intro to postgis

Basic GIS

- Maps
- Projection
- X,Y,(Z),time
- types, point, line,
- Attributes

Projection

- Define standard on what coordinate system you will use
- SRID=4326, X and Y coordinates min -180, -90 max 180 -90 flat
- Others for different purposes, some define a z axis, time axis etc

File types

- shp
- GML
- KML
- geojson
- georss

GIS objects

Geography data type

- 2d plane

Geometry data type

- coordinates mapped to a sphere
- harder math, not as complete

Data types

POINT(0 0)

LINESTRING(0 0,1 1,1 2)

POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))

MULTIPOINT(0 0,1 2)

MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))

MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1 -2,-2 -2,
-2 -1,-1 -1)))

GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4))

```
ST_AsText()
ST_GeomFromEWKT('SRID=4326; POINT(-10 30)');
st_intersects(geo,geo)
st_contains(geo,geo)
ST_Area(geo)
```

Installation:

Install postgresql

Install postgresql-9.3-postgis-2.1

CREATE EXTENSION postgis;

CREATE EXTENSION postgis_topology;

create type json;

sample data

data.sa.gov.au

- Suburbs shp file
- Roads - shp file
- Earthquakes - csv with long lat
- playgrounds - ; seperated file with long lat

Suburbs

```
http://www.dptiapps.com.au/dataportal/Suburbs.zip
unzip Suburbs.zip
shp2pgsql -s 4326 -e  Suburbs.shp > Suburbs.sql
psql playground < Suburbs.sql
```

Roads

```
http://www.dptiapps.com.au/dataportal/Roads.zip
unzip Roads.zip
shp2pgsql -s  4326 -e Roads.shp > Roads.sql
psql playground < Road.sql
```

Earthquakes

http://data.sa.gov.au/storage/f/2013-05-21T05%3A51%3A01.742Z/dmitre-earthquake-new.csv

```
copy earthquake_staging from '/var/lib/postgresql/dmitre-earthquake-new.csv' CSV header;
```

```
create view earthquake as select ROW_NUMBER() OVER () as id, cast(dated||' '||floor(cast(time as integer)/100)||':'||cast(time as integer)%100 as timestamp) dated, GeomFromEWKT('SRID=4326; POINT(' ||longitude ||' '|| LATITUDE||')') as geom,cast(depth as real),place,cast(magnitude as real),cast(intensity as integer),accuracy,cast(arrivals as integer),cast(residual as integer), cast (stations as integer) from earthquake_staging ;
```

Playgrounds

http://data.sa.gov.au/storage/f/2013-05-24T02%3A36%3A40.
663Z/playgrounds-for-datagovau.txt

```
create table playground_staging ( XCoord text, YCoord text, Name text,
Location text, Council text);
```

```
copy playground_staging from '/var/lib/postgresql/playgrounds-for-datagovau.
txt' DELIMITER ';' CSV header;
```

```
create view playground as select ROW_NUMBER() OVER () as id,
GeomFromEWKT('SRID=4326; POINT('||XCoord ||' '|| YCoord||')') as geom,
name,location,council from playground_staging;
```

Software

QGIS

- Desktop software

Geoserver

- Tile map server

Earthquakes by suburbs

```
select suburbs.suburb,count(*) from suburbs,earthquake
where st_intersects (earthquake.geom,suburbs.geom)
group by suburbs.suburb order by 2 desc;
```

Suburb	Count
FLINDERS RANGES	720
MELROSE	378
BOOLEROO CENTRE	223
MANNANARIE	181
WITCHELINA	169

GeoJson example

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [102.0, 0.6]
      },
      "properties": {
        "name: "value0"
      } }  ]}
```

export to GeoJSON

```
SELECT row_to_json(fc) FROM ( SELECT
'FeatureCollection' As type, array_to_json(array_agg(f)) As
features
FROM (SELECT 'Feature' As type, ST_AsGeoJSON(lg.
geom)::json As geometry, row_to_json
((SELECT l FROM (SELECT place,dated,depth,magnitude,
intensity,accuracy,arrivals,residual,stations) As l)) As
properties   FROM earthquake As lg   ) As f )  As fc;
```

importing geojson

```
SELECT json_staging.id AS json_id,
    geomfromewkt('SRID=4326; POINT('::text ||  cast(a.value -> 'geometry'->
'coordinates'->> 0 as text) || ' ' ||  cast(a.value -> 'geometry' -> 'coordinates' ->>
1 as text) || ')') AS geom,
   cast(a.value -> 'properties'->> ‘name' as text) AS name,
   FROM json_staging,
    LATERAL json_array_elements(json_staging.data -> 'features') a;
```