

Postgre/PostGIS Tutorial

Gathering and Adding Data



University
of Colorado
Denver

Created by: Ricardo Oliveira
ricardo.oliveira@ucdenver.edu


July 2014

On this second tutorial we will be covering the basic steps for gather real world data and load it into a database.

Adding Data.

For this tutorial we will be using free data from the City and County of Denver.

Census Neighborhood Demographics (2010)

Last updated 4/24/2013 

Based upon the Neighborhoods layer from Community Planning and Development. This layer is a duplicate of that layer, but with additional demographic information compiled from the 2010 US Census. Although every effort was made to ensure that blocks were assigned to their proper neighborhoods, due to geographic problems inherent in the original 2010 census block data, errors may exist. This data-set includes the housing and population data from redistricting file P.L. 94-171 of the 2010 census. The data-set does not contain data for any enclaves administered by other jurisdictions that are located within the City and County of Denver's boundary.

Resources

Description	Format	Action
Census Neighborhood Demographics (2010) - Shapefile	shp	<input type="button" value="Download"/>
Census Neighborhood Demographics (2010) - AutoCAD DWG	dwg	<input type="button" value="Download"/>
Census Neighborhood Demographics (2010) - Google KML	kml	<input type="button" value="Download/Open with Google Earth"/> <input type="button" value="Open with Google Maps"/> <input type="button" value="Open with Bing Maps"/>
Census Neighborhood Demographics (2010) - Comma-Separated Values	csv	<input type="button" value="Download"/>
Census Neighborhood Demographics (2010) - Metadata	xml	<input type="button" value="Download"/>
Census Neighborhood Demographics (2010) - Esri File Geodatabase	gdb	<input type="button" value="Download"/>

Details

Maintainer	City and County of Denver, Technology Services / GIS Data
Maintainer Email	denvergis@denvergov.org
Version	1.0.2

Tags

[census](#) [demographics](#) [gis](#) [society](#)

Open Data License

You are free to copy, distribute, transmit and adapt the data in this catalog under an open license.

For details, please review the [terms of use](#).

About This Catalog

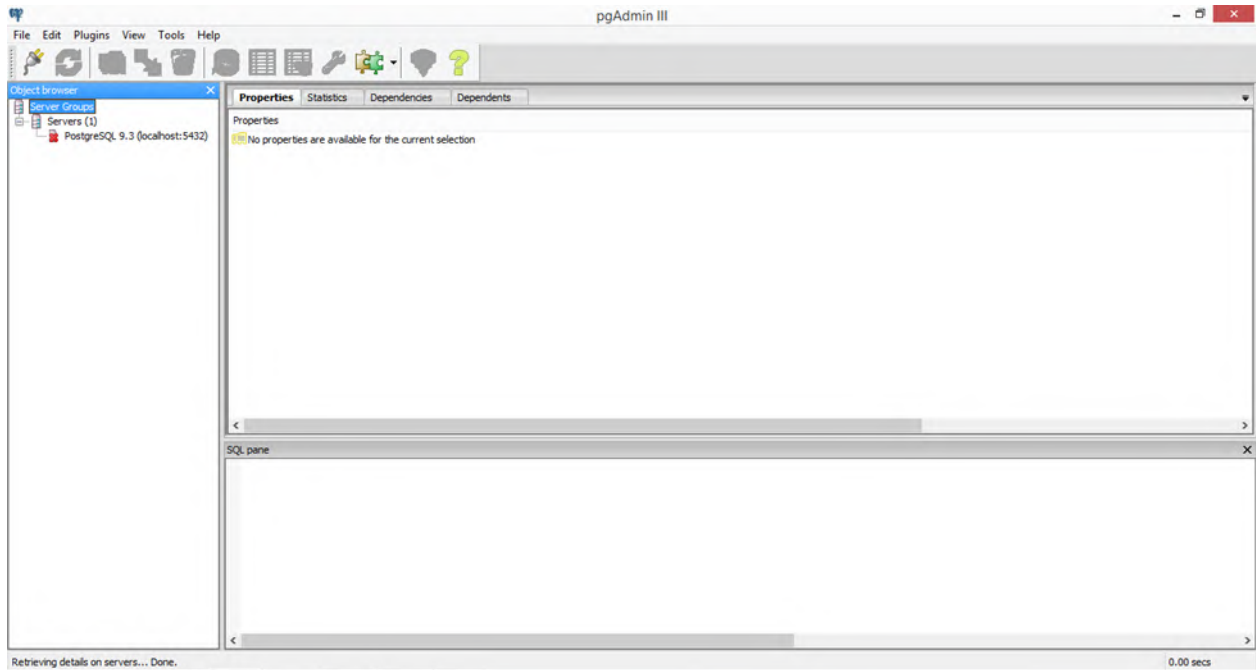
This catalog was created in partnership with [Open Colorado](#).

To search for open data provided by other government organizations in Colorado please visit the [Open Colorado Data Catalog](#).

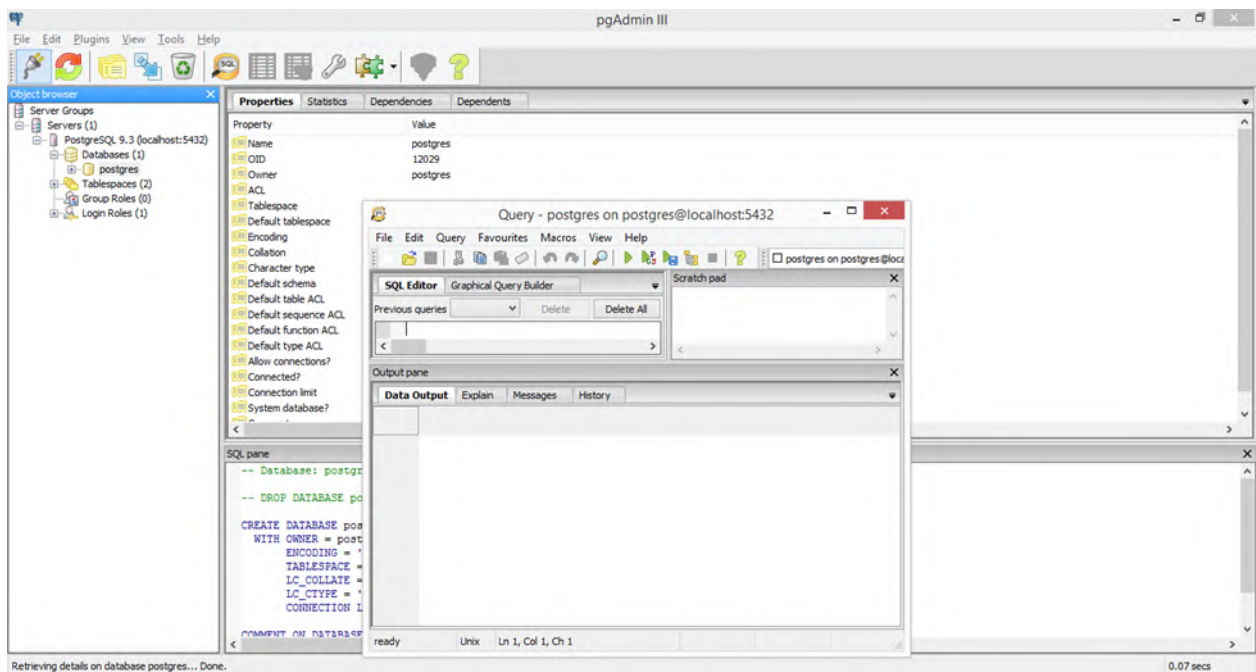
At the <http://data.denvergov.org/> we will download a simple shapefile for the neighborhoods of Denver.

Search for Neighborhood at the search box and download the shp file for Census Neighborhood Demographics (2010). Extract the data and save in a place with easy access.

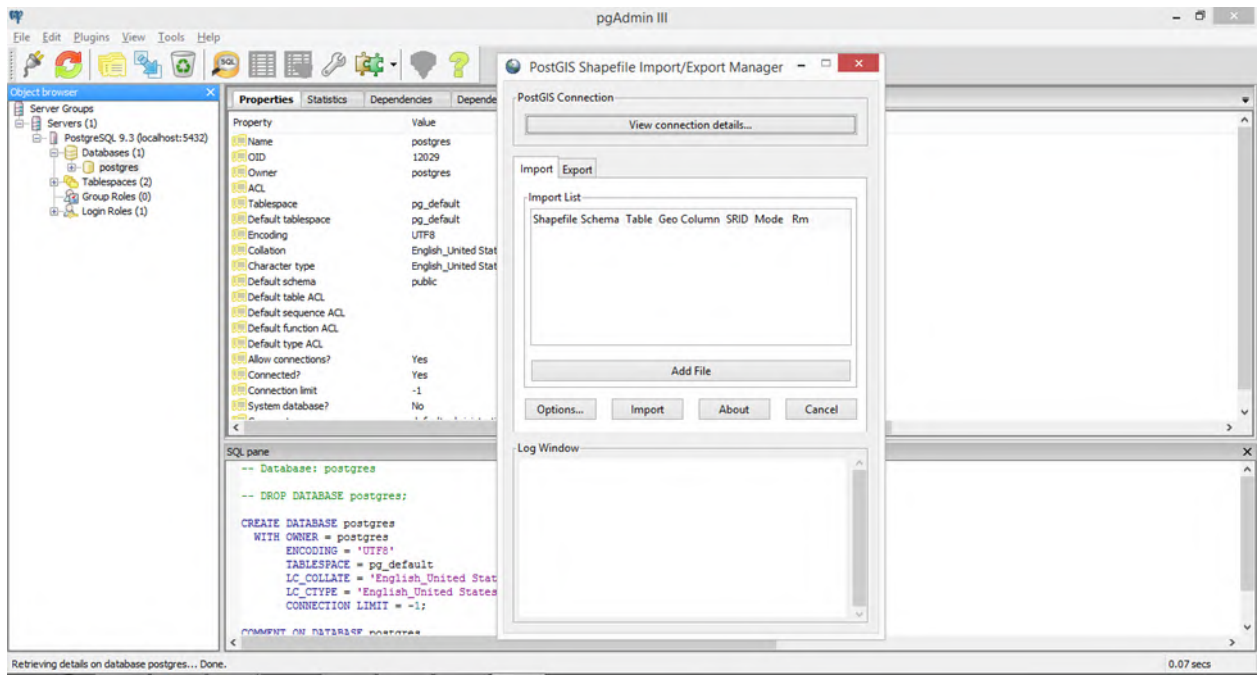
Now that we have our data let's take a look at the PostgreSQL. Open the pgAdmin III program, this is our command center to work with all databases.



This is how the screen looks like for the first time we open pgAdmin. pgAdmin is the Graphical User Interface (GUI) that allow us to manipulate postgresSQL without the need to input our commands into a written only terminal. On the left side there is a navigation box, since we are not working with multiple database servers we have only one connection here, double click it. Explore a little the directories, don't worry for now what each category means. The two main tools that we have to be familiarized here are the SQL builder and the PostGIS importer.

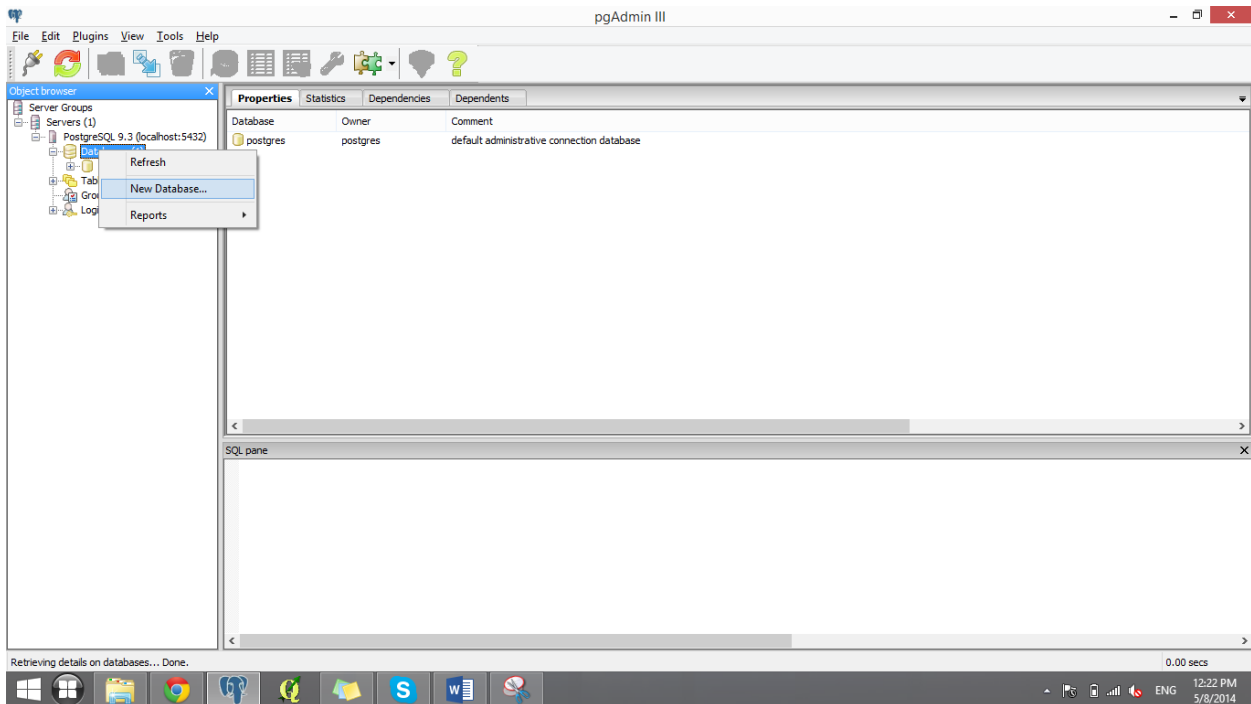


Query builder.

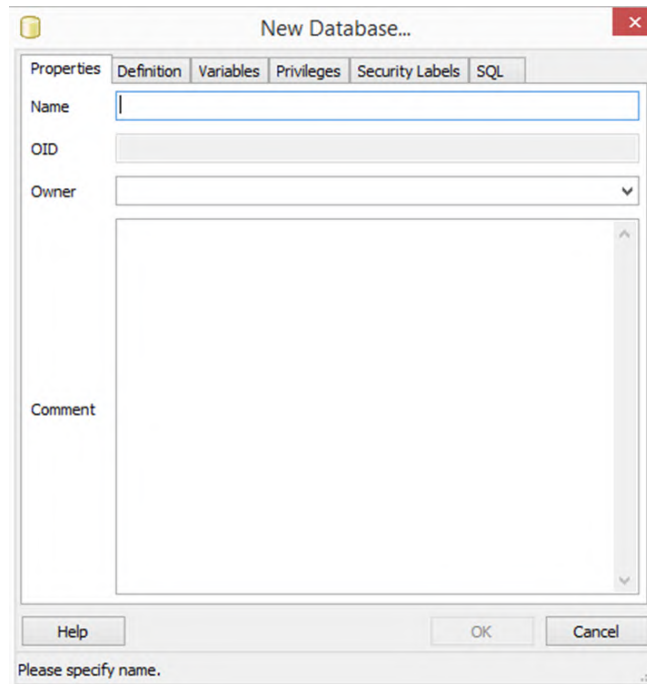


PostGIS importer.

Now let's create a new database for our Denver data.

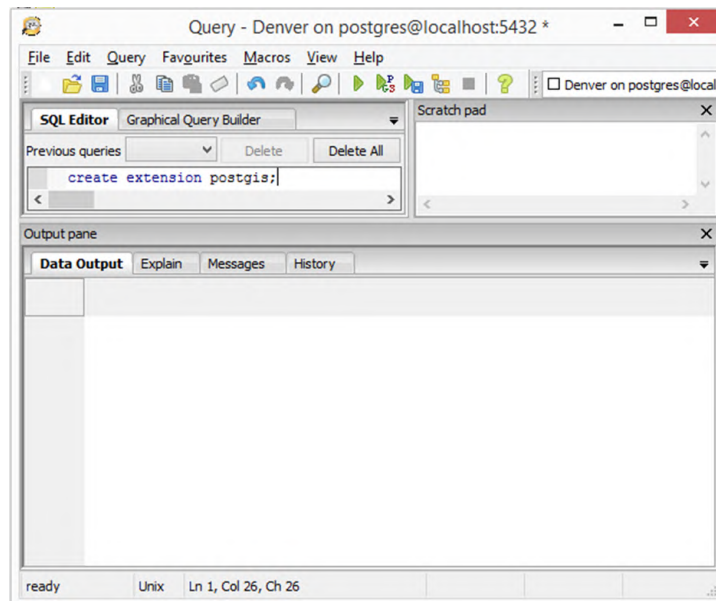


Double click the Databases category and select New Database...



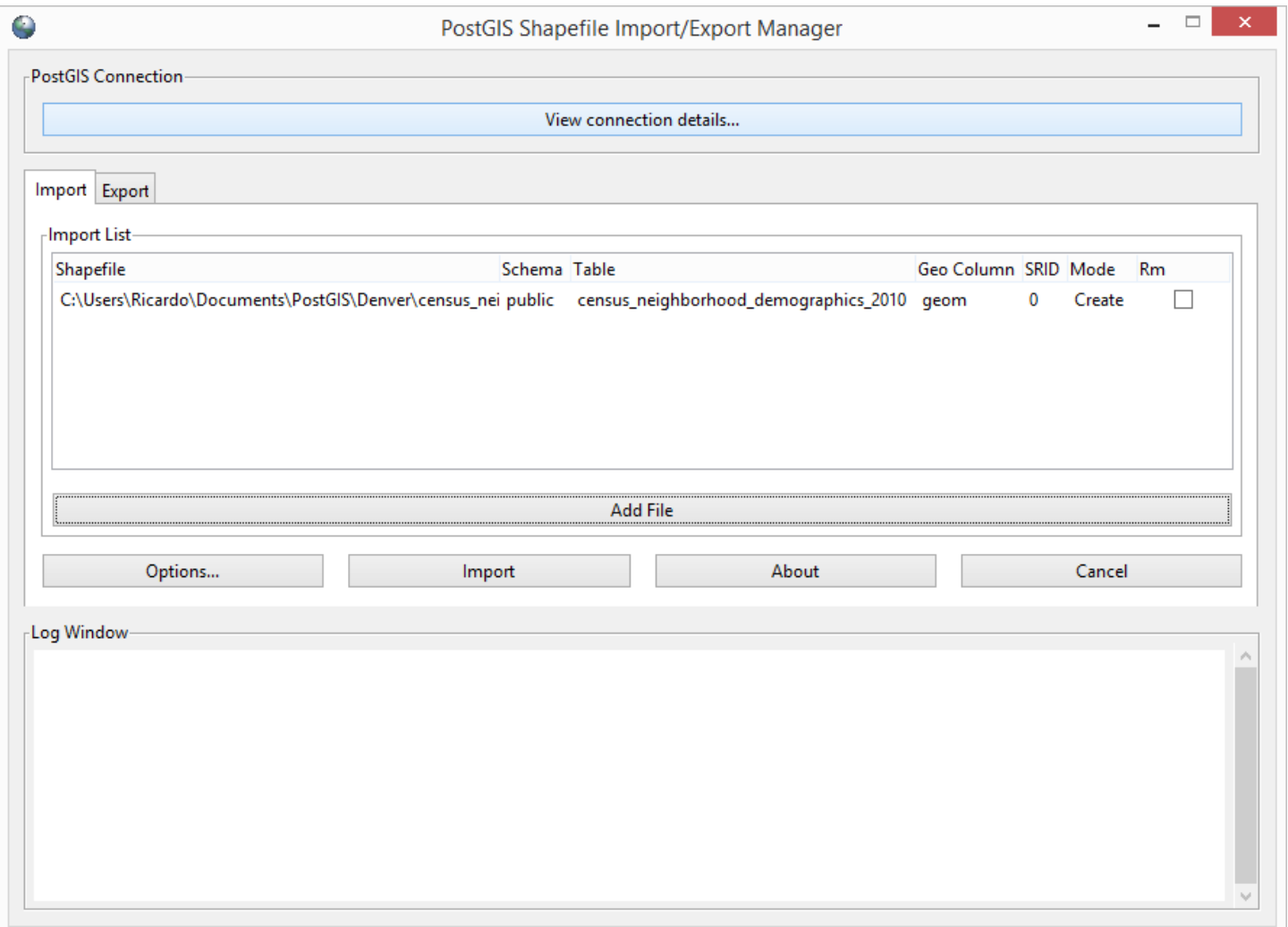
Call it *Denver*, choose *postgres* as owner and press OK.

Now we have to load the PostGIS extension inside this database. Open the SQL query builder.



Type: `create extension postgis;` and press execute query or f5.

Now using the PostGIS Importer let's load the Denver neighborhood shapefile into our new database. Open the PostGIS importer and load the **census_neighborhood_demographics.shp** file into the importer.

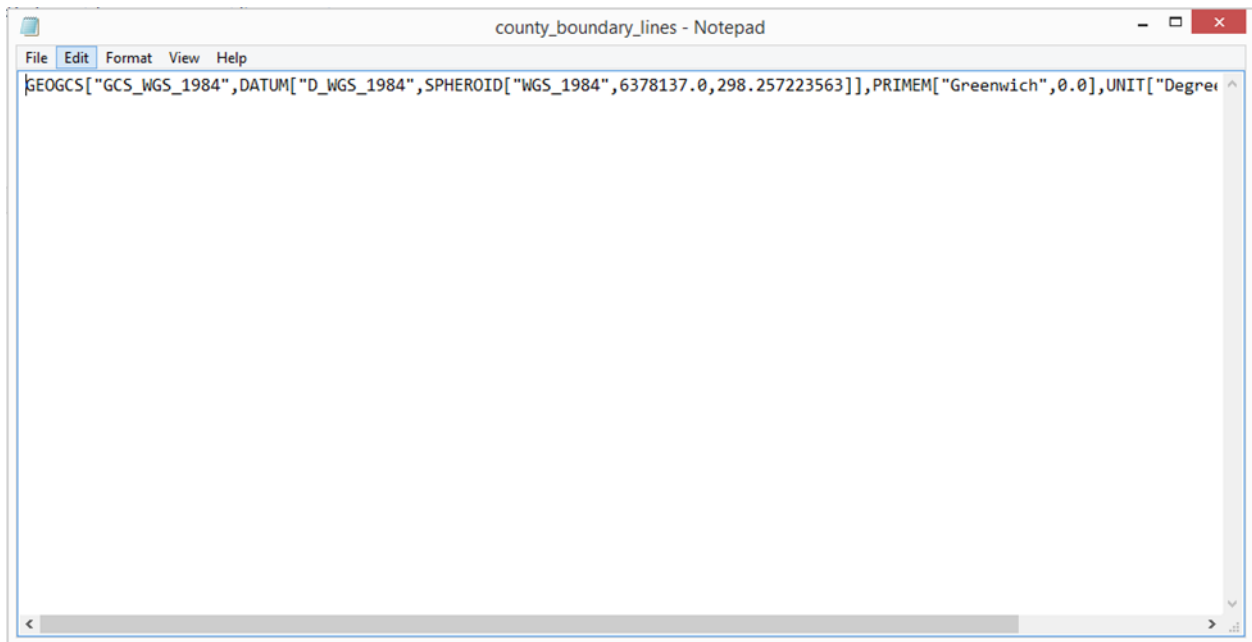


FAQ: But what is a SRID?

SRID stands for Spatial Reference Identifier, it is a single code that compiles all the information about the data's geographic coordinate system and projection.

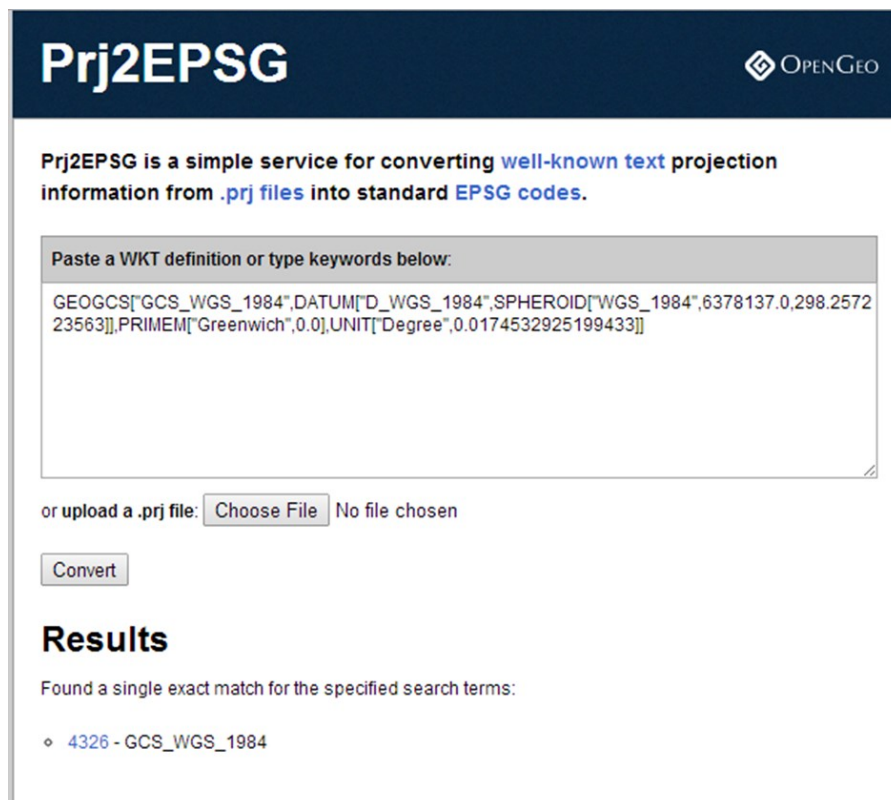
More information can be found at: <http://en.wikipedia.org/wiki/SRID>


Even though there are more than 3000 codes you don't need to know every single of them. For our Denver data we have to do some investigation. Open the .prj file in a word pad.



```
county_boundary_lines - Notepad
File Edit Format View Help
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree"
```

Here we have all the required information to have our data projected. One easy way to figure out which SRID code refers to this system is to copy all this data and paste it into the <http://prj2epsg.org/search> website.



Prj2EPSG 

Prj2EPSG is a simple service for converting **well-known text projection information** from **.prj files** into standard **EPSG codes**.

Paste a WKT definition or type keywords below:

```
GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984",6378137.0,298.257223563]],PRIMEM["Greenwich",0.0],UNIT["Degree",0.0174532925199433]]
```

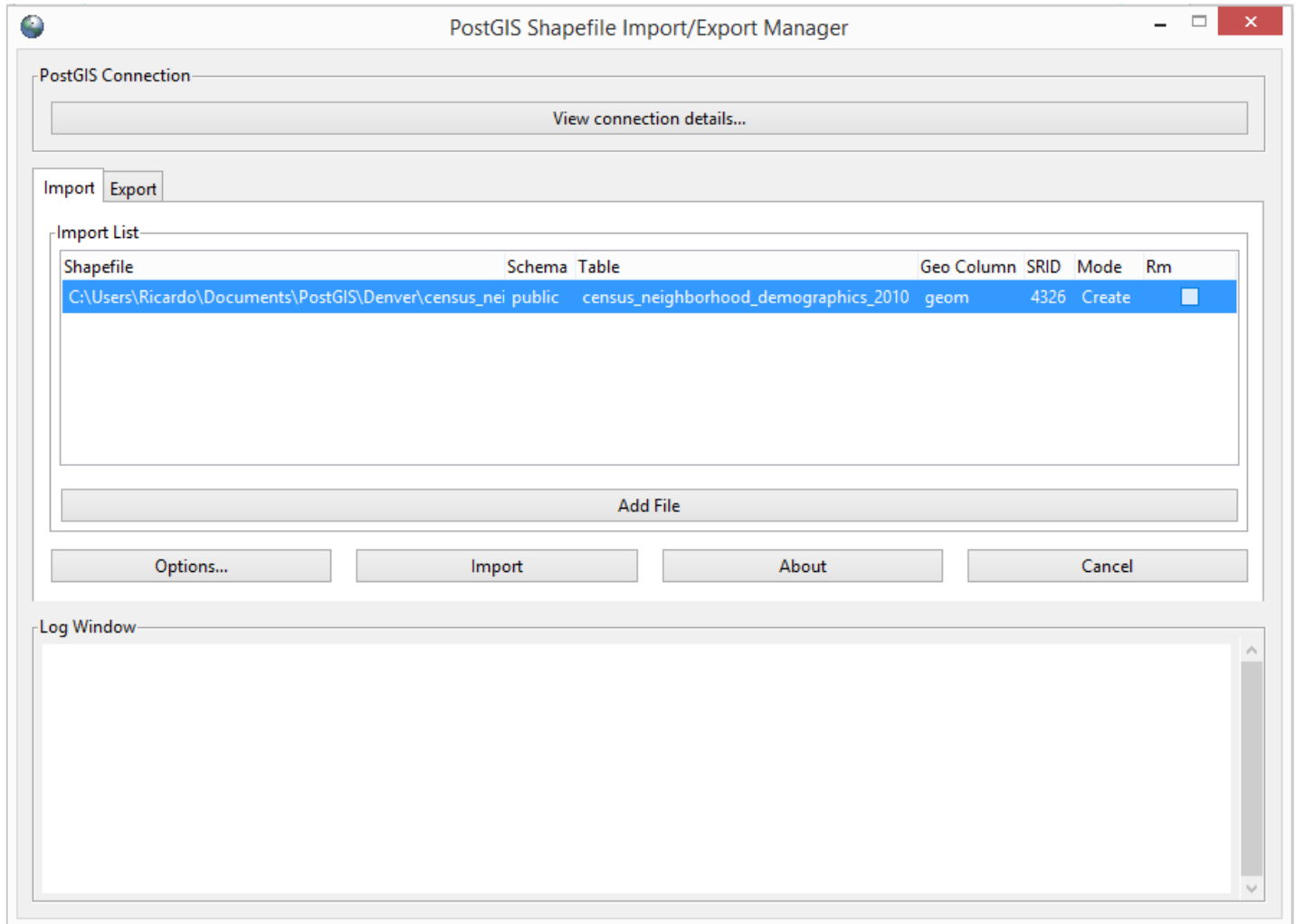
or upload a .prj file: No file chosen

Results

Found a single exact match for the specified search terms:

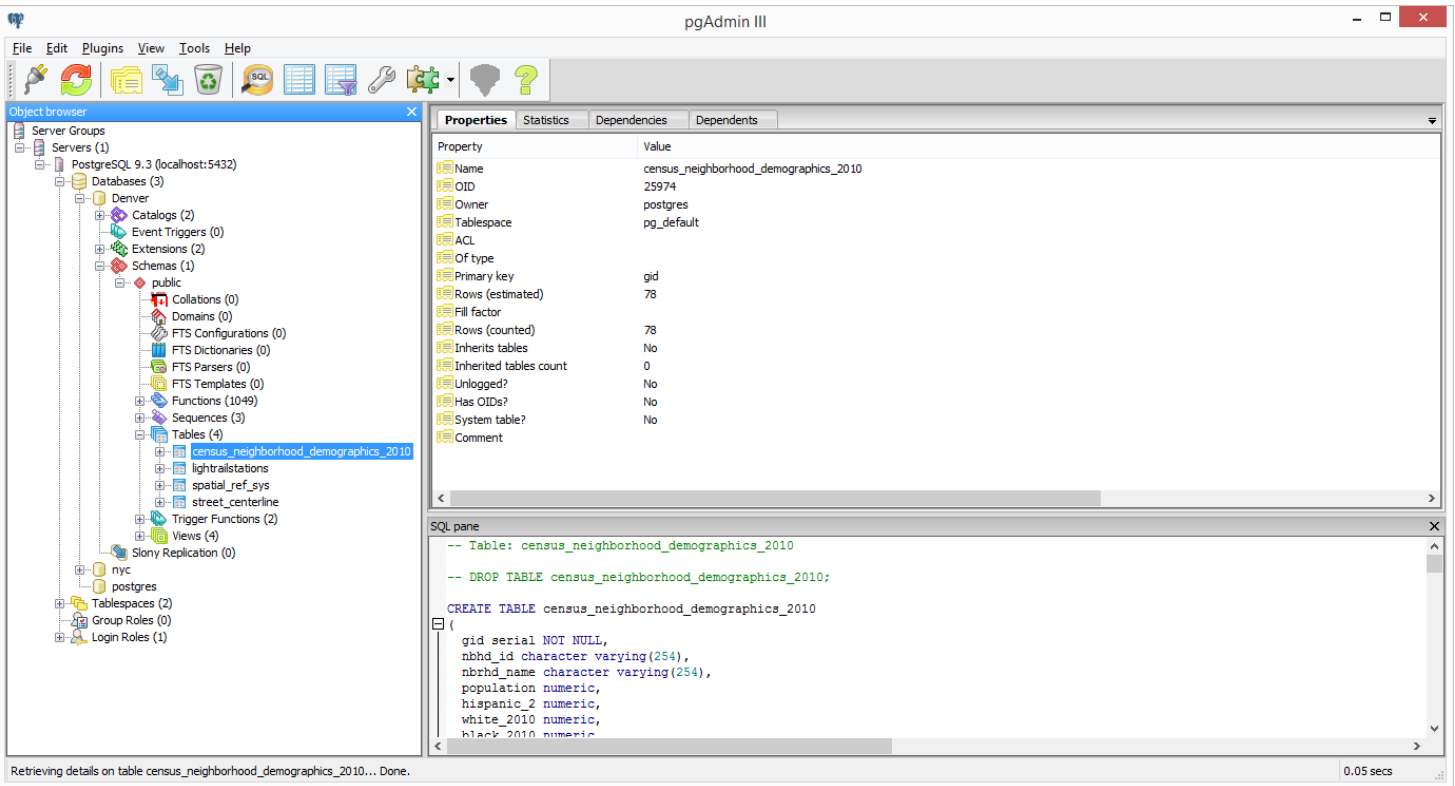
- 4326 - GCS_WGS_1984

Now we know that code 4326 refers to GCS_WGS_1994. Let's plug this number in the PostGIS importer.



You may have notice that this data is in fact unprojected data, in order to work if it geometries we will have to reproject it, this procedure will be covered on the fifth tutorial of this series.

Now you're ready to import your first data into the database. If you want to import several data at the same just keep adding new data and updating their SRID. Press Import.



The screenshot shows the pgAdmin III interface. On the left, the Object browser displays a tree view of the database structure. The 'census_neighborhood_demographics_2010' table is selected under the 'public' schema. The Properties pane on the right shows the table's details, including its name, OID, owner, tablespace, and column definitions. The SQL pane at the bottom displays the SQL code used to create the table.

Property	Value
Name	census_neighborhood_demographics_2010
OID	25974
Owner	postgres
Tablespace	pg_default
ACL	
Of type	
Primary key	gid
Rows (estimated)	78
Fill factor	
Rows (counted)	78
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	

```
-- Table: census_neighborhood_demographics_2010
-- DROP TABLE census_neighborhood_demographics_2010;
CREATE TABLE census_neighborhood_demographics_2010
(
gid serial NOT NULL,
nbhd_id character varying(254),
nbrhd_name character varying(254),
population numeric,
hispanic_2 numeric,
white_2010 numeric,
black_2010 numeric
)
```

Congratulations, your first data is already inside your first database!
(More data will be add on the third tutorial.)