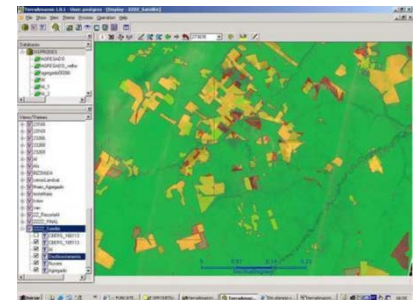
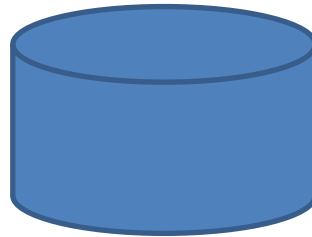
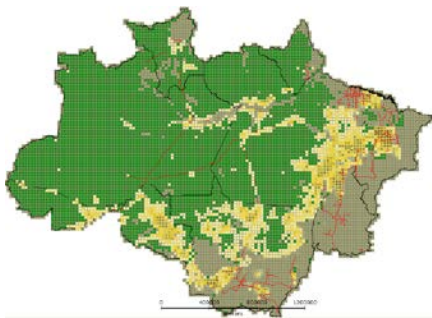


Spatial Databases: Lecture 5

Institute for Geoinformatics
Winter Semester 2014



Malumbo Chipofya: room 109

Topic Overview

1. Prelude: Data and problem solving in science and applications
2. The Relational Database model
- 3. Interacting with relational databases**
- 4. Spatial Relational Database Management Systems**
5. Applications: Terraview and Terralib: Prof. Dr. Gilberto Camara
6. A sample of Nosql Databases: brief introductions + example applications
 - a. Array databases: SciDB
 - b. Document databases: MongoDB
 - c. Graph databases: Neo4J
7. Summary of all lectures given.

Recap

Practical

Getting Started

- Choose a computer in the lab and stick to it
- All the systems we need for the practical are in the virtual machine (VM) “Win7CIP Local” located in the “VMs” directory on the local disk “C:\”.
- If you have any problems finding, adding, and/or starting the VMs please let me know.
- Once logged in to your VM explore the start menu to see what programs have been preinstalled in your VM. You should see, among others, the following:
 - Neo4j
 - PostGIS
 - Postgresql
 - QGIS

Getting Started: test drive Postgres

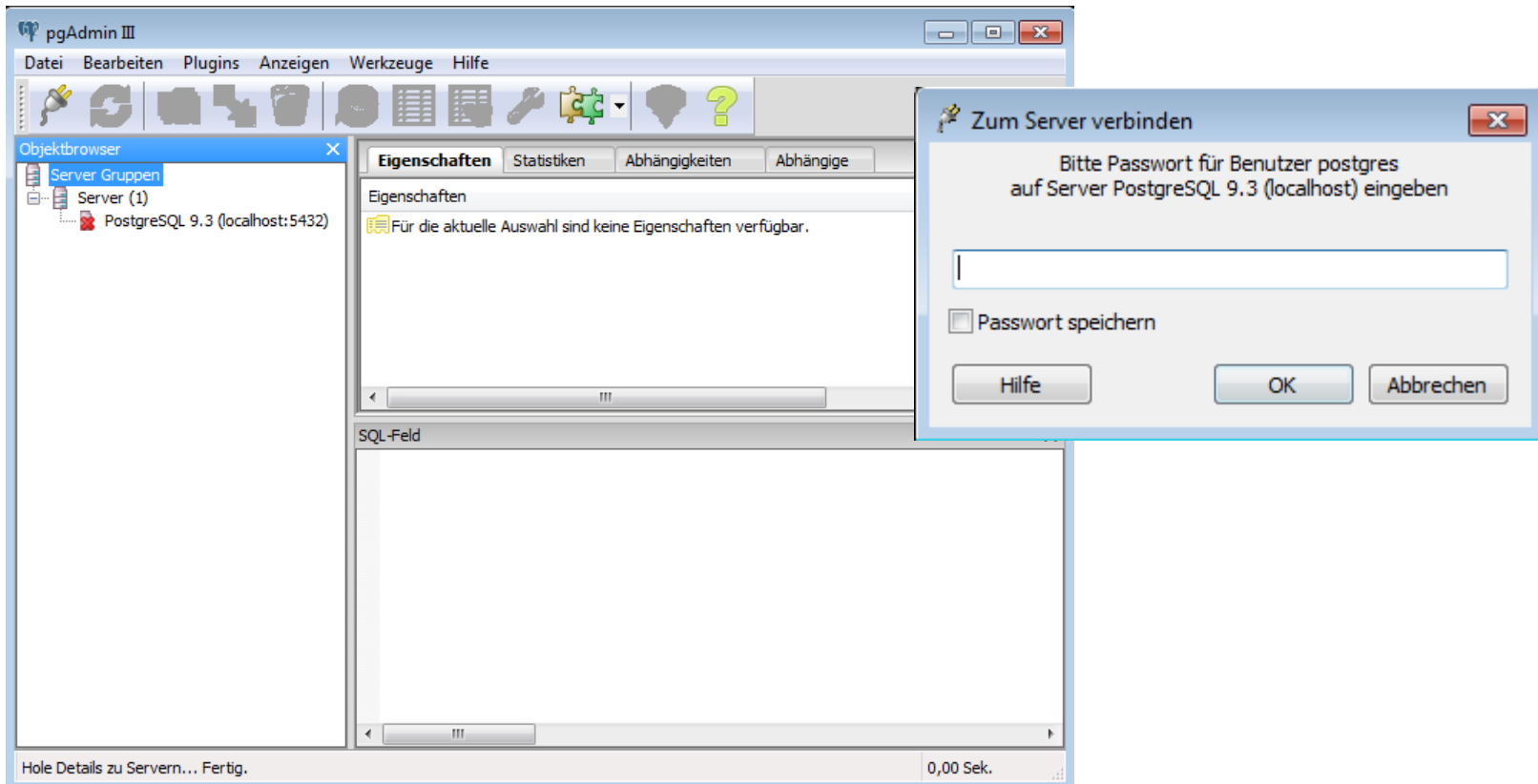
- Go to the postgres folder on the start menu and select SQL Shell (psql).
- Psql is a postgres shell utility for commandline interaction with the postgres server. It is mostly useful for administrative purposes.
- To login as the default user, press enter until you are asked for a password then type “postgres” – that is the default password for the default user (also called “postgres”). Press enter.

Getting Started: test drive Postgres

- Now first change your password by typing “\password” and pressing enter. You will be prompted to enter your new password twice.
- Now create a new database using the “CREATE DATABASE” sql command – don’t forget the ending semi-colon;
- You can now close psql by typing “\q”.
- Now go back to the start menu postgresql folder and start the pgAdmin tool.

Getting Started: test drive Postgres

- You'll see something like this



- Double-click the postgresSQL 9.3 (localhost:5432) icon to login and enter your

Getting Started: test drive Postgres

- Check if your database was created
- If it was then let's enable it for PostGIS by installing postgis in using the following command:

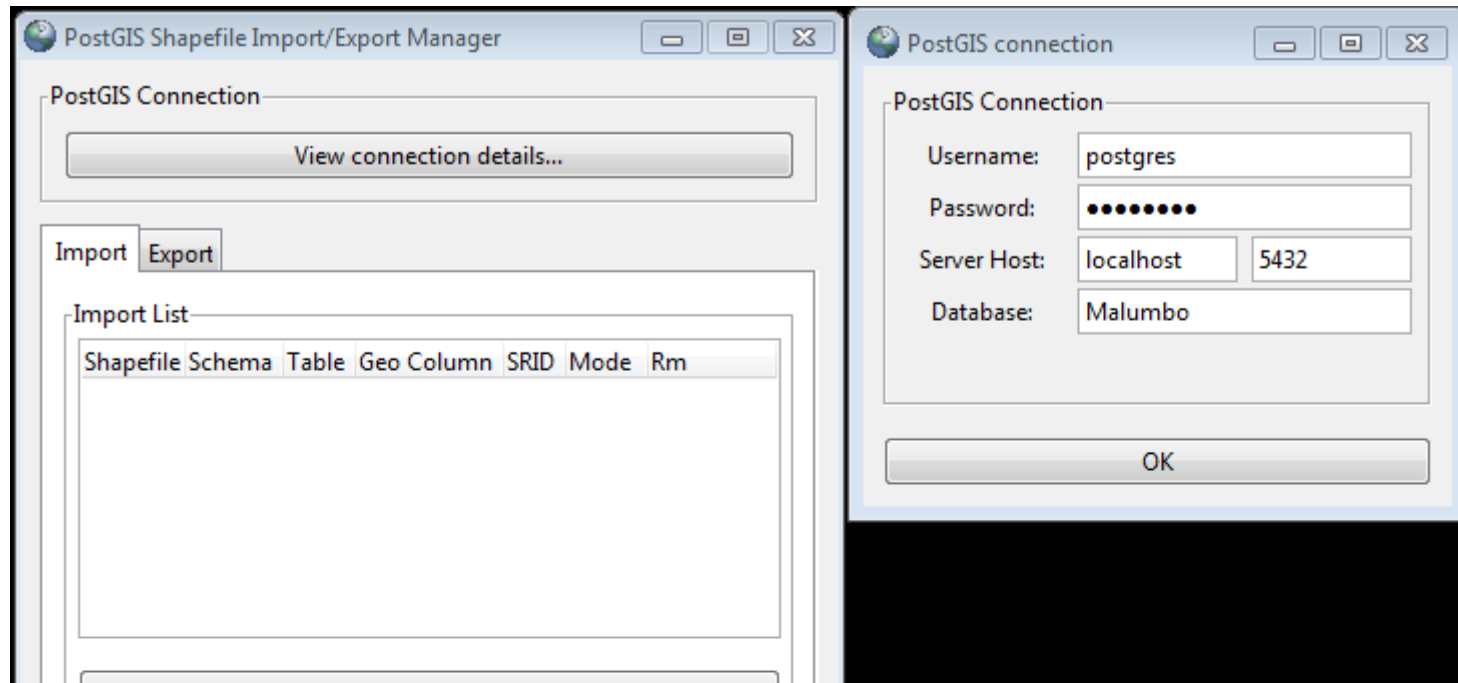
```
CREATE EXTENSION postgis;
```

- You can check if the command was successful by typing

```
SELECT postgis_full_version();
```


Getting Started: test drive Postgres

- Import the data given at http://www.geoinformatics.cc/doku.php?id=spatial_databases_classes
- To import the data use the postgis data import tool found in the postgis folder under the windows menu.

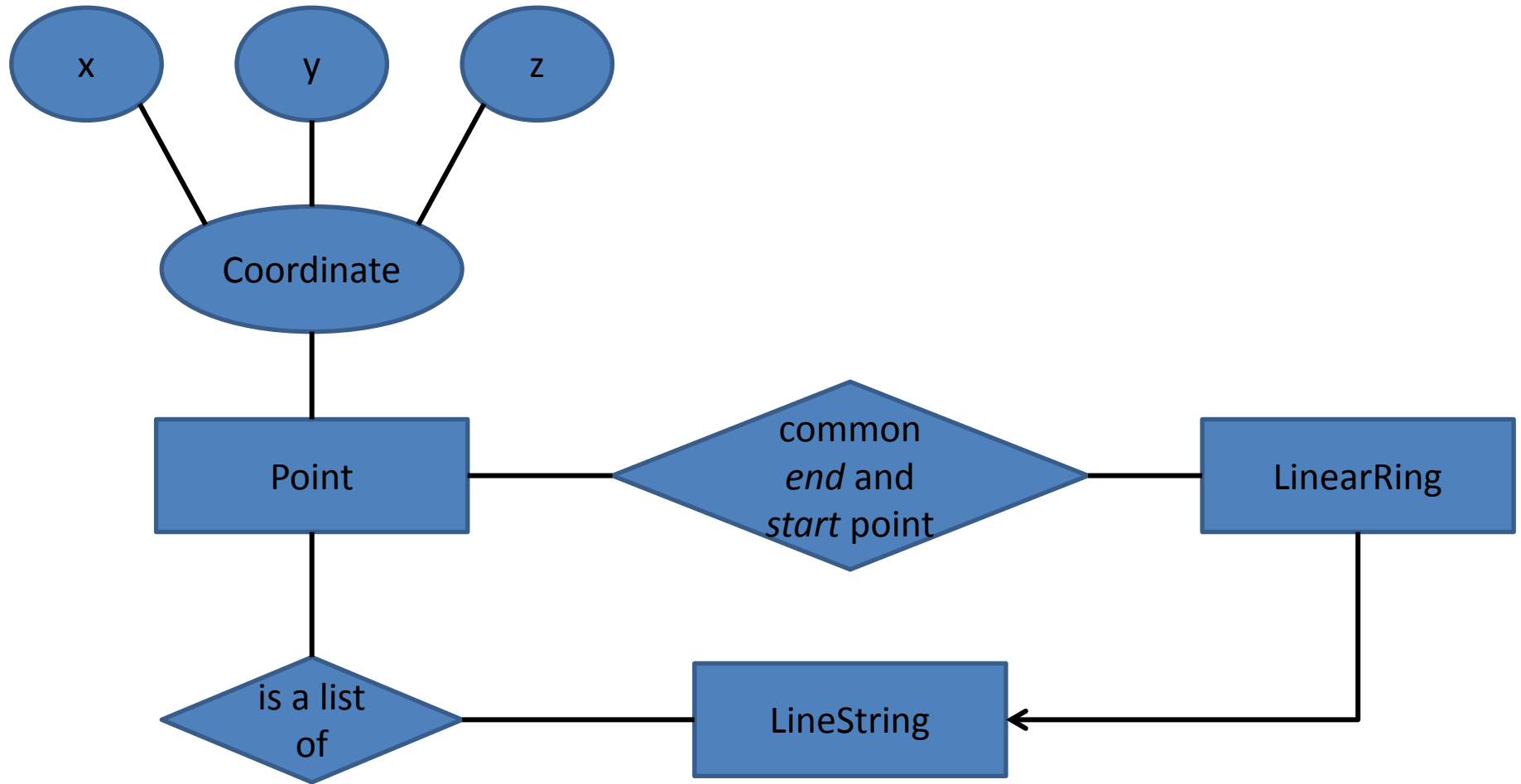


Getting Started: test drive Postgres

- View your data using QGIS: start QGIS Desktop

But let us digress on other
issues for now

Creating of Geometry Storage in Postgresql



Creating of Geometry Storage in Postgresql

```
CREATE DATABASE mygeoms;
```

```
CREATE TABLE point (  
    pid serial,  
    x real,  
    y real,  
    z real  
);
```

```
CREATE TABLE LineString (  
    lsid serial,  
    someAttribute text  
);
```

Creating of Geometry Storage in Postgresql

```
CREATE TABLE LineStringPointLists (  
  LString integer REFERENCES LineString (lsid ) ON  
  DELETE CASCADE,  
  point1 integer REFERENCES point (pid) ON  
  DELETE RESTRICT,  
  point2 integer REFERENCES point (pid) ON  
  DELETE RESTRICT,  
  PRIMARY KEY (LString, point1)  
);
```

Creating of Geometry Storage in Postgresql

```
CREATE TABLE LinearRing (  
  LRing integer REFERENCES LineString (lsid ) ON  
  DELETE RESTRICT,  
  cpoint integer REFERENCES point (pid) ON  
  DELETE RESTRICT,  
  PRIMARY KEY (LRing, cpoint)  
);
```

Creating of Geometry Storage in Postgresql

- What must we do to consistently add a
 1. Point
 2. LineString
 3. LinearRing
- What about asking the following
 - Get me all LinearRings that are equal (have exactly the same vertices)

Creating of Geometry Storage in Postgresql

- Insert LineString
 - Add a record to the LineString relation
 - Starting with the first vertex of the line string, insert consecutive vertex points into the LineStringPointLists relation.
 - End

Creating of Geometry Storage in Postgresql

- So we need some sort of procedural facility to do this
- Enter PL/SQL (Procedural Language SQL)

```
DROP FUNCTION IF EXISTS LineString_Verbose(integer, text) CASCADE;
```

```
/*
```

```
*Function to insert LineString
```

```
*/
```

```
CREATE OR REPLACE FUNCTION LineString_Verbose(LStrId integer, point_ids text) RETURNS VOID AS $$
```

```
BEGIN
```

```
    FOR EACH point1, point2 consecutive in point_ids DO
```

```
        INSERT INTO LineStringPointLists VALUES (LStrId, point1, point2);
```

```
    END LOOP;
```

```
RETURN;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

Creating of Geometry Storage in Postgresql

- Starting with a clean slate – immediately before (re)-creating the tables do:

DROP TABLE IF EXISTS LinearRing CASCADE;

DROP TABLE IF EXISTS LineStringPointLists
CASCADE;

DROP TABLE IF EXISTS LineString CASCADE;

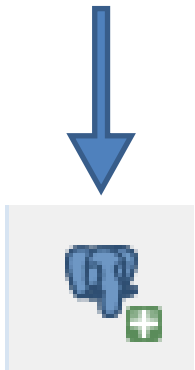
DROP TABLE IF EXISTS point CASCADE;

Creating of Geometry Storage in Postgresql

- What other problems would we face with this type of geometry persistence?
 1. ??
 2. ??
 3. ??
 4. ??
 5. ??

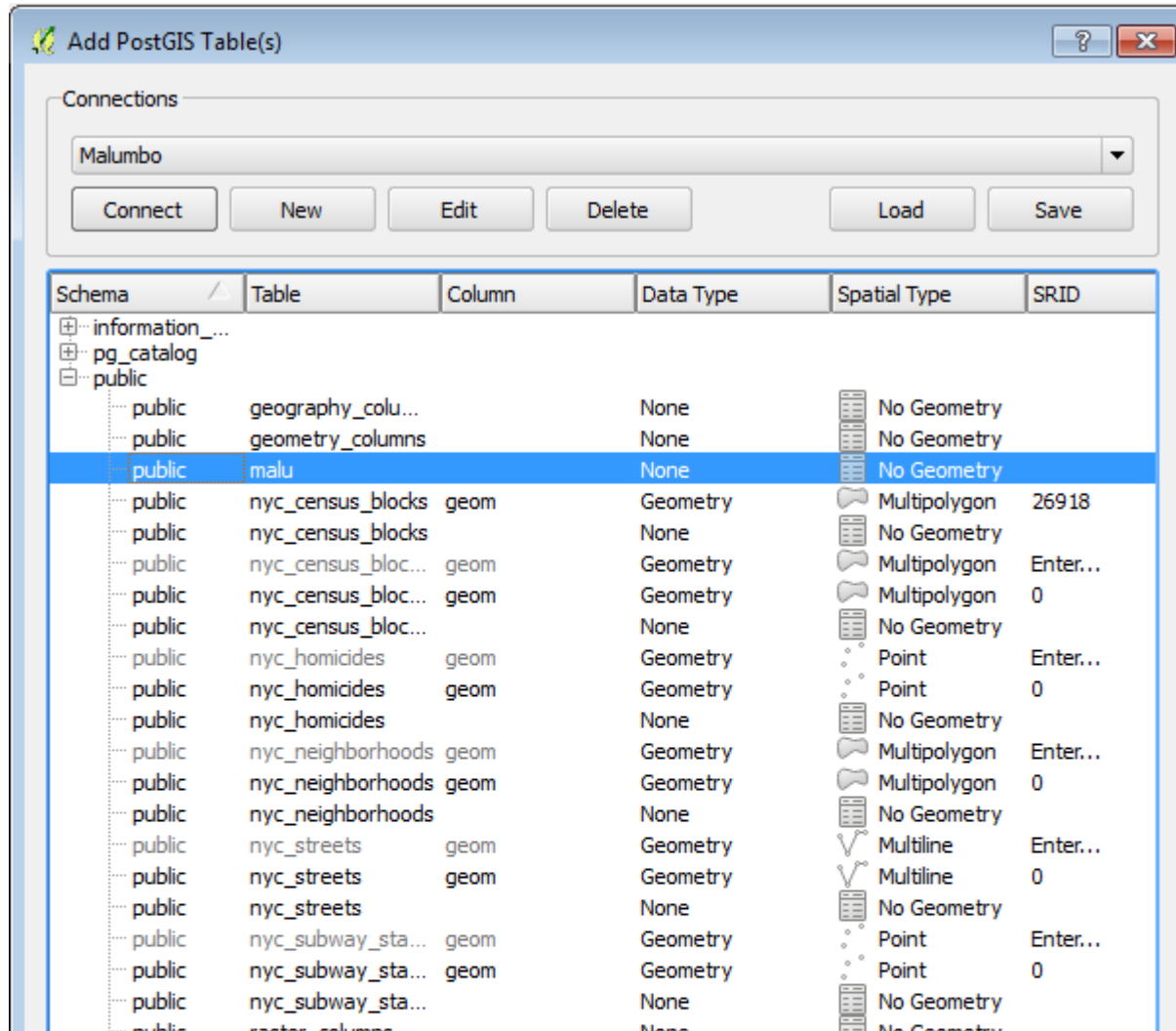
Back to the Getting Started

- View your data using QGIS: start QGIS Desktop

A screenshot of the 'Create a New PostGIS connection' dialog box. The dialog has a title bar with a question mark and a close button. The main area is titled 'Connection Information' and contains several input fields: 'Name' (malumboc), 'Service' (empty), 'Host' (localhost), 'Port' (5432), 'Database' (Malumbo), 'SSL mode' (disable), 'Username' (postgres), and 'Password' (masked with dots). Below these fields are several checkboxes: 'Save Username' (unchecked), 'Save Password' (unchecked), 'Only show layers in the layer registries' (unchecked), 'Don't resolve type of unrestricted columns (GEOMETRY)' (unchecked), 'Only look in the 'public' schema' (unchecked), and 'Also list tables with no geometry' (checked). A 'Test Connect' button is located to the right of the 'Save Username' and 'Save Password' checkboxes.

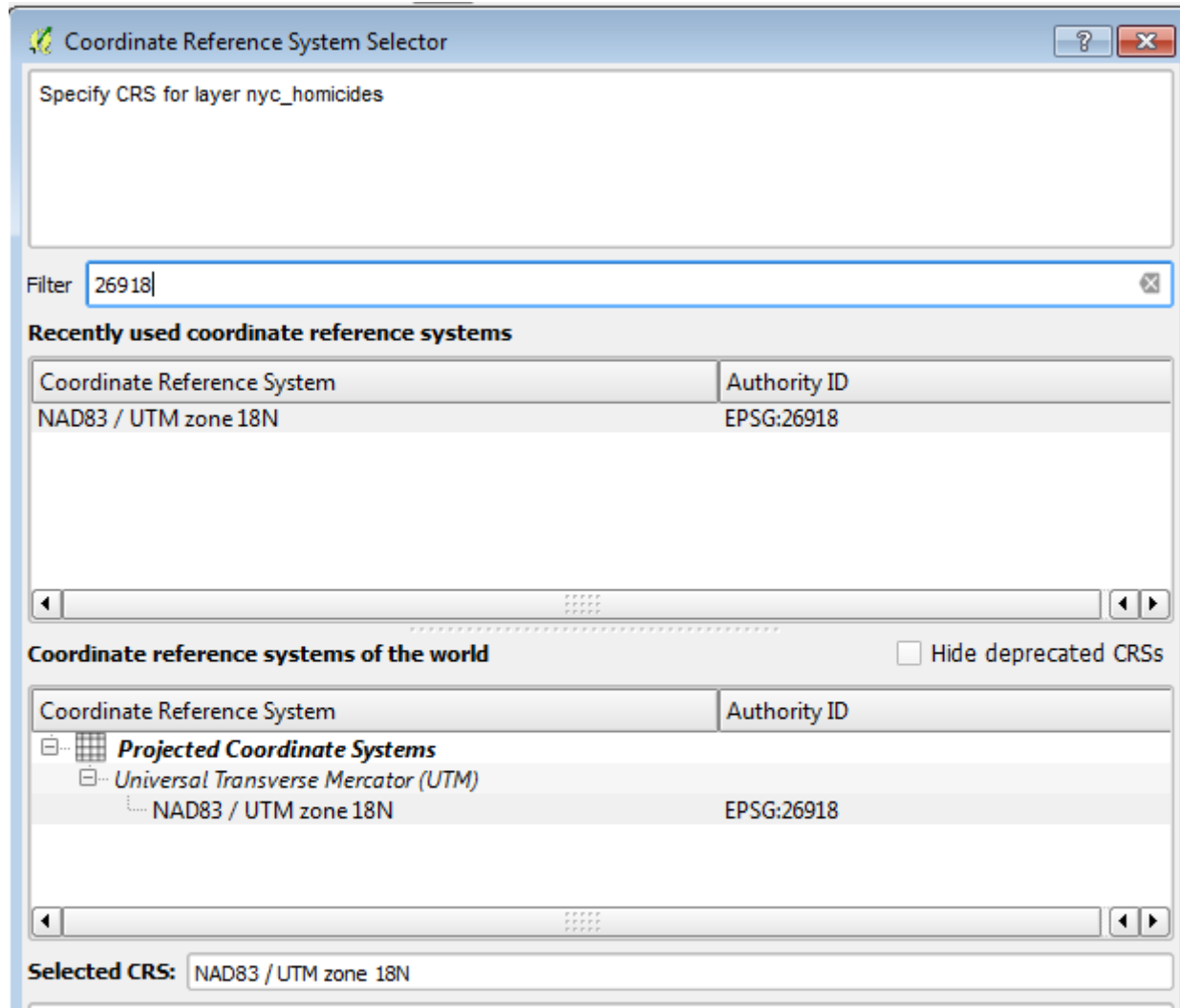
Back to the Getting Started

- View your data using QGIS: start QGIS Desktop



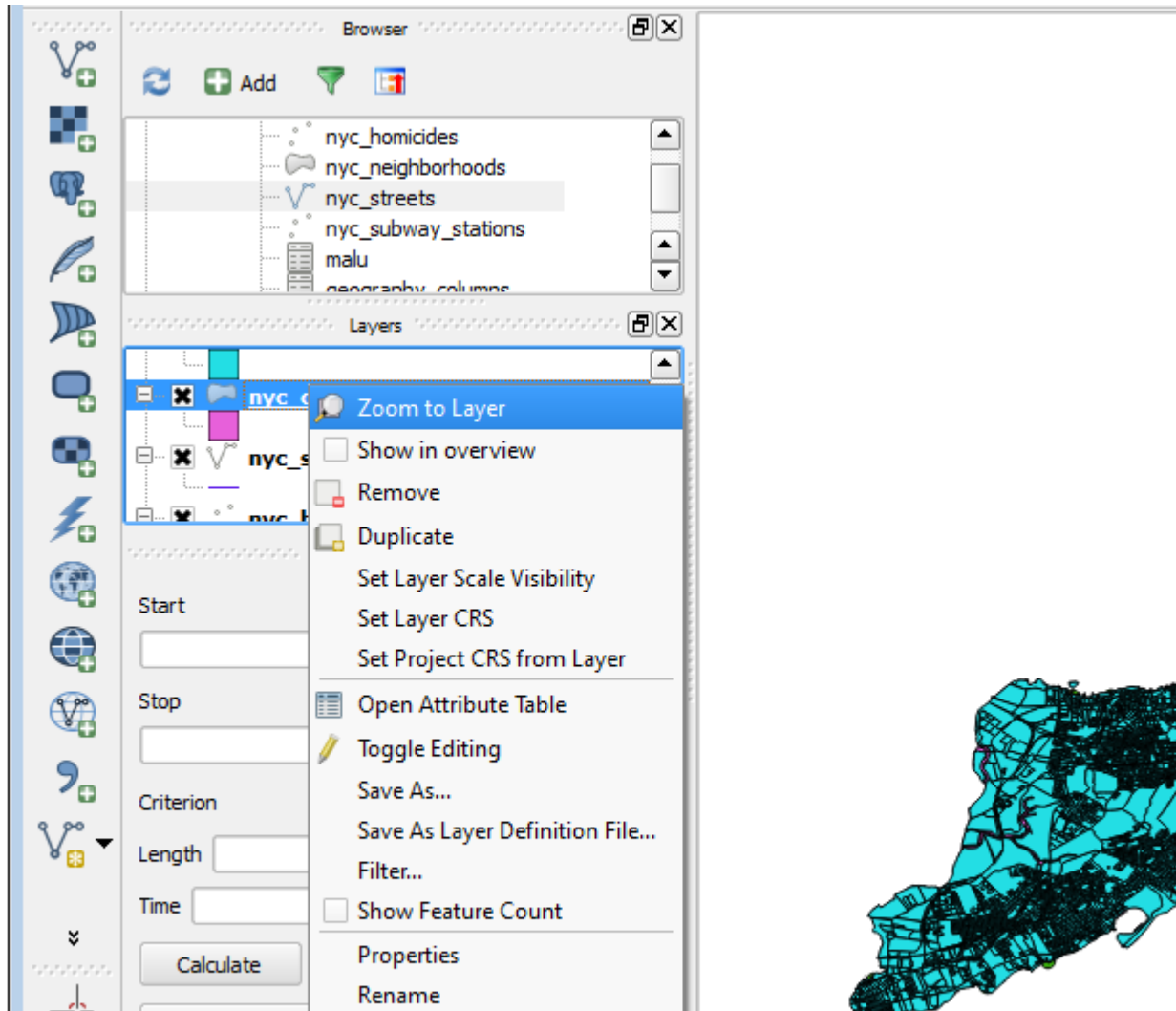
Back to the Getting Started

- View your data using QGIS: start QGIS Desktop



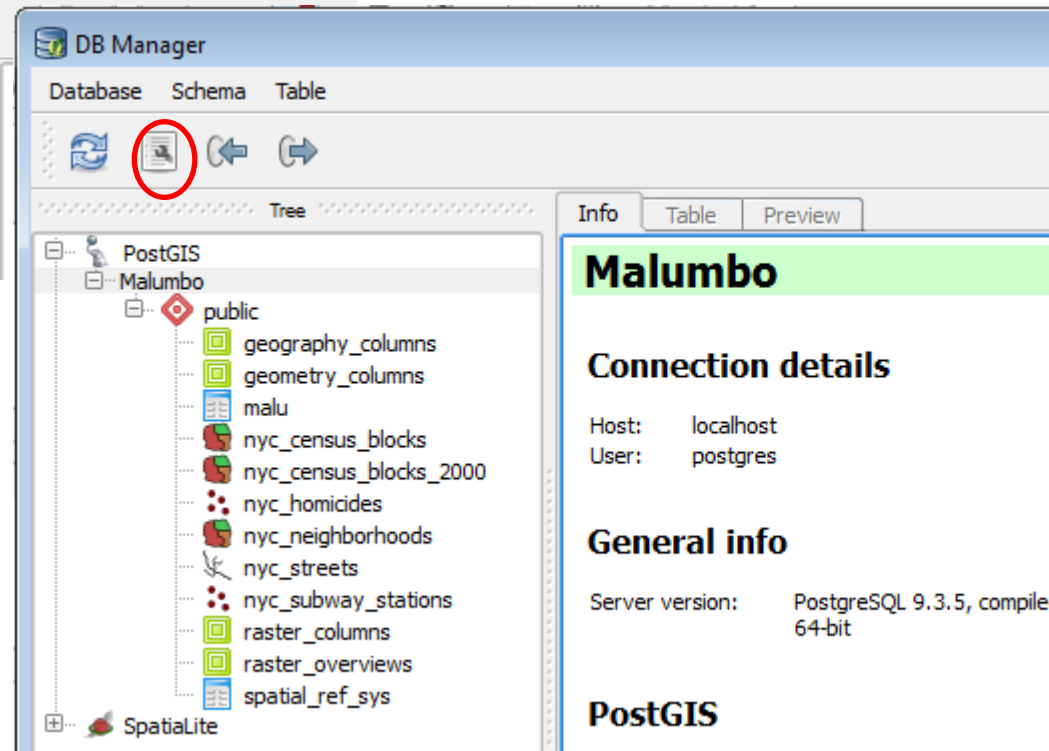
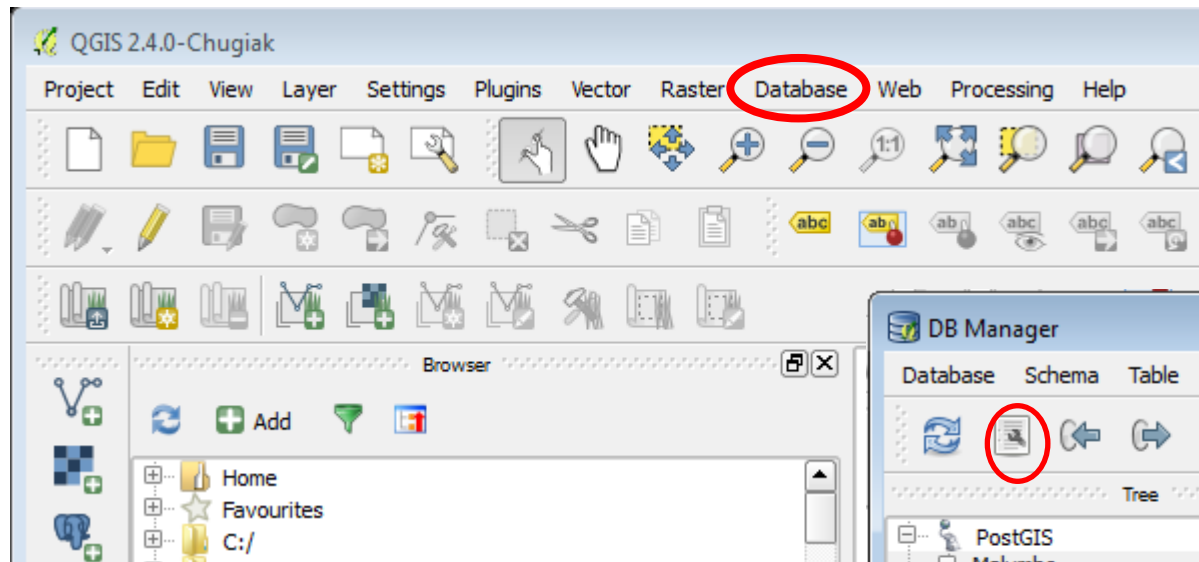
Back to the Getting Started

- View your data using QGIS: start QGIS Desktop



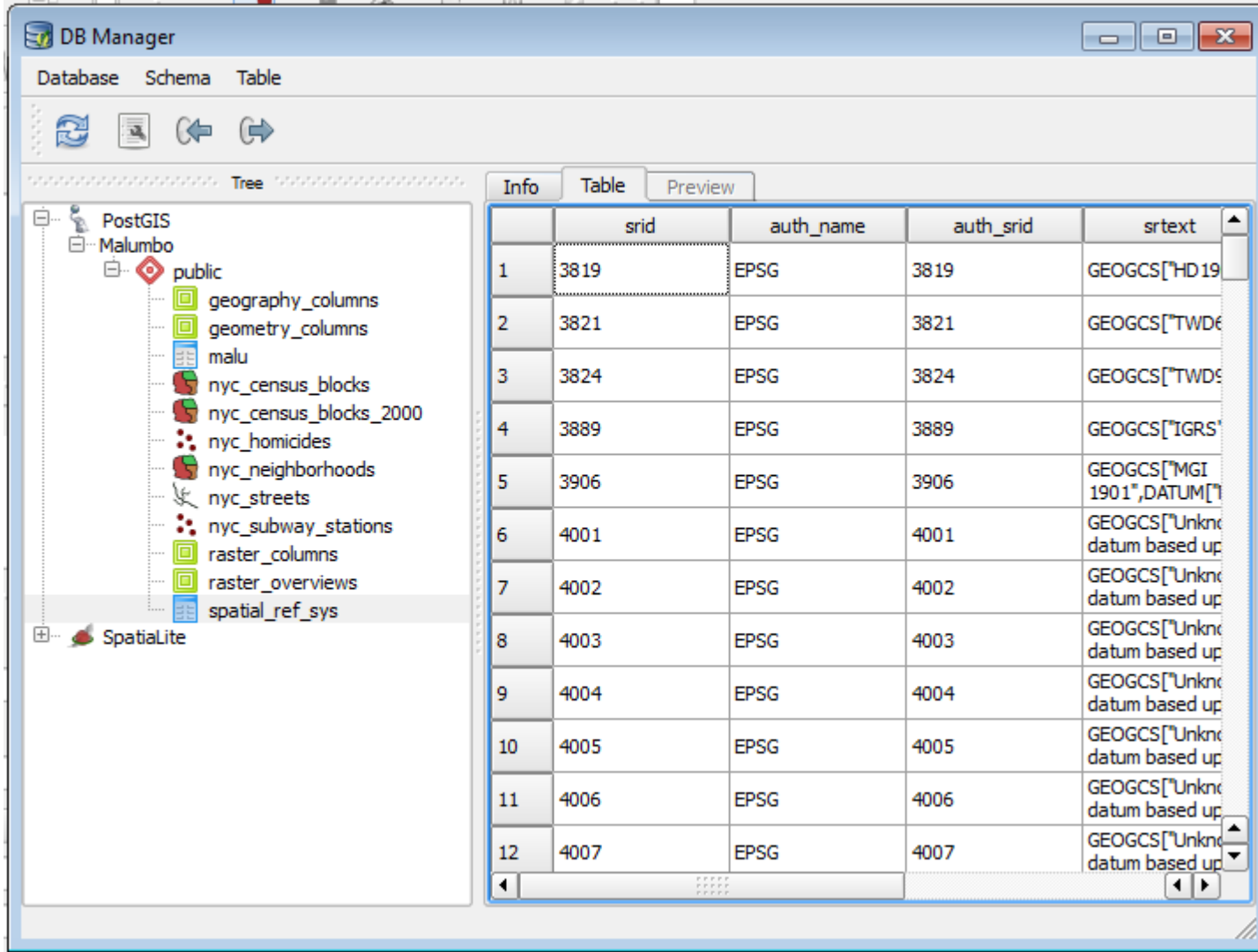
Back to the Getting Started

- View your data using QGIS: start QGIS Desktop



Back to the Getting Started

- View your data using QGIS: start QGIS Desktop



The screenshot shows the DB Manager application window. The left pane displays a tree view of the database structure:

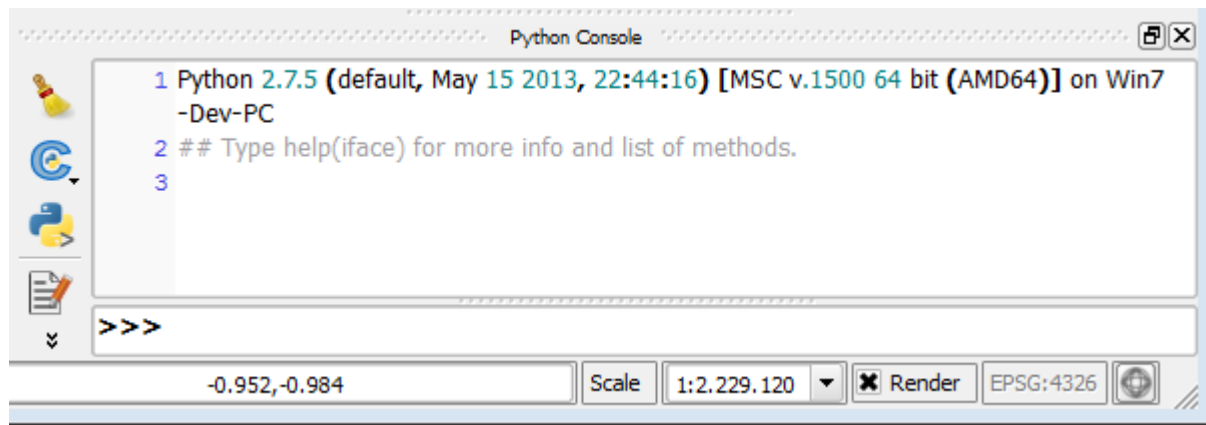
- PostGIS
 - Malumbo
 - public
 - geography_columns
 - geometry_columns
 - malu
 - nyc_census_blocks
 - nyc_census_blocks_2000
 - nyc_homicides
 - nyc_neighborhoods
 - nyc_streets
 - nyc_subway_stations
 - raster_columns
 - raster_overviews
 - spatial_ref_sys
- SpatialLite

The right pane shows the 'Table' view of the 'spatial_ref_sys' table, displaying the following data:

	srid	auth_name	auth_srid	srtext
1	3819	EPSG	3819	GEOGCS["HD19
2	3821	EPSG	3821	GEOGCS["TWDE
3	3824	EPSG	3824	GEOGCS["TWDS
4	3889	EPSG	3889	GEOGCS["IGRS'
5	3906	EPSG	3906	GEOGCS["MGI 1901",DATUM["
6	4001	EPSG	4001	GEOGCS["Unknc datum based up
7	4002	EPSG	4002	GEOGCS["Unknc datum based up
8	4003	EPSG	4003	GEOGCS["Unknc datum based up
9	4004	EPSG	4004	GEOGCS["Unknc datum based up
10	4005	EPSG	4005	GEOGCS["Unknc datum based up
11	4006	EPSG	4006	GEOGCS["Unknc datum based up
12	4007	EPSG	4007	GEOGCS["Unknc datum based up

Back to the Getting Started

- View your data using QGIS: using Python in QGIS.
- We'll follow the tutorial at http://docs.qgis.org/testing/en/docs/pyqgis_developer_cookbook/intro.html



```
Python Console
1 Python 2.7.5 (default, May 15 2013, 22:44:16) [MSC v.1500 64 bit (AMD64)] on Win7
  -Dev-PC
2 ## Type help(iface) for more info and list of methods.
3
>>>
```

That's all for today

Thank you!

Questions?