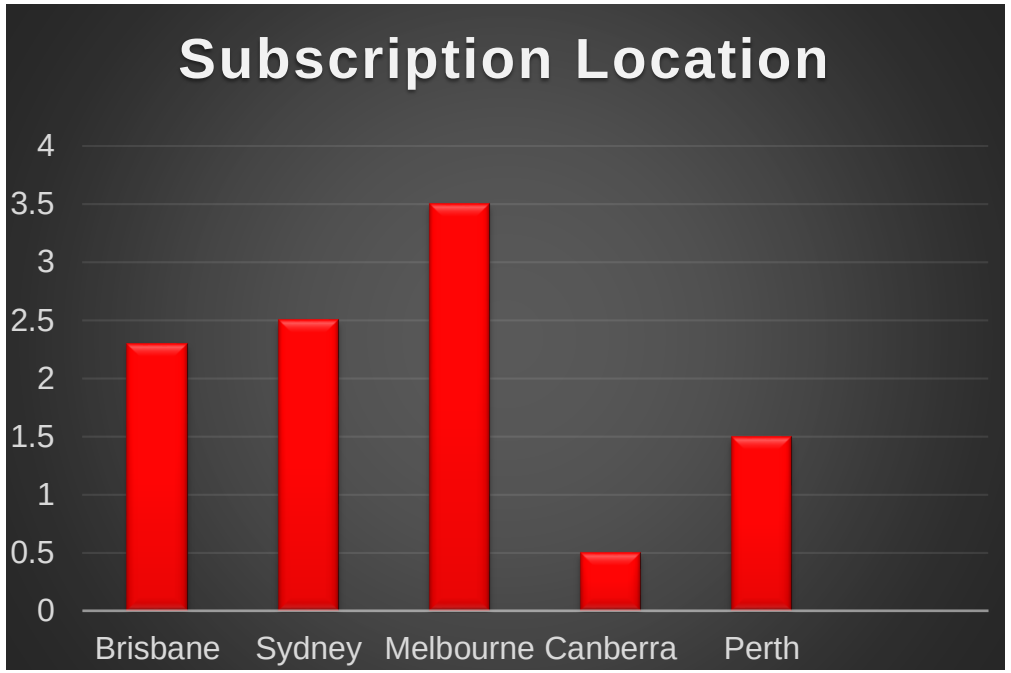
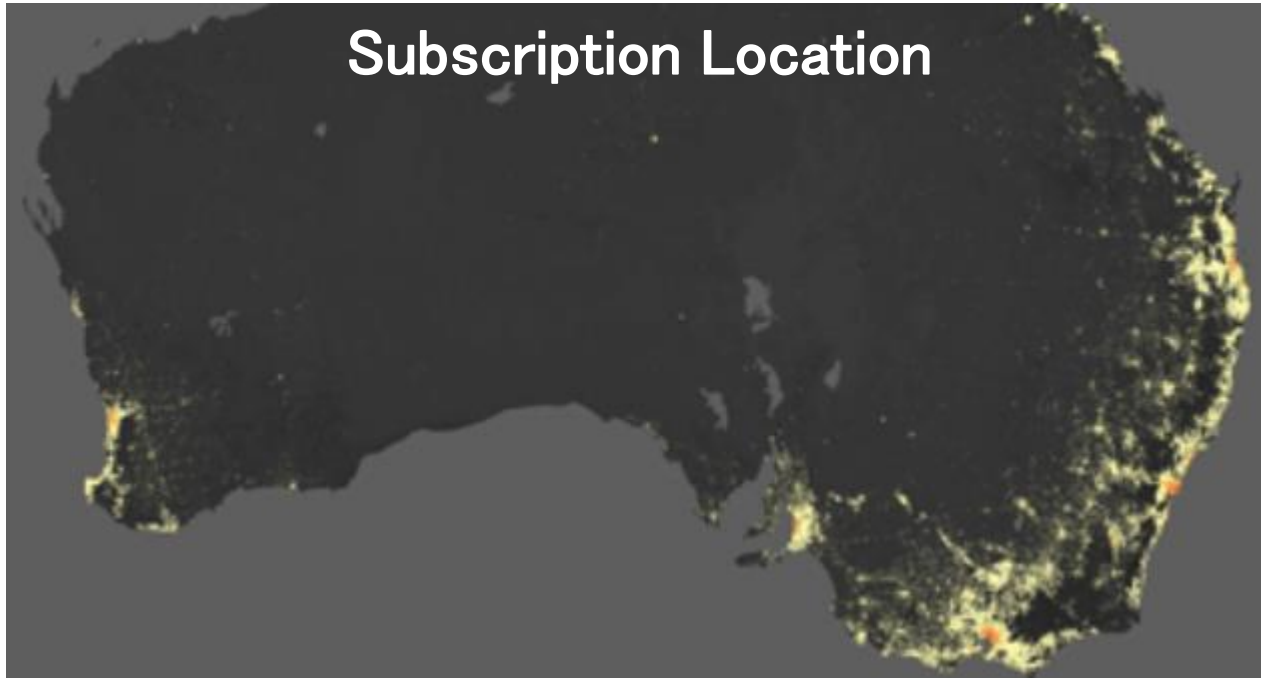
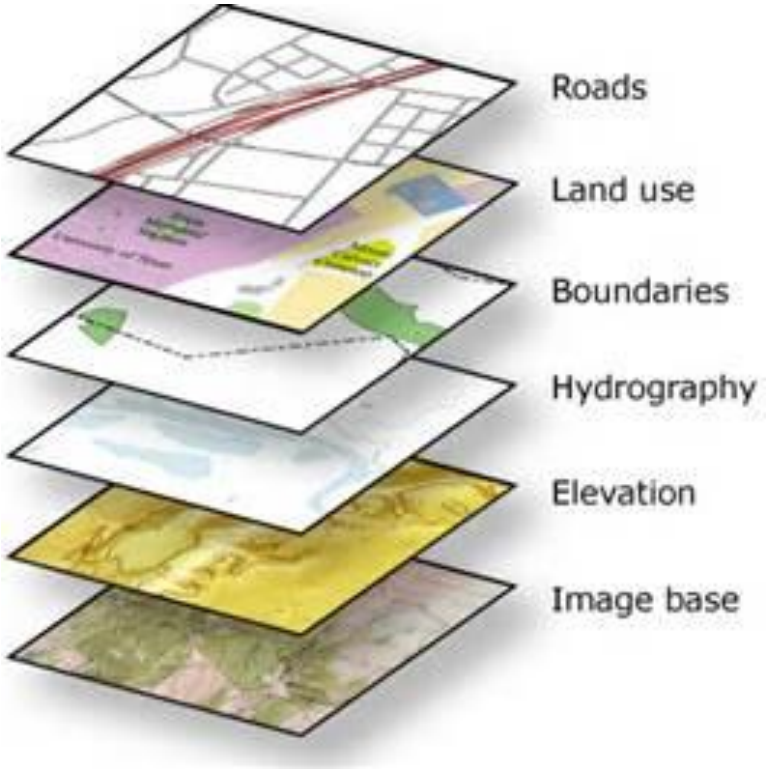
Not Interested in spatial data?
Let me change your mind!


Subscription Location
4
3.5
3
2.5
2
1.5
1
0.5
0
Brisbane
Sydney
Melbourne
Canberra
Perth


Subscription Location

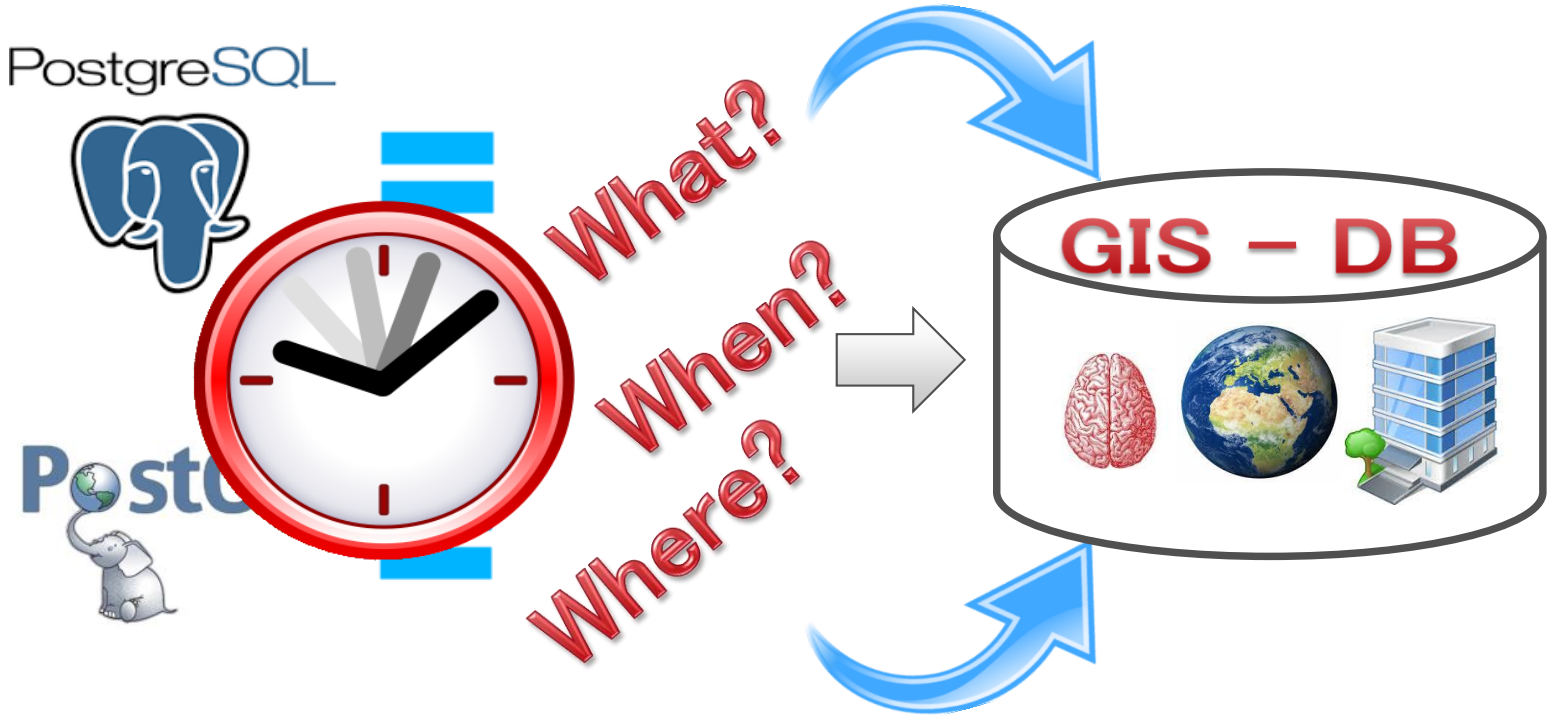
What is a Geographical Information System?



- The key feature of the GIS system is to organise lots of data into one or a few simple geographic view(s)
- Geographic views are generally built by putting together a set of what are referred to as map layers

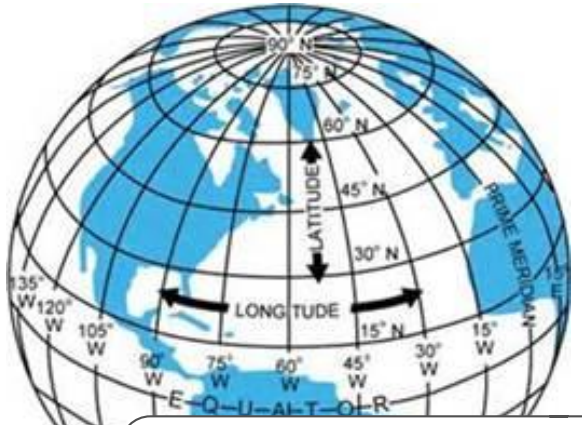
Why a geospatial database ?

- PostgreSQL is a relational database; doesn't know anything about storing geometry
- PostGIS is an extension that adds powerful geometric storage, search and manipulation



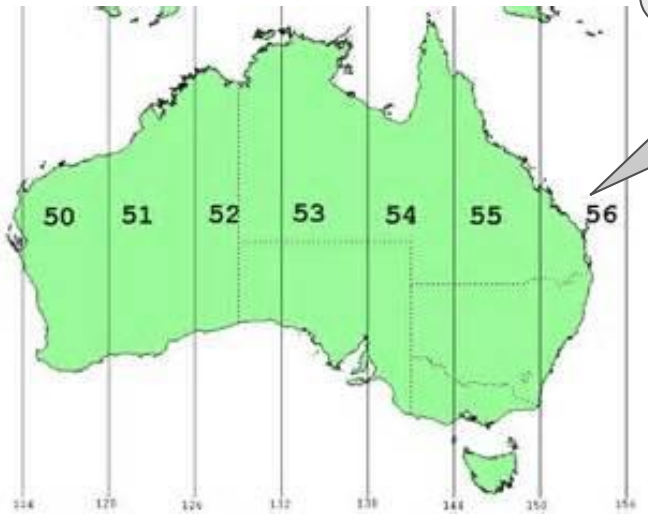
Co-ordinate systems

- Geographic Co-ordinate system (spherical based)
  - Eg: WGS1984



Brisbane 27.4698° South
153.0251° East

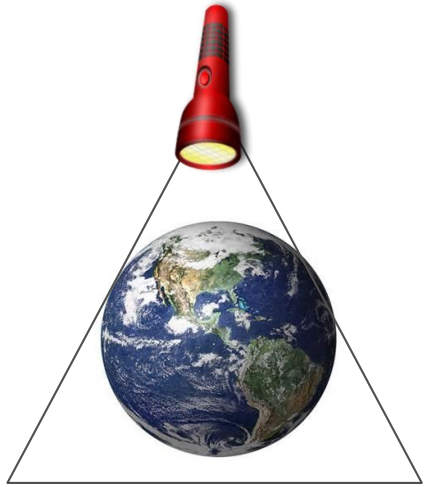
Brisbane in UTM is 56J
6861256N 502648E

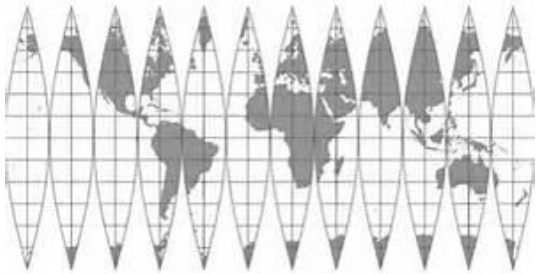


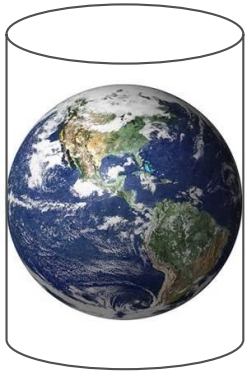
- Projected Co-ordinate system (two dimensional based)
  - Eg: Universal Transverse Mercator (UTM)

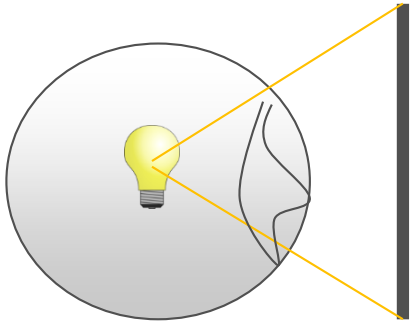
Projections

- Different projections can represent the object (earth) differently.









- European Petroleum Survey Group (EPSG) created the most commonly used SRID codes.
  - Must know the SRID for the geometry data to be accurate

Datum

- Point of reference for measurements
  - Many different datum for different geographic areas, and new more accurate ones replace older ones

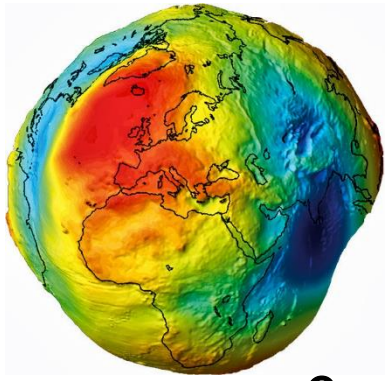
Is Everest the highest mountain in the world?

They are only models

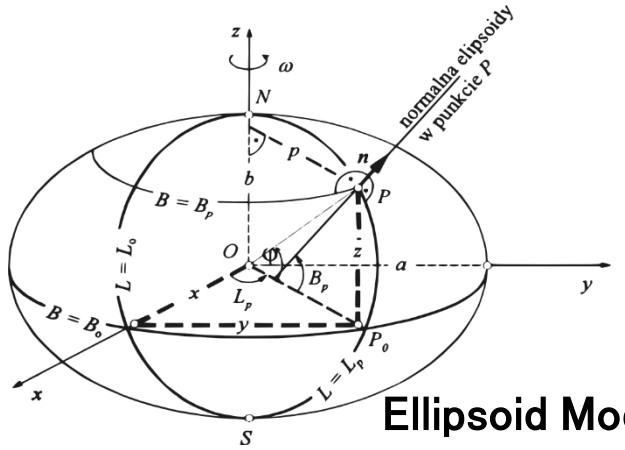
- Models are a simplification of the real world
  - Assist our understanding
  - Faster to process calculations

But …

- They have compromises, flaws and inaccuracies
- Always a compromise of speed with visible errors



Geoid earth



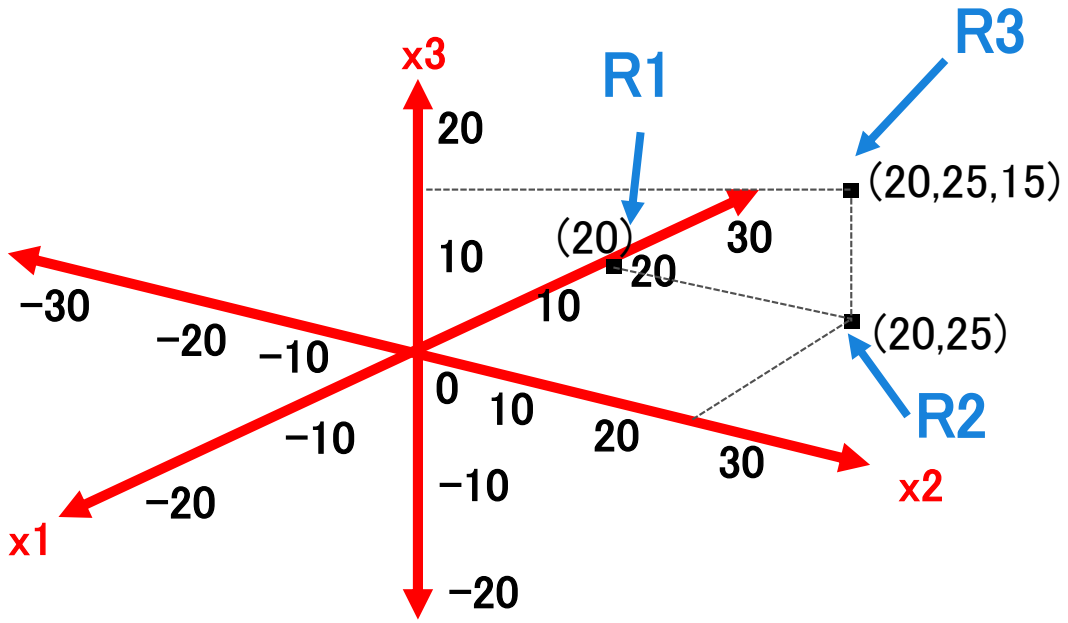
Ellipsoid Model of Earth

Not a great fit!

What will happen to distances and shapes?

GEOMETRY Type

- Represents a feature in the Euclidean coordinate system
  - R1 – one dimension (line)
  - R2 – two dimensions (plane)
  - R3 – three dimensions (space)



Smart City Singapore

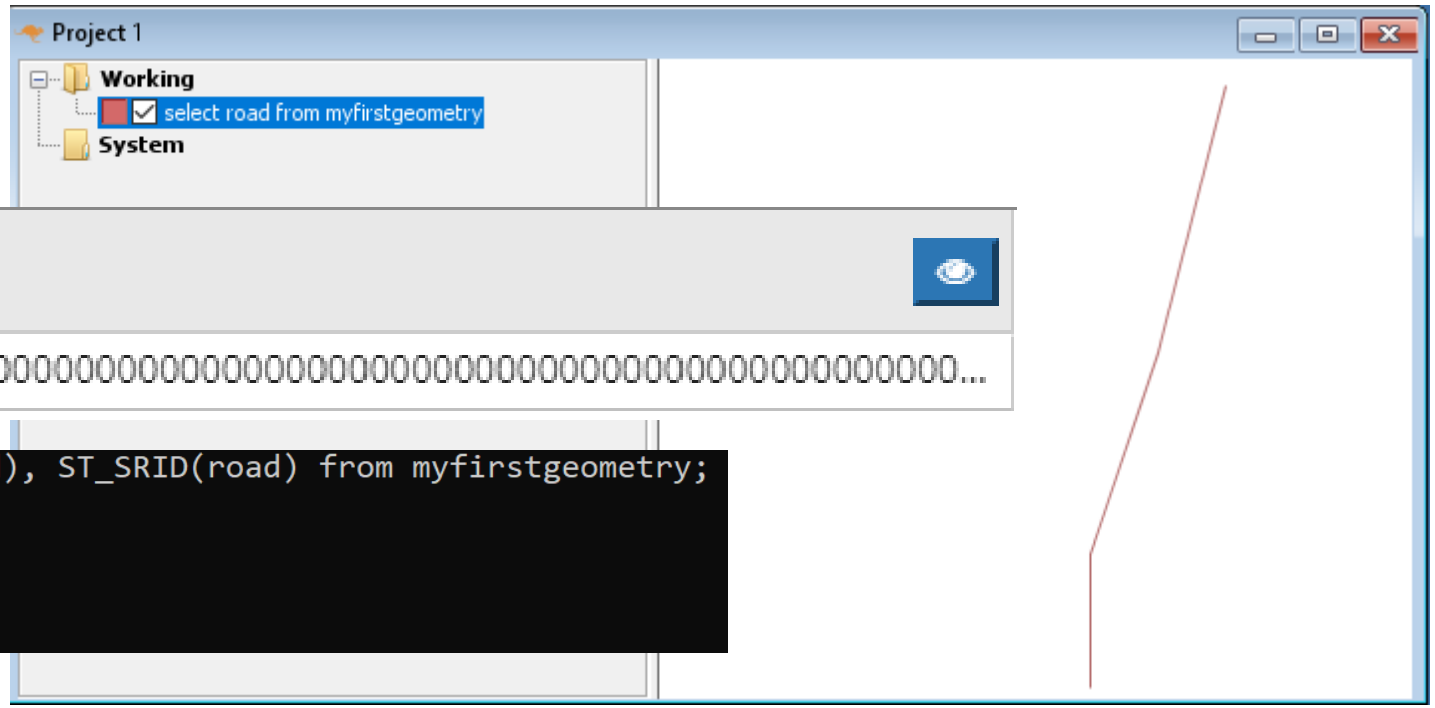
- Postgis stored objects and functions

PostGIS Types

- Geometry

```
Create table myfirstgeometry (id integer, name text, road geometry);

Insert into myfirstgeometry values
     (1,'Smith', ST_GeomFromText('LINESTRING(0 0,0 2,1 5, 2 9)'));
```



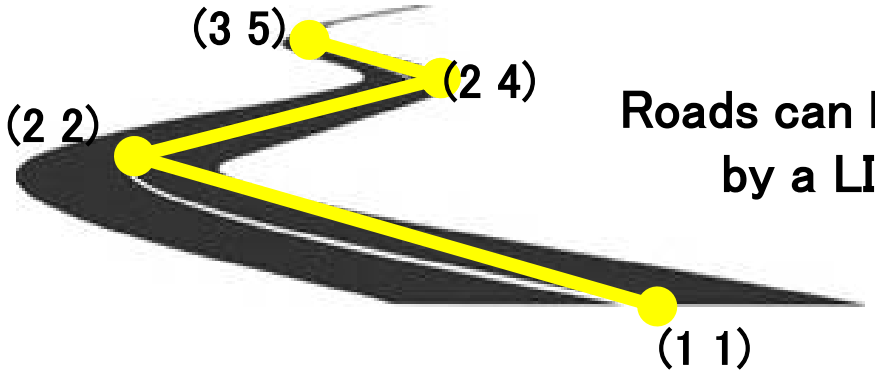
	id integer	name text	road geometry
1	1	Smith	0102000000040000000000000000000000000000000000000000000000000000000000000000000000...

```
pgdu=# SELECT name, ST_GeometryType(road), ST_NDims(road), ST_SRID(road) from myfirstgeometry;
 name  | st_geometrytype | st_ndims | st_srid
-------+-----------------+----------+---------
 Smith | ST_LineString   |        2 |       0
(1 row)
```

Well Known Text (WKT)

- Point(7 8)
  - A point identify a location in a coordinate space

- LINESTRING(1 1,2 2,2 4,3 5)
  - A path between locations in a coordinate space



Well Known Text (WKT)

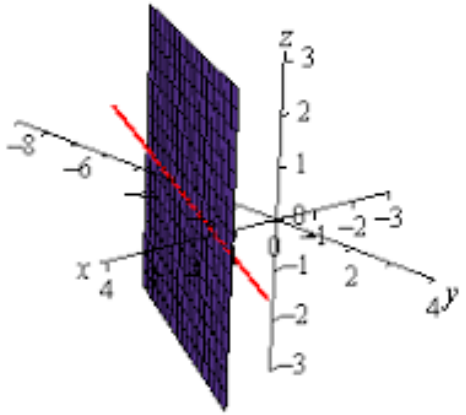
- POLYGON(1 1,2 2,2 4,3 5)
  - An area specified by a number of points

Exterior ring (no interior)
POLYGON(1 1,2 2,2 4,3 5,･･･,7 8, 9 12)

Exterior and interior rings
POLYGON((1 1,2 2,2 4,3 5,･･･,7 8, 9 12),
(3 2,4 4,5 4,･･･,4 6))

3 Dimensions

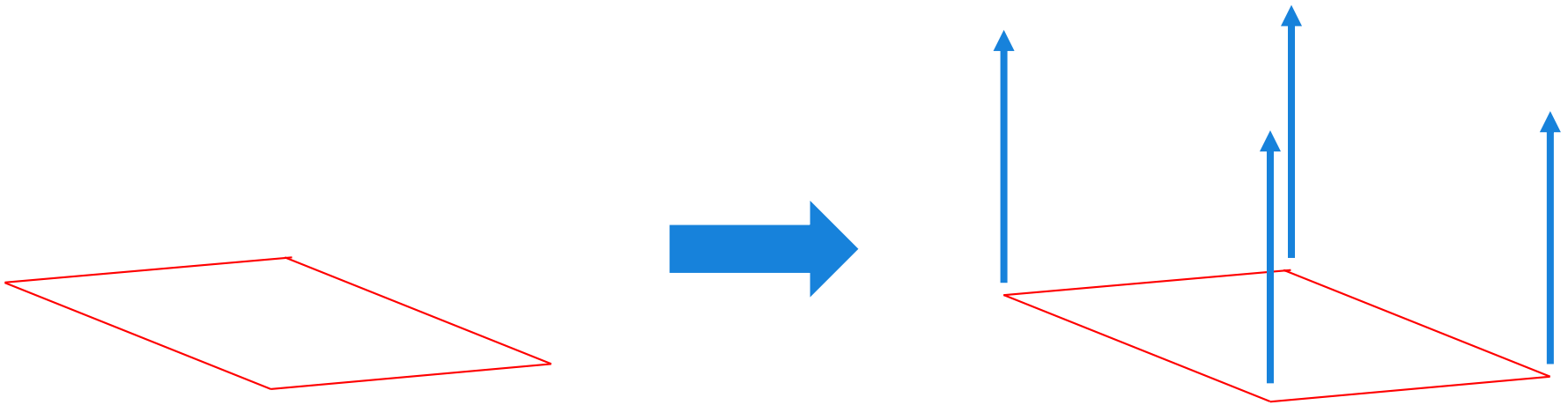
- PostGIS geometry supports three coordinate dimensions, but not volumetric geometry



- PostGIS geometry also supports polyhedral surfaces, Which are lots of 2 dimensional geometry surfaces linked together in a three dimensional co ordinate system (again not volumetric)

3 Dimensions

- PostGIS geometry has x, y, z and the M value



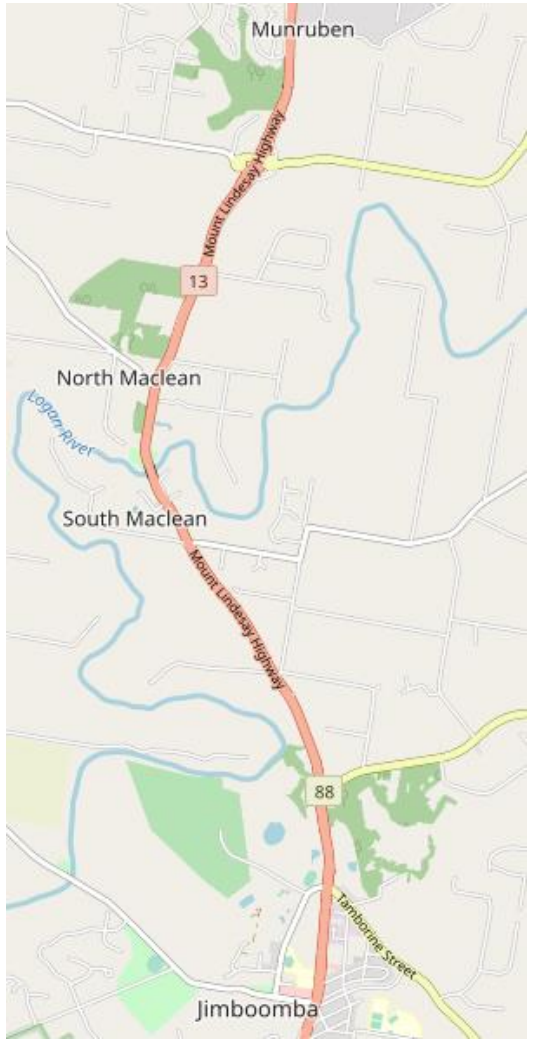
2D shape

With height

Measuring Distance

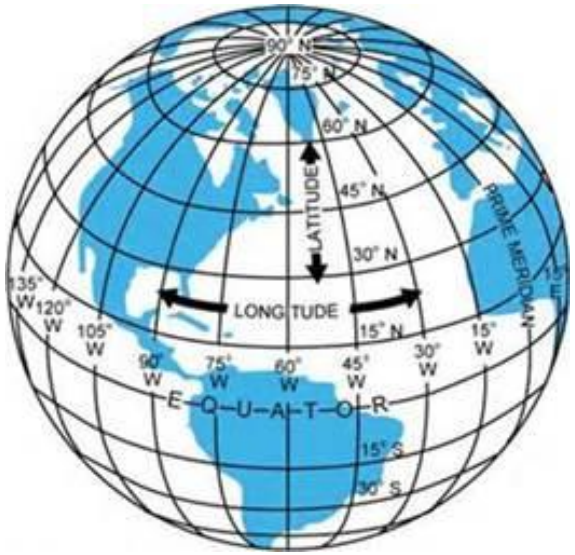
- How many motorcycle club members are within 15kms of Mount Lindesay Highway

```
SELECT count(distinct m.id) as total
 FROM member m
  INNER JOIN planet_osm_roads as r
  ON ST_Dwithin(m.location, r.way, 1000 * 15)
 WHERE r.name = 'Mount Lindesay Highway';
```



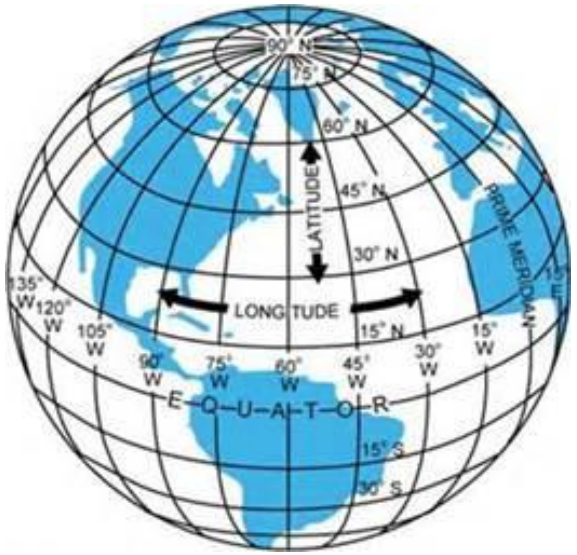
GEOGRAPHY Type

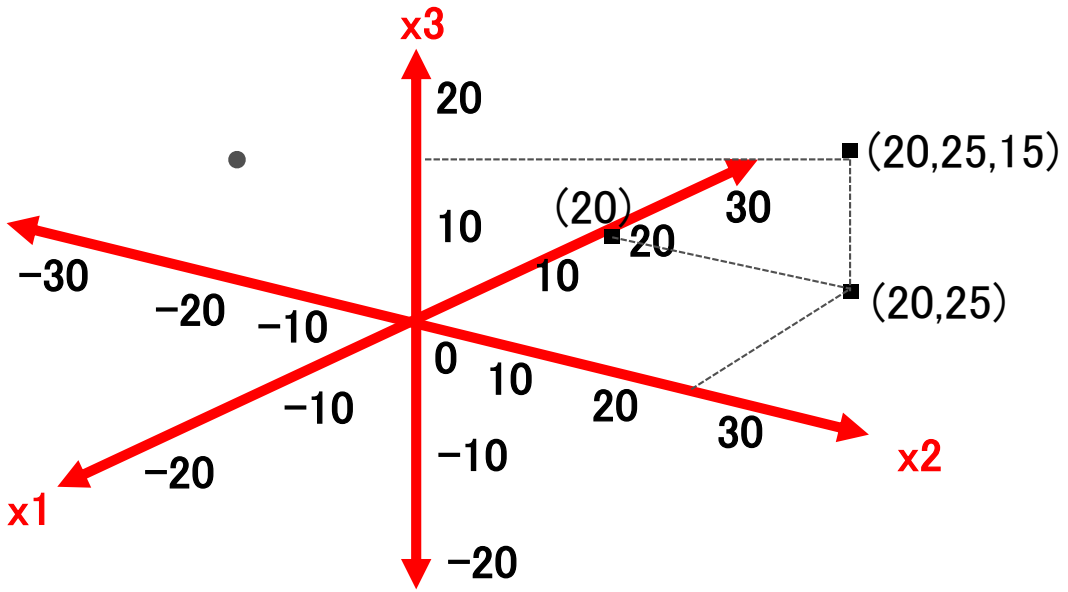
- Represents a feature in a spherical/Ellipsoid co-ordinate system
  - Measurements work off of Datum's which vary depending on the co-ordinate system used



GEOGRAPHY versus GEOMETRY

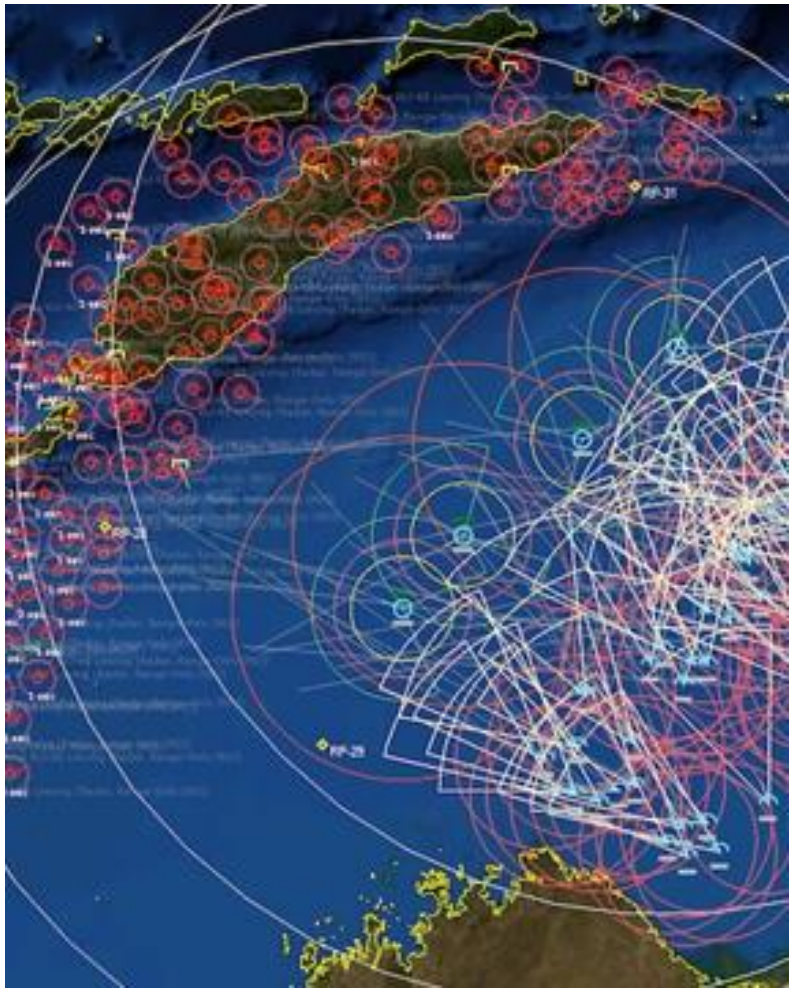
- Both could be used to represent a geographical location on our earth
  - Functions that work on geography types tend to return values that we are familiar with (areas in square kilometres, distances in kilometres). Geometry types would return square degrees.
  - Geometry has a richer set of functionality at present





Useful applications for PostGIS (Geography)

- Navigation



Measuring

Sydney
151.2093 East
−33.8688 South

Singapore
103.8198 East
1.3521 North

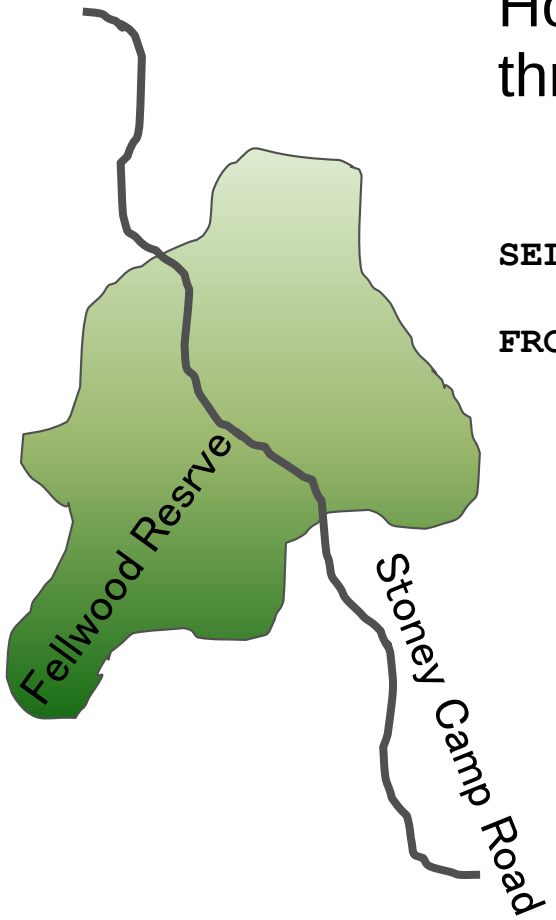
```
st_length(st_geogfromtext('linestring(103.8198 1.3521,151.2093 -33.8688)'));

st_length_spheroid(st_geomfromtext('linestring(103.8198 1.3521,151.2093 -33.8688)')
                  ,'SPHEROID["GRS_1980",6378137,298.257222101]') ;

st_distance(st_geogfromtext('point(103.8198 1.3521)')
                  ,st_geogfromtext('point(151.2093 -33.8688)')));
```

Intersections

- ST_Intersects – Returns true if one geometry intersects another
- ST_Intersection – Returns the geometry of the intersected area

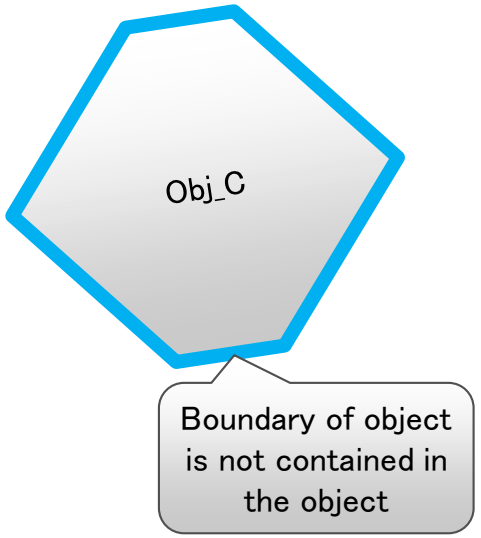


How much of Stoney Camp Road goes through Fellwood Reserve?

```
SELECT
    ST_Intersection(g.reserve,g.road) AS intersection
FROM
    (   SELECT way FROM reserves
            WHERE name = 'Fellwood Reserve') AS reserve,
         SELECT way FROM roads
            WHERE name = 'Stoney Camp Road') AS road
     ) AS g;
```

Contains and Within

- ST_Contains(Obj_A.geom, Obj_B.geom)
  - True When
    - Obj_B has no points in the **exterior** of Obj_A
    - Obj_B must have at least one point in the **interior** of Obj_A
  - ST_Contains(Obj_C,ST_Boundary(Obj_C)) = False

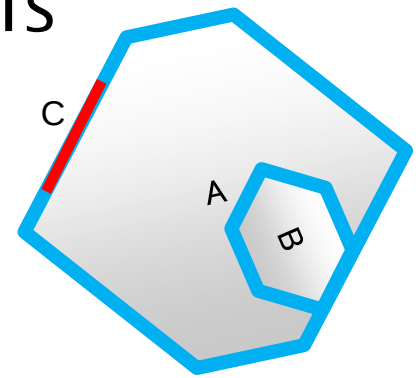


- ST_Within(Obj_A.geom, Obj_B.geom)
  - Inverse of ST_Contains
  - ST_Within(A,B) = ST_Contains(B,A)

Covers and CoveredBy

- ST_COVERS
  - Similar to ST_Contains except that
    - Obj_B has no points in the exterior of Obj_A
    - Obj_B must have at least one point in the interior **or boundary** of Obj_A
  - Not a OGC standard

- ST_CoveredBy – inverse of ST_Covers



ST_Contains(A,B)= True

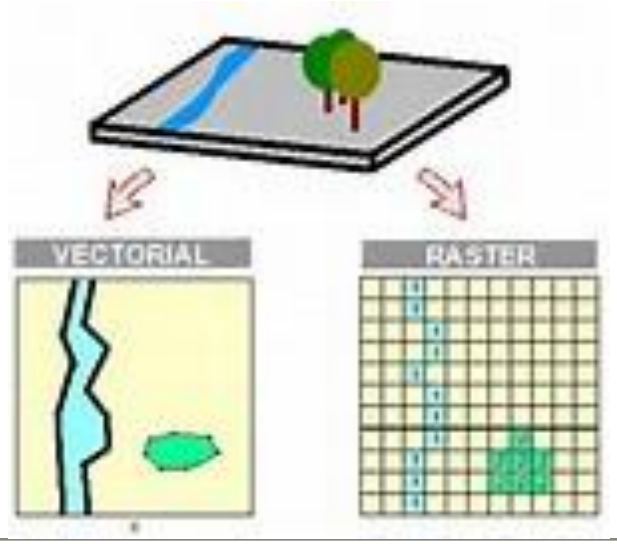
ST_Covers(A,B) = True

ST_Contains(A,C)= False ⇐

ST_Covers(A,C) = True ⇐

Raster Types

- Raster data is similar to a bunch of different coloured cells (like pixels on your computer screen)
- Each cell is a store for numeric data, and can store more than just colours
- Cells don't need to be the same shape, but we try to keep them that way for more efficient processing

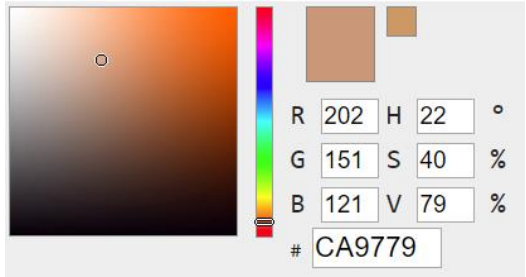


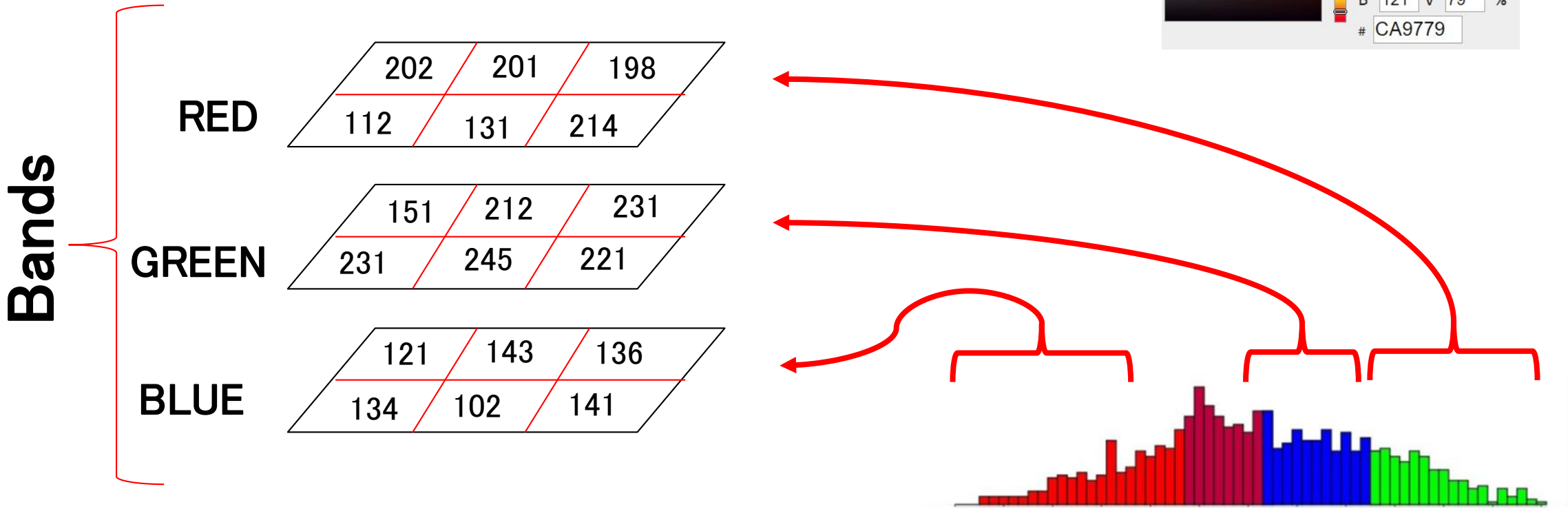
Raster Types

- Each bit of data that a cell can store is referred to as a band or channel
- Rather than trying to store one very large raster, it can be broken into tiles with cells arranged into rows and columns
- Typically satellite images are raster
- Often used as a base from which vector objects can be created

Raster Type Properties

- Width and Height - measured in cells
- Bands – at least one and can have hundreds



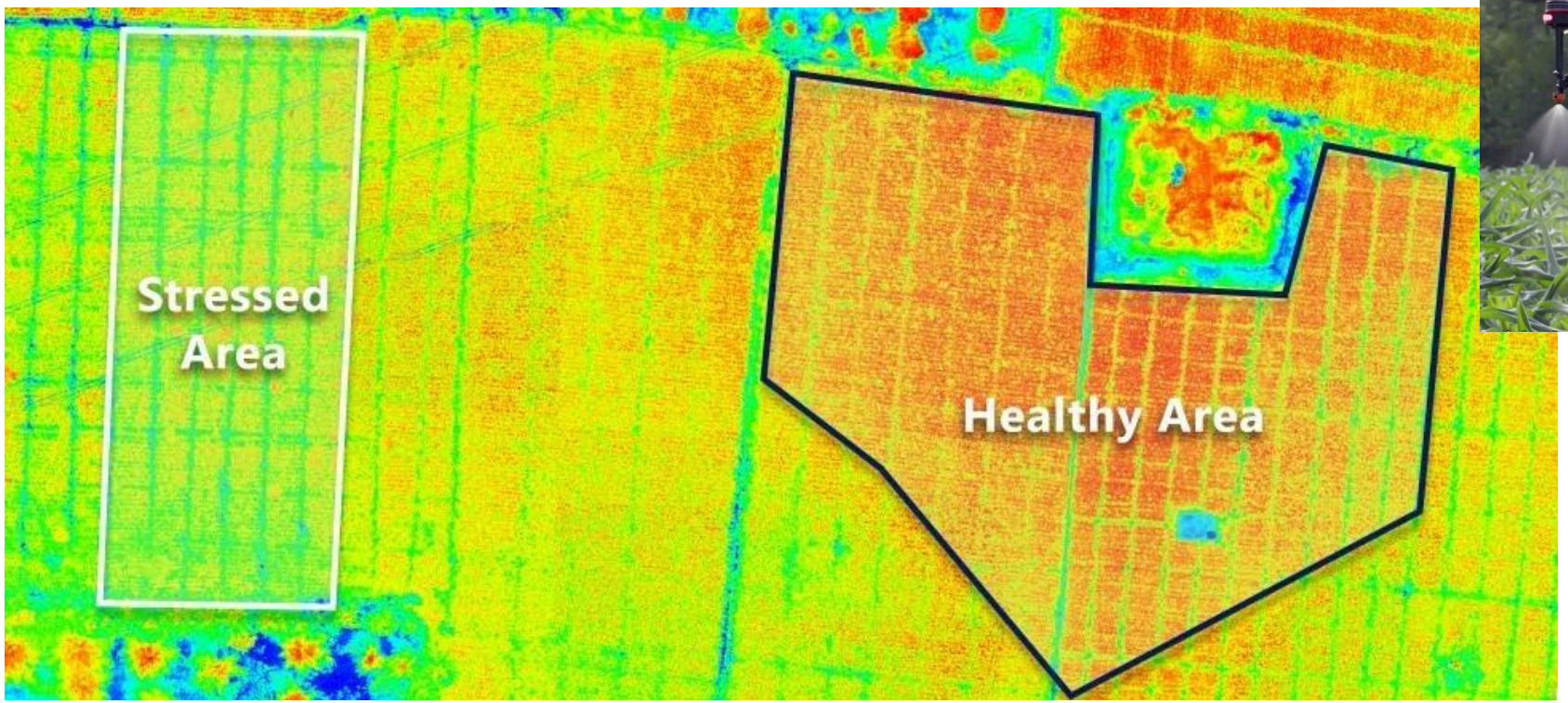


Raster Type Properties

- Width and Height - measured in cells
- Bands – at least one and can have hundreds
- Spatial Reference System – an SRID gives the reference system
- Cell Width and Height – relates to real life measurement
  - eg 5 meters wide by 10 meters high
- Cell Scale – required to identify the location of a particular cell relative to spatial co-ordinates
  - In the above example of 5M X 10M the scale would be x=1:5 and y=1:10 (normally measured from top left corner of a tile
- Skew – 0 means no skew, but if one is supplied it refers to the angle that the raster is rotated to

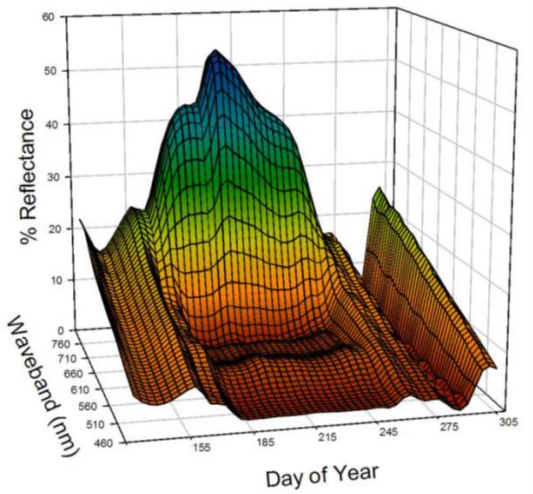
Useful applications for PostGIS (RASTER USAGE)

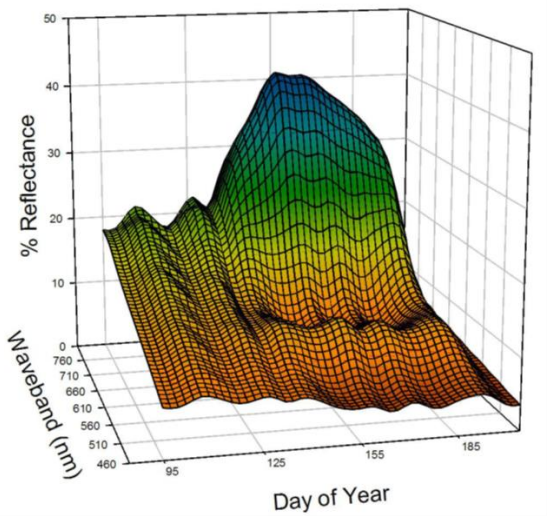
- Precision agriculture

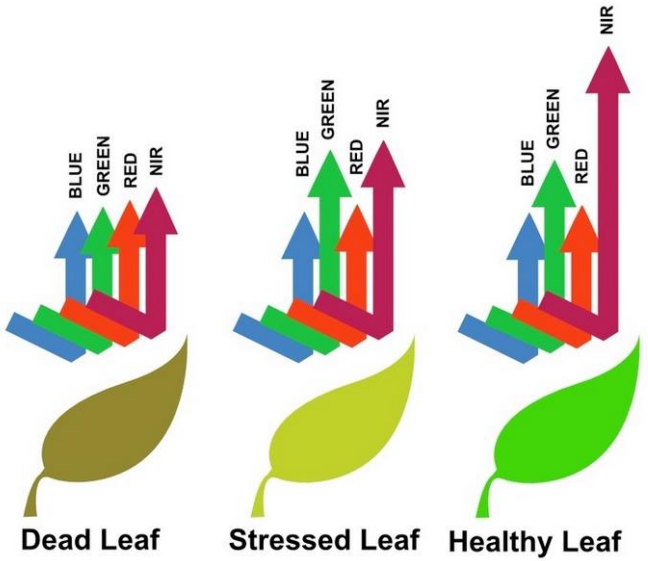


Useful applications for PostGIS (RASTER USAGE)

- Drone flight path management
- Spectral (RGB) image analysis

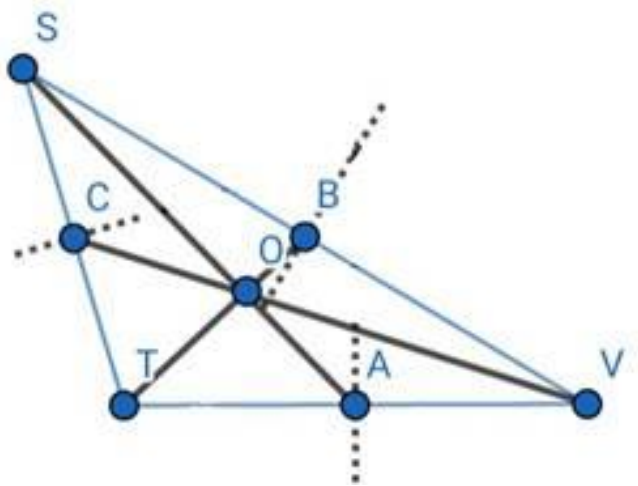


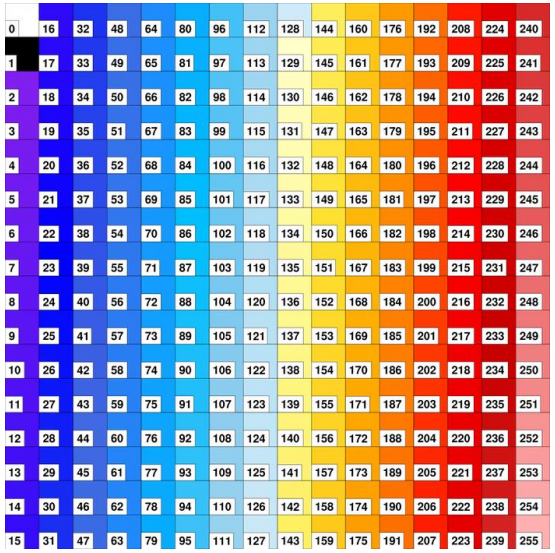


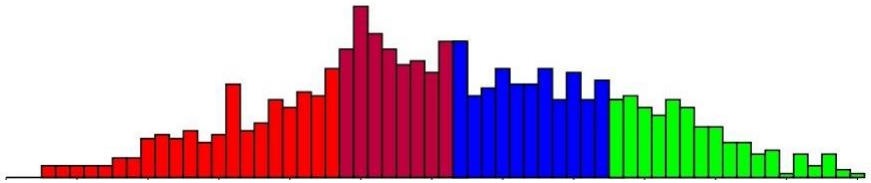


Useful PostGIS Functions

- ST_MapAlgebra
- ST_Colormap
- ST_PixelAsPoint
- ST_Contains
- ST_Histogram
- ST_AsJPEG / ST_asTIFF





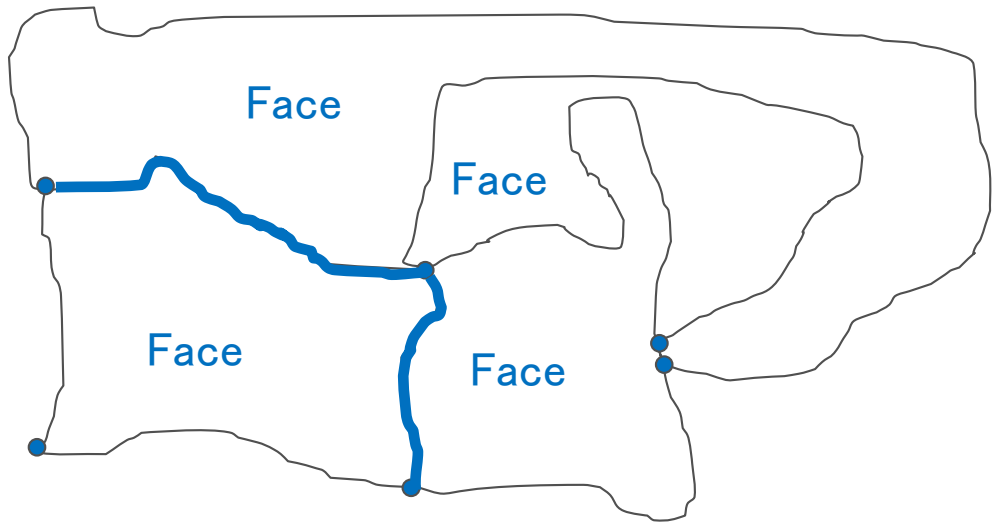


Topology

- Topology
  - Topology types maintain the relationship between spatial features
  - Understands the impact of modifying a spatial object
  - Became a significant component of PostGIS in version 2.0
  - Installed as an extension
    - Create extension postgis_topology

Topology

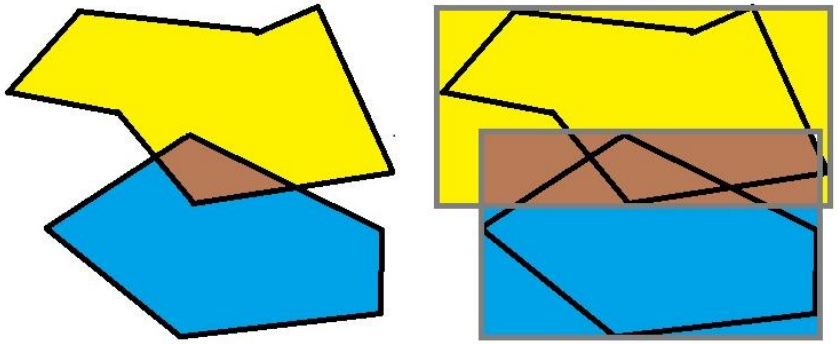
- Every intersection is a node
- Edges are shared (not separate entities like with a geometry polygon)
- Faces share nodes and edges



PostGIS Indexes

Spatial Indexes

- Generalised Search Tree (GIST) - need to specify as PostgreSQL implements extended indexes that allow a B-Tree to be created on any type for example



Object Overlap
(Slow)

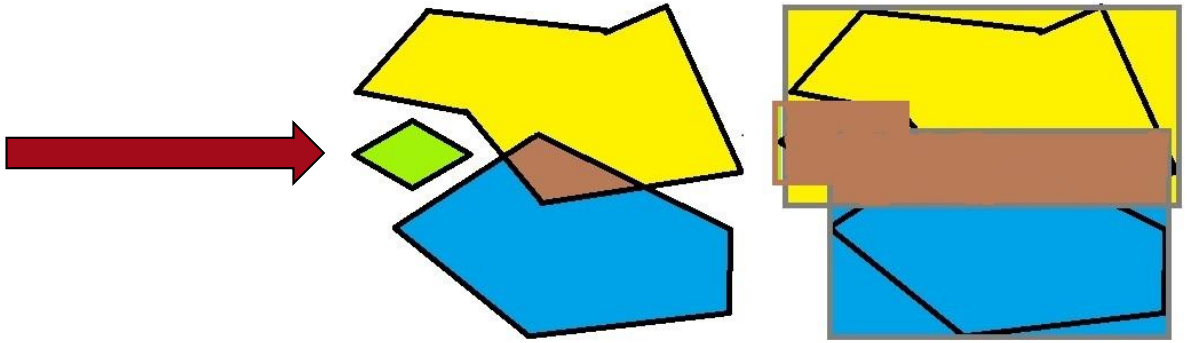
Bounding Box Overlap
(fast)

If we use the default B-Tree

If we use the GIST

Spatial Database

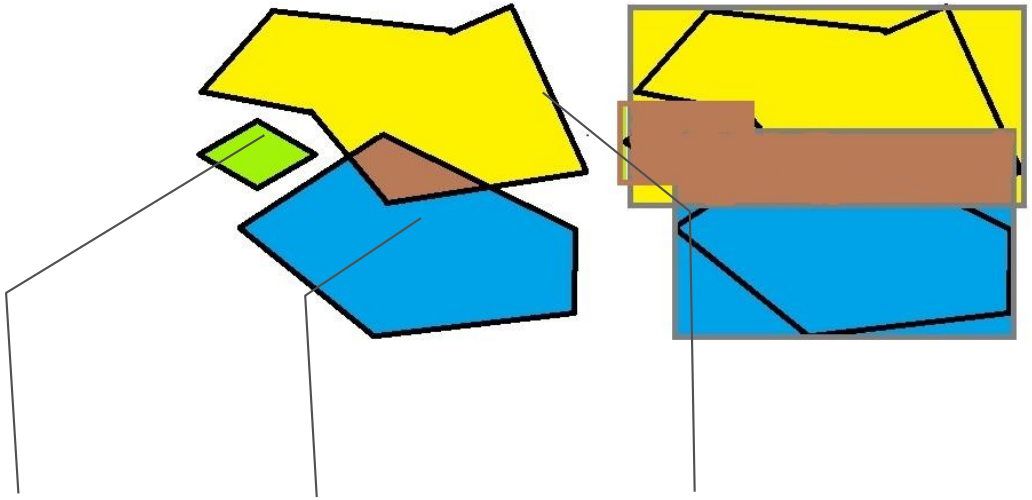
- **Spatial Indexes**
  - Generalised Search Tree (GIST)



Spatial Database

Spatial Indexes

- Generalised Search Tree (GIST)



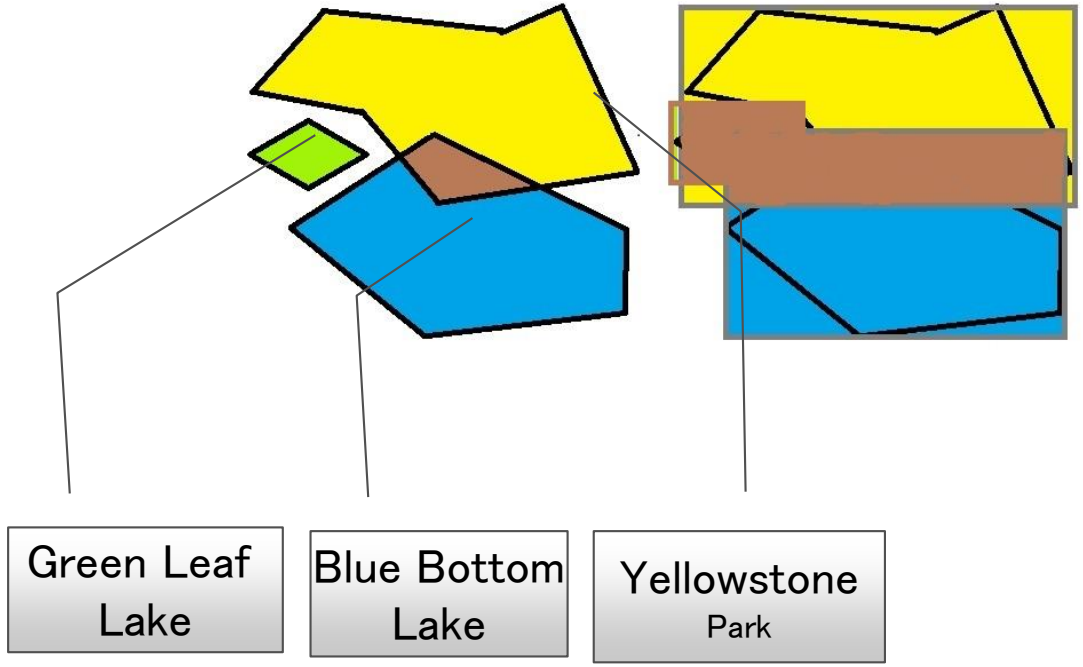

```
SELECT l.name
FROM lake l, park p
WHERE l.geom && p.geom;
```

Returns both ‘Green Leaf Lake’ And ‘Blue Bottom Lake’

Spatial Database

Spatial Indexes

- Generalised Search Tree (GIST)




```
SELECT l.name
FROM lake l, park p
WHERE l.geom && p.geom;
```

Returns both ‘Green Leaf Lake’ And ‘Blue Bottom Lake’

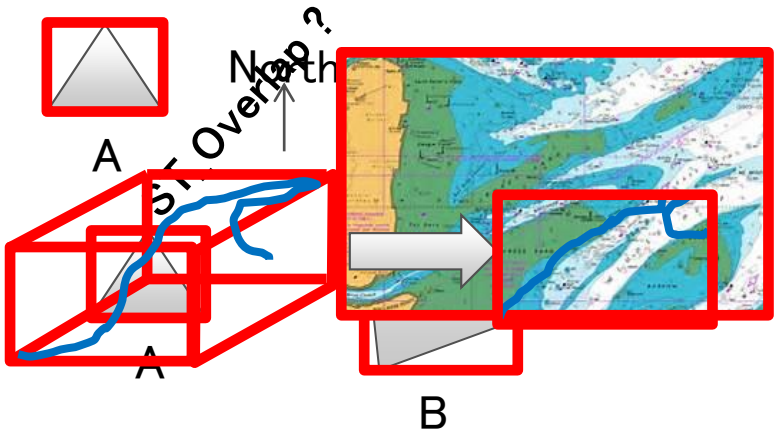
```
SELECT l.name
FROM lake l, park p
WHERE ST_Overlaps(l.geom,p.geom);
```

Returns only ‘Green ‘Blue Bottom Lake’

Bounding Boxes and Comparisons

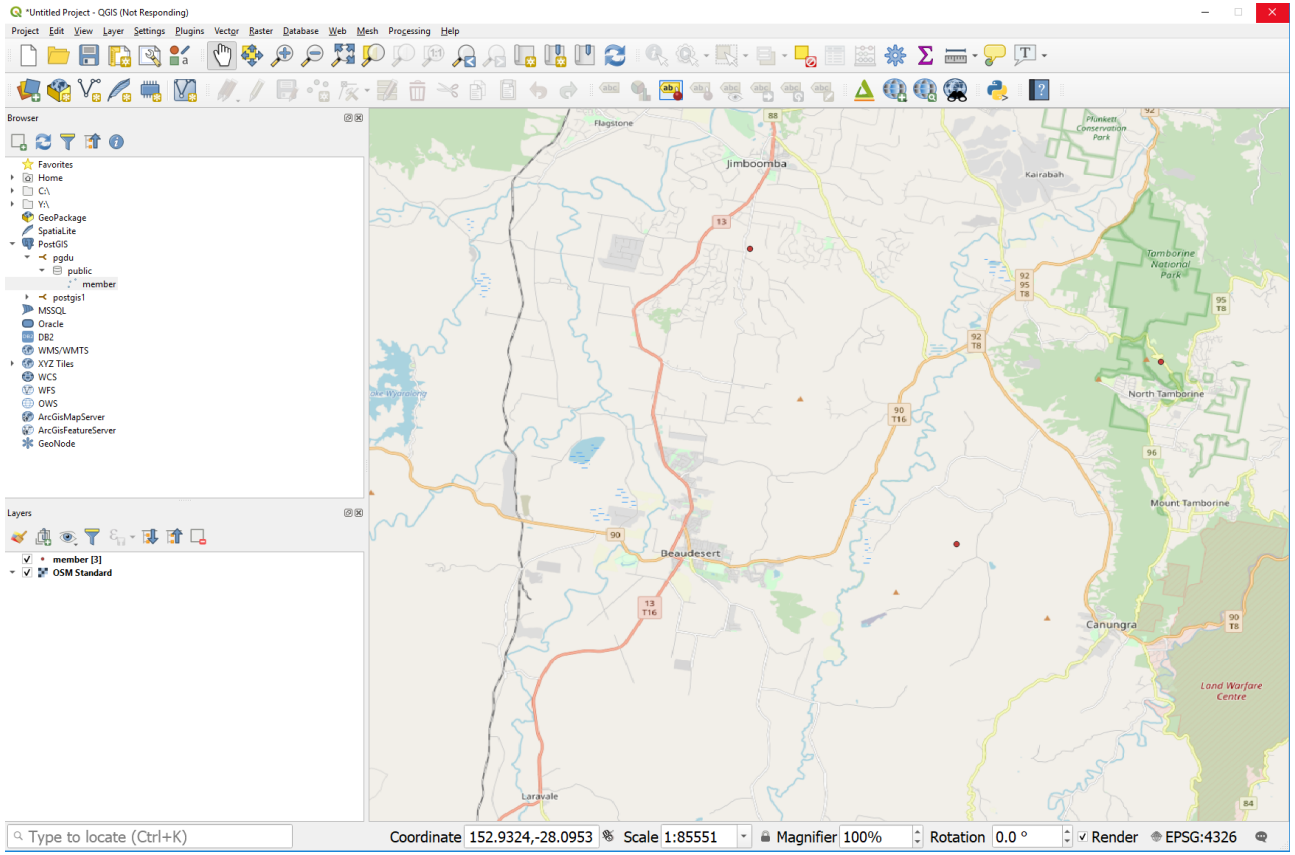
GIST indexes

- All geometries have a bounding box
  - Geometry (vector) objects
  - Geography (vector) objects
  - Raster objects
- Can easily and quickly be compared
  - Is bounding box of geometry **A** West of bounding box of geometry **B** ?
  - Raster bonding boxes can be compared with geometry vector bounding boxes
  - Geography bounding boxes must be converted to geometry bounding boxes before they can be compared to Geometry or raster types

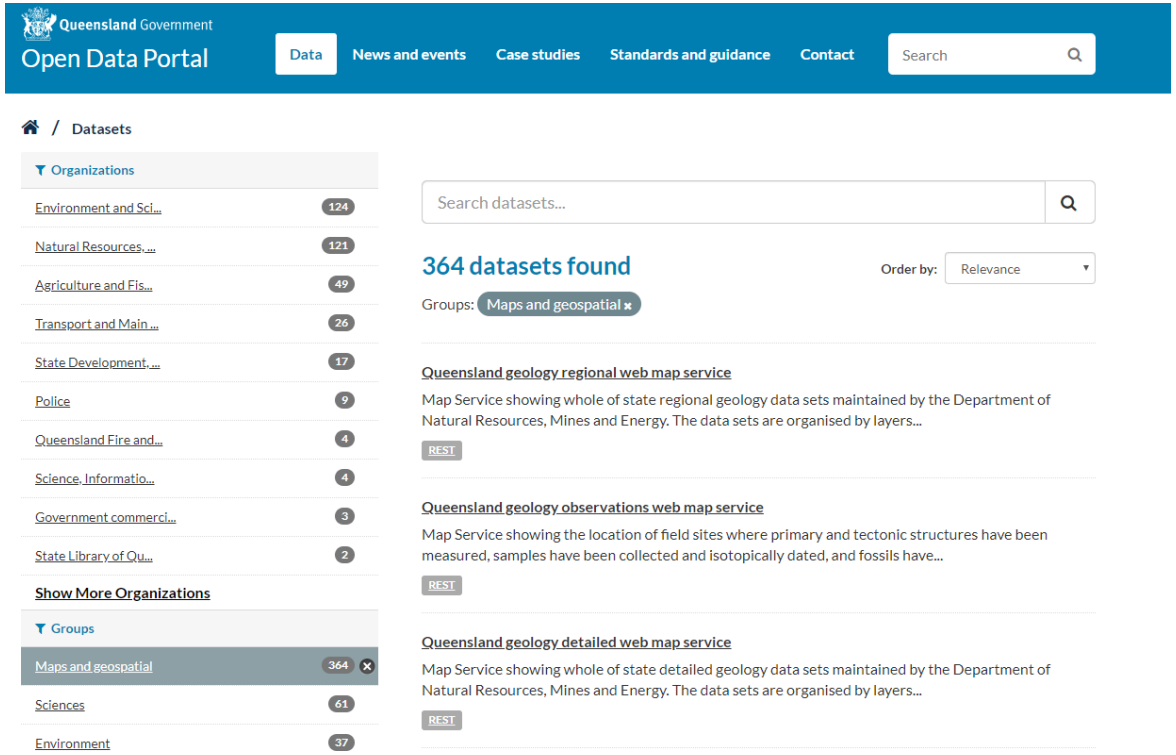


Visualisation Tools

- QGIS
- OpenJump



Loading Data




```
shp2pgsql -I -s 4283 SHAPEFILE.shp DATATABLE |
psql -U DATABASE_USER -d DATABASE_NAME
```