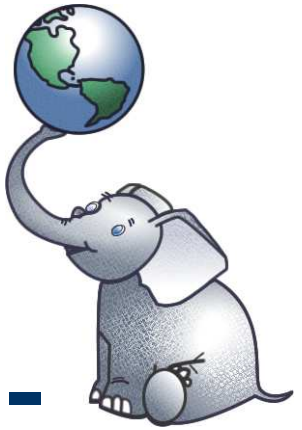


Building Location Aware Apps - Get Started with PostGIS

PART I

Lasma Sietinsone

l.sietinsone@ed.ac.uk

EDINA®

Slides

http://gismatic.com/dev8d/docs/

Who am I?

- Work for **EDINA**, University of Edinburgh
- GIS Technician & Database Engineer
- Manage UK data for **online maps** & services (approaching **terabytes of data**!)
- Provide **geo-support** for spatially enabled apps

My Aim?

- Stimulate your creative thinking!
- Give you an ability to integrate spatial technologies in your own stuff!

Space is a natural join between otherwise seemingly unrelated things!

- Save you hours/days of faffing around on your own!
- Give you a kick for accelerated learning of new technology!

Hands-on!
I MUST get you rolling independently!

Who are you?

- **Who attended prep-workshop?**
- **Experience of audience:**
  - SQL?
  - RDBMS?
  - OS?
- **This demo requires:**
  - SQL – basic knowledge (SELECT, UPDATE, DELETE) – check your cheat-sheet!
  - RDBMS – none
  - Windows but Mac OS, Linux will do too

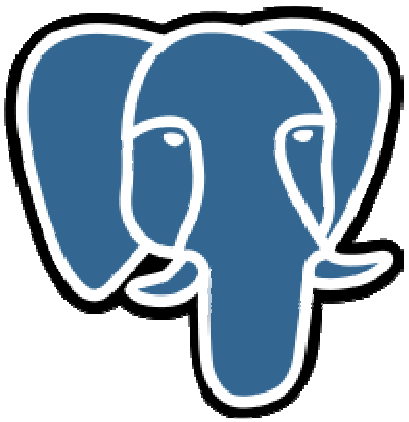
Contents

1. ***Pre-workshop (Install and configure PostgreSQL 9.1 with PostGIS 1.5)***

2. Some background (~15 mins)
3. Practical (1hr & 15 minutes):
   - Import some data
   - Use some PostGIS functions
   - View data externally
   - Demo with OpenStreetMap data to perform location based queries
4. Questions and answers (~15 mins)

EDINA®

What is PostgreSQL?



- Proper object-relational database management system
- Massive OS project
- Super advanced
- Fast!
- Standard compliant
- Well documented
- Global user community

EDINA®

Advantages of a DBMS

- Data Independence from application programs
- Efficient Data Access
- Data Integrity and Security
- Centralized Data Administration
- Concurrent Access and Crash Recovery
- Reduced Application Development Time – common data access functions

PostGIS what?

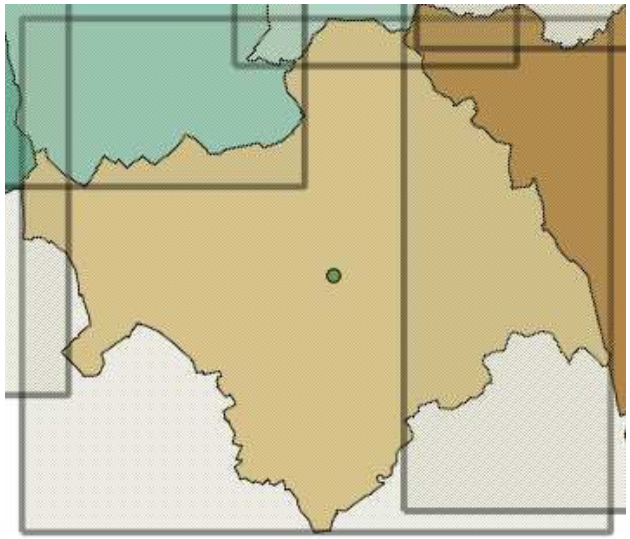
PostGIS is a spatial extension to the PostgreSQL object-relational database that supports **work with geographic data**

EDINA®

PostGIS extension provides

- **Geometry/geography** data **types** for points, linestrings, polygons, curves, geometry collections, 3D geometries

- Use geometry to store object:



(A) shape

(B) centroid

(C) bounding box

EDINA®

PostGIS extension provides (cont.)

- Spatial **operators** such as contains @, overlaps, intersects &&, same as =, is to the left/right/below/above
- Loads of **spatial functions**

Spatial functions

- **Constructors**

  e.g. ST_GeomFromText(), ST_MakePoint(), ST_MakePolygon

- **Output**

  e.g. ST_AsGML(), ST_AsGeoJSON(), ST_AsBinary()

- **Accessors**: *getters and setters*

  e.g. ST_Transform(), ST_GeometryType(), ST_IsValid(), ST_NPoints(), ST_SetDRID()

- **Measurement**

  e.g. ST_Distance_Sphere(), ST_Length_2D(), ST_Perimeter(), ST_Azimuth()

Spatial functions (continued)

- **Decomposition**

  e.g. ST_Centroid(), ST_Box2D(), ST_Boundary(), ST_GeometryN(), ST_DumpRings()

- **Composition**

  e.g. ST_MakePoint(), ST_MakePolygon(), ST_BuildArea(), ST_Polygonize()

- **Simplification**

  e.g. ST_Simplify(), ST_SimplifyPreserveTopology(), ST_SnapToGrid()

Common workflow

1. **Load**/populate **table**
2. **Add geometry column** with certain dimensions and spatial reference system (SRS/SRID)
3. [**Register geometry column** in *geometry_columns* table]
4. Optimise performance by:
   - Creating spatial index
   - Clustering data based on spatial index

Common workflow (continued)

5. View/Edit/Query data with:
   - desktop GIS, e.g. QGIS, OpenJUMP, GRASS etc.
   - Web map servers, e.g. GeoServer, MapServer, Deegree etc.
   - Custom applications communicating via database connectors such as JDBC

Use online references

PostgreSQL docs *****

http://www.postgresql.org/docs/9.0/interactive/index.html

PostGIS docs *****

http://www.postgis.org/docs/