Shapefile
Select Counties by attribute: county name, Contra Costa or Alameda
Select tracts by Location, if intersects Selection of Contra Costa and Alameda County
Select ZCTAs by Location, if intersects Selection of Contra Costa and Alameda County
Write to new layer or shapefile
Spatial Join and Write to Shapefile
Remove Unnecessary Columns
Delete unnecessary shapefiles/layers (if desired)
Census GeoPortal
Download National Counties, Zip Code Tabulation Areas, and California Tracts (626.2 mb)
Load into GIS application
Remove Geometries (if required by hardware/software limitations)
With Example Use Case: Proposed vs Current Workflow

```
pgsql2shp -f "/path" -h host -u
user -P password database "SQL"
```

SQL:

```
SELECT zctas.zctas_id, tracts.tract_id,
tracts.the_geom, counties.county_name
FROM tracts
JOIN zctas
ON St_Intersects(zctas.the_geom,
tracts.centroid)
JOIN counties
ON counties.county_id = tracts.county_id
WHERE counties.county_name =
'Alameda County';
```

Figure 1: Proposed vs Current Workflow

1.3 Project Scope

The sections following encompass details on 1) a proof of concept database for US Census Tiger Lines and 2) integration of parcels and city boundaries into said database for potential future uses.

2 DATABASE

Postgresql 9.3 v2.1.0-2 with the PostGIS 2.1 [3] spatial extension was chosen because of its open source and free nature, its performance track record, and its ability to read and write out various data formats. The proof of concept version of the database is hosted on a local machine running an OS X 10.9 operating system with a 2.2 GHz Intel Core 2 Due processor and 6 GB of memory.

2.1 Data Sources

Tiger Line boundary files were downloaded from The US Census Bureau. The files downloaded were the national shapefile coverages from the 2013 vintage of State, ZCTA, and County areas and each state coverage for block and tract areas. [2]

The national division geometries used in the database were created from the the state geometries by dissolving by the division field. The country geometry was created from the state geometries by dissolving them completely.

Nine of the nine Bay Area Counties offer parcel geometries and five of them offer city boundary geometries. Both were downloaded as shapefiles from their respective county geoportals. [4]

2.2 Database Fields and Relationships

The EAR diagram describes the extent of and relationships between the data (see Figure 2). All the fields are character varying set to 255 for name fields and the maximum needed length for id fields. The exceptions being parcel_id and city_id which are integers.

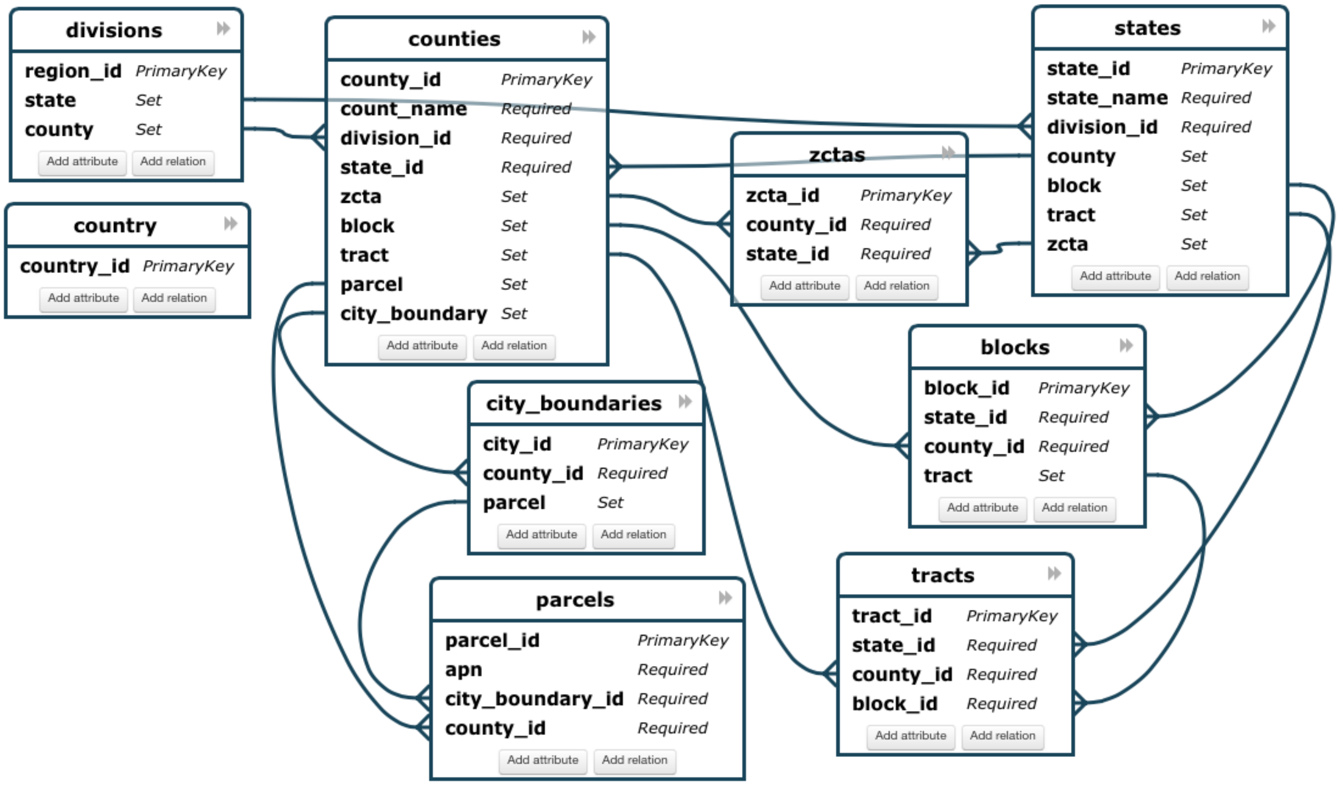


Figure 2: EAR Diagram

2.3 Data Processing Methodology

The goal of the data processing was to end up with data that is easily queriable for customized selections of spatially related geometries. To this end, the most important step was to unify across all geometry levels the column names and give unique names to each geometry primary key id field. For example changing the primary key for tracts from the label `geoid` to `tract_id`. The sequential steps of the workflow are outlined in Figure 3.

Bringing the data into the PostGIS database was done with the `shp2pgsql | psql` command. The explicit command used was:

```
shp2pgsql -W "latin1" -s 4269 /path | psql -h host -d database -U user
```

The parcels and city_boundaries geometries were converted to from SRID 2226 or 2227 (depending on the county) to SRID 4269 in the `shp2pgsql` command. This SRID was chosen because it is the SRID used by the Tiger Line geometries.

The tracts and blocks were mass upzipped and loaded by using an `ipython notebook` and the `os, os.path` libary. They were then all joined together with the `UNION ALL` command.

The geometry types of the parcels and city boundary files varied from geometry to multipolygon to three dimensional. The PostGIS commands ST_Multi and 2D_Force were used to create a unified multipolygon geometry type.

Nonessential fields were stripped from the tables along with a primary key for joining and placed in the "storage" schema for future use. For the purposes of this database, the idea was to provide reference geometries with their unique identifiers so that attributes can be mapped to them. To this end, only the essential fields were kept in the same table. For a listing of the original complete files please see the 2013 TIGER Geodatabase Records Layouts document. [5]

Primary and foreign keys were specified for data consistency and better queries. Parcels and city boundaries had to have their own primary key generated as APN numbers not unique fields for some counties.

Centroids were added to save centroid calculation time for certain queries when spatially relating geometries that are not already related. Additionally the spatial viewer program GeoCanvas allows for connections to postgres databases and loads geometries significantly faster when centroids have been precalculated.

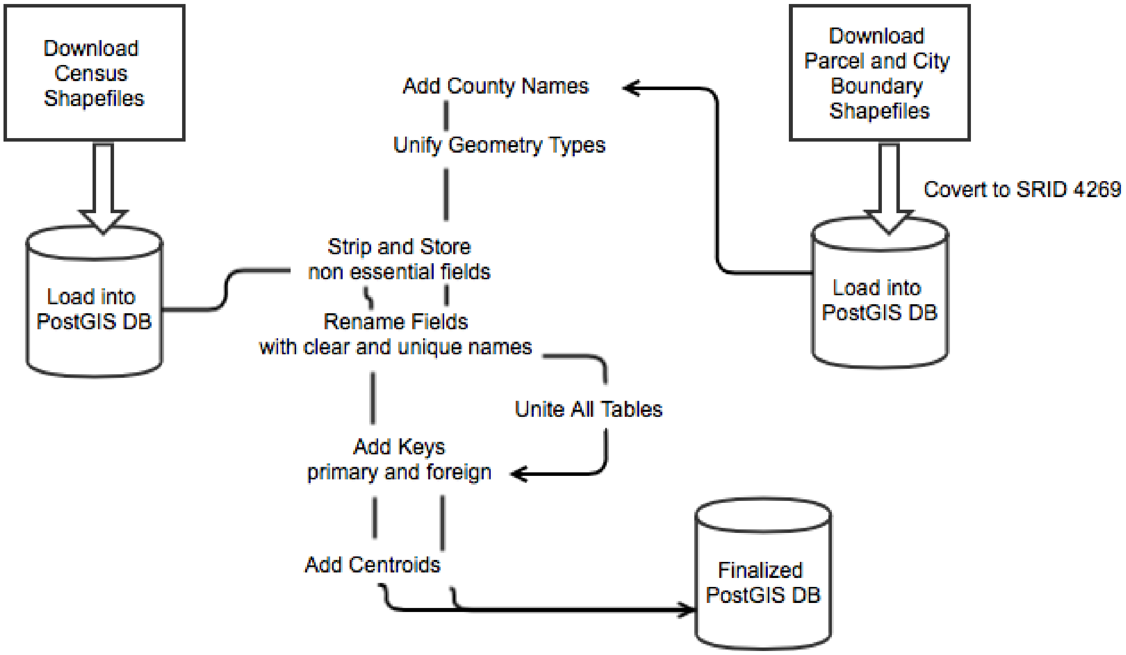

Download Census Shapefiles
Load into PostGIS DB
Add County Names
Unify Geometry Types
Strip and Store non essential fields
Rename Fields with clear and unique names
Unite All Tables
Add Keys primary and foreign
Add Centroids
Download Parcel and City Boundary Shapefiles
Covert to SRID 4269
Load into PostGIS DB
Finalized PostGIS DB

Figure 3: Data Processing Workflow

3 Discussions

This project was born of personal experience. In 2013 I did my first GIS project *ever* and it required a shapefile along the lines of the use case described in the introduction. It was very time consuming. Since then, I have downloaded the same files over and over again (I do not always use the same workstation) from the census website and created custom shapefiles for various projects. As my GIS skills have improved, I have gotten faster at it, but it is still an aggravating and a sometimes error prone process that *still* is very time consuming. This database has *significantly* improved my productivity. My hope is that I can extend this boon to other GIS users.

It is a similar hope I imagine that is propelling papers studying optimization of spatial queries [6] and how to build efficient spatially enabled medical imagery databases [7], both in PostGIS.

The trickiest part of process was setting up postgresql enabling PostGIS. All 3 of the key systems, Postgresql, PostGIS, and OS X were within a month of their latest release. Things did not go smoothly in the beginning. I could not get the homebrew installation process to work with OS X 10.9 and I opted for the graphical installer instead. One issue was that a dynamic library was missing from the Postgresql installer and another issue was that PostGIS would not automatically enable itself after install. Both issues were solved in the forums of stackoverflow by the users Ken Thomases and underdark. Thank you.

The loading of tracks and blocks would have have been significantly more time consuming if it were not for the help of Conor Henley. There are over 50 shapefiles for each of these geography levels and the automatic loading techniques that he shared were a great asset. Thank you.

One current road block is that the centroids for parcels are still unable to be calculated. I am guessing it is because the geometries are still not formatted properly across all the different counties. This is a serious problem that needs to be troubleshooted as it prevents many useful spatial joins.

4 Future Work

The next step is to make this database publicly accessible online. The necessary work is on the back end. A server must be aquired, roles and permissions must be set up, analytics must be gathered on the types of queries being performed, and the database must be optimized for the type and number of queries being performed as well as for downloads. Using Postgresql tools users of the command line can already write out the data into the desired format. But to be truly accessible and user friendly, some front end work that would need to be done includes a web page, documentation, advertising, and a GUI.

Potentially, this could be the system used by the US Census to serve its reference geometries and extend as well to census table data.

Ideally, this would grow into a complete national (international?!) political spatial reference database. All administrative boundaries from the local, state, and federal levels would reside in the same place where they could be selected and related by data users. With the right user and permissions settings, the data could be loaded and maintained by the appropriate data providers. It would an excellent way to pool resources and improve data accessibility across all levels of government.

5 REFERENCES

[1] USGS National Map Viewer and Download Platform.

http://nationalmap.gov/viewer.html.

[2] TIGER ® TIGER/Line ® and TIGER ®-Related Products.

http://www.census.gov/geo/www/tiger

[3] Postgresql and PostGIS.

http://www.postgresql.org. http://postgis.net.

[4] Parcel Download County Sources: Alameda, Contra Costa, Marin, Napa, San Mateo, San Francisco, Santa Clara, Sonoma, and Solano.

http://www.acgov.org/government/geospatial.htm,
http://www.ccmap.us/catalog.asp?UserChoice=1&Layercntrl=0000000000000000000000#1,
http://www.marinmap.org/DNN/Data/GISDataDownLoad.aspx,
http://gis.napa.ca.gov/giscatalog/viewXML.asp?name=MAINGIS.GIS.PARCELS_PUBLIC&meta_style=fgdc#Identification_Information,
http://www.co.sanmateo.ca.us/portal/site/SMC/menuitem.e4c29a0f12f966047b830e43917332a0/?vgnextoid=9122f9b53ff3b210VgnVCM1000001937230aRCRD&vgnextfmt=DivisionsLanding,
https://data.sfgov.org/Geography/City-Lots-Zipped-Shapefile-Format-/3vyz-qy9p,
http://www.sccgov.org/sites/gis/GISData/Pages/Available-GIS-Data.aspx,
https://gis.sonoma-county.org/catalog.asp,
http://regis.solanocounty.com/data.html.

[5] 2013 TIGER Geodatabase Records Layouts.

http://www.census.gov/geo/maps-data/data/pdfs/tiger/tgrshp2013/TIGER_GDB_Record_Layouts.pdf.

[6] Simion, Bogdan. Ray, Suprio. Brown, Angela. Surveying the landscape: an in-depth analysis of spatial database workloads. SIGSPATIAL, ISBN 1450316913, pages 376 - 385, 2012

[7] Aji, Ablimit. Wang, Fusheng. Saltz, Joel. Towards building a high performance spatial query system for large scale medical imaging data. SIGSPATIAL, ISBN 1450316913, pages 309 - 318, 2012