Spatial Database

PostGIS

ORACLE SPATIAL

Microsoft SQL Server 2008

Widely Supported

AutoCAD
Map 3D

uDig

Corporations

Ball Aerospace
& Technologies Corp.

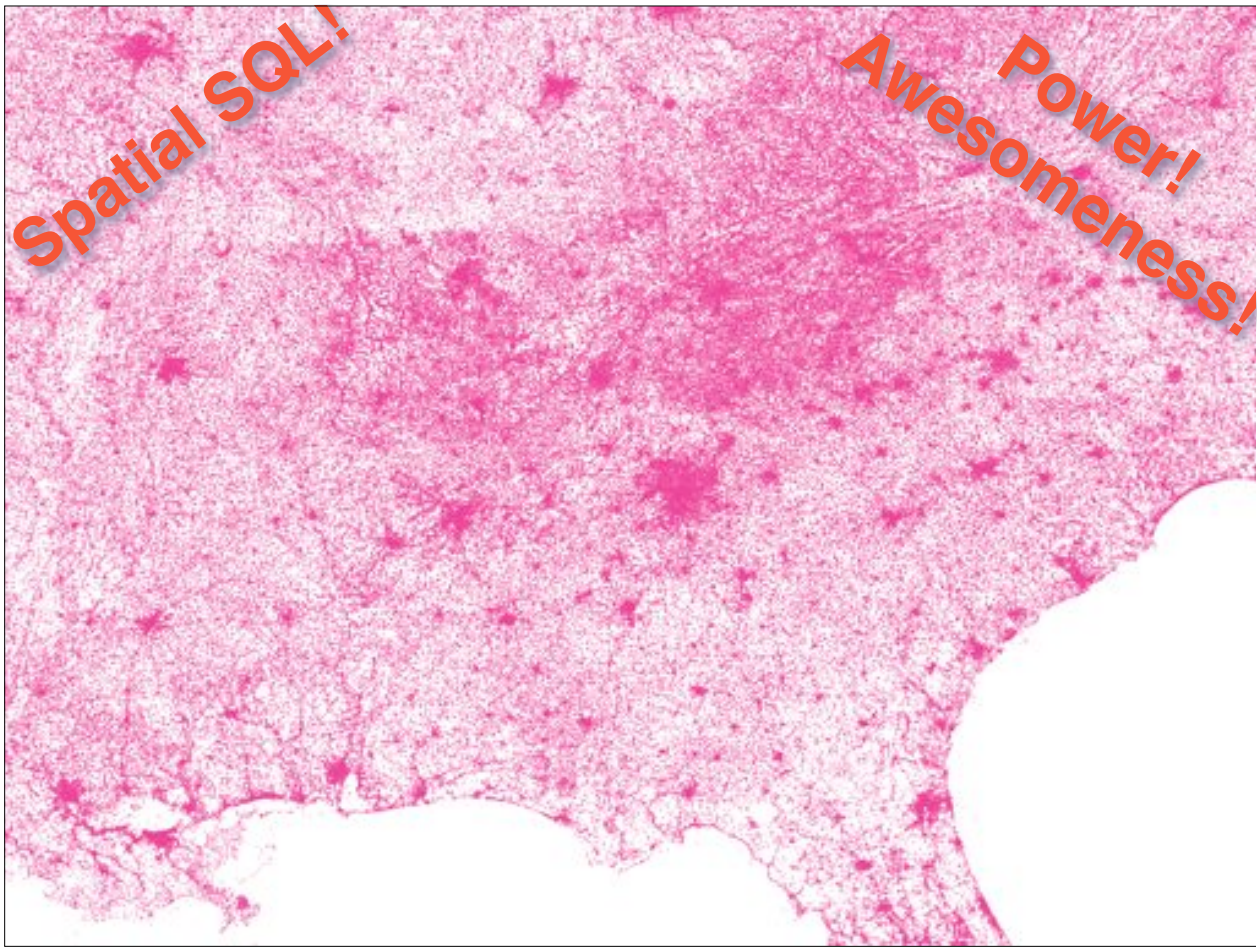
Governments

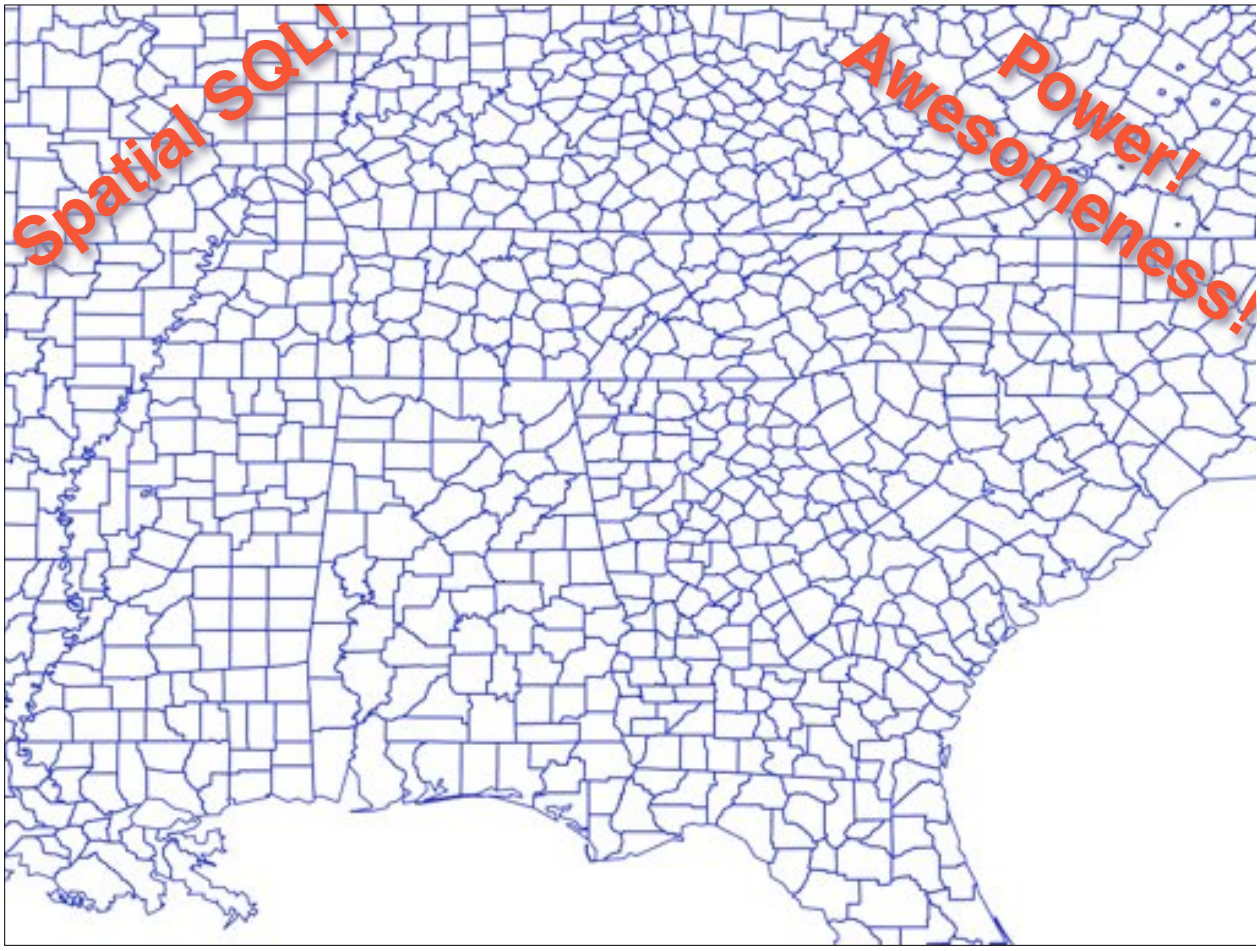
Freedom!

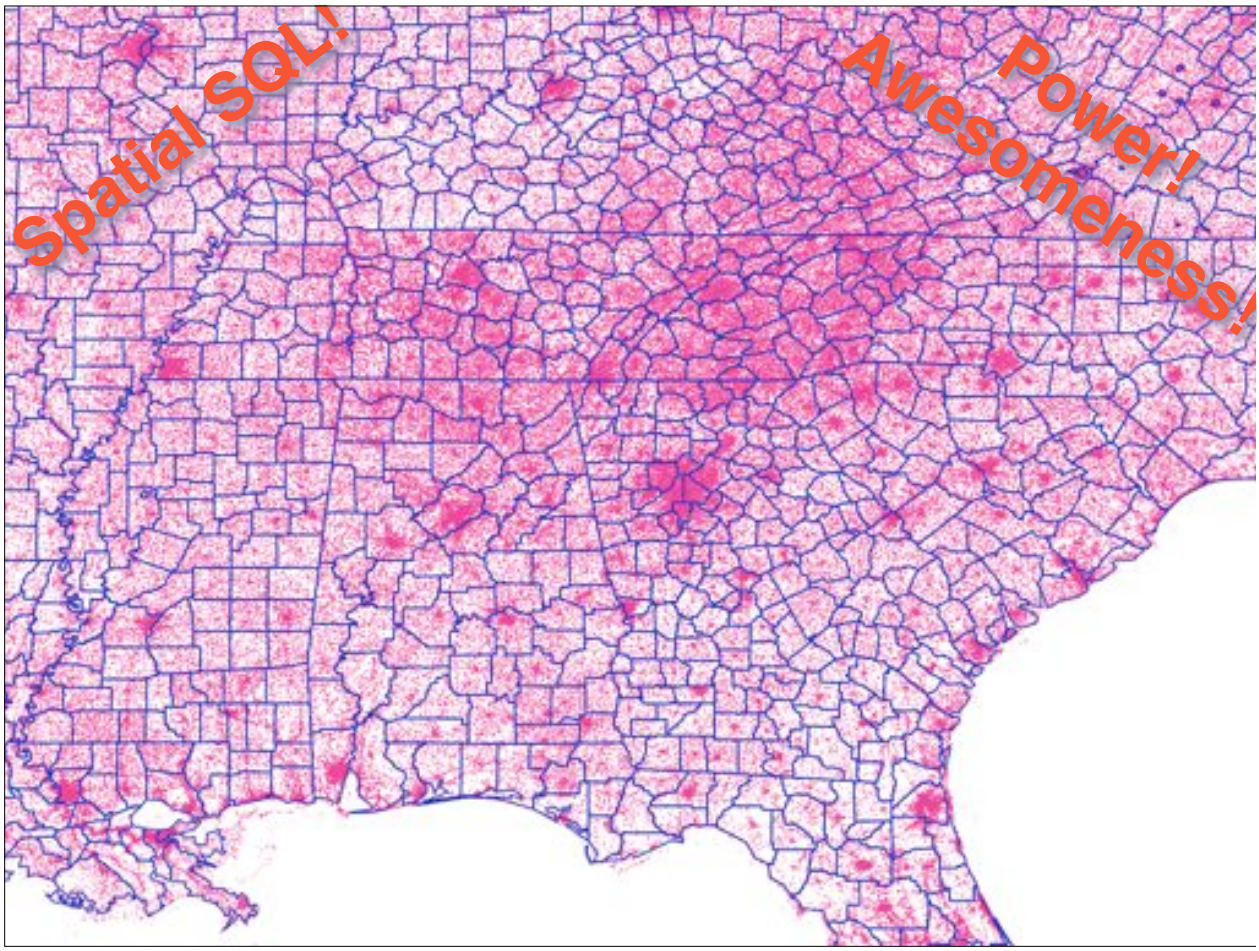
Liberty!

SQL Goodness!

NoSQL?


Spatial SQL!
Power!
Awesomeness!


Spatial SQL!
Power!
Awesomeness!



Spatial SQL!

Power!
Awesomeness!

```
SELECT
  census.*, customers.*
FROM census
JOIN customers
ON ST_Contains(
    census.geom,
    customers.geom
);
```

Costed, Planned Spatial Queries!

Try this on MySQL!

```
SELECT ...
FROM geotable_a a
JOIN geotable_b b
  ON ST_Intersects(b.geo,a.geo)
JOIN attrtable_c c
  ON (b.id = c.id)
JOIN attrtable_d d
  ON (a.id = d.id)
```

The Basics!

ST_Length(A)

ST_Distance(A, B)

ST_DWithin(A, B, r)

ST_Area(A)

ST_Intersects(A, B)

The Basics!

ST_AsText(A)

ST_AsBinary(A)

ST_GeomFromText()

ST_GeomFromBinary()

Fun Formats!

ST_AsGeoJSON()

ST_AsGML()

ST_AsKML()

ST_GeomFromGML()

ST_GeomFromKML()

Geometry Construction!

ST_Buffer()

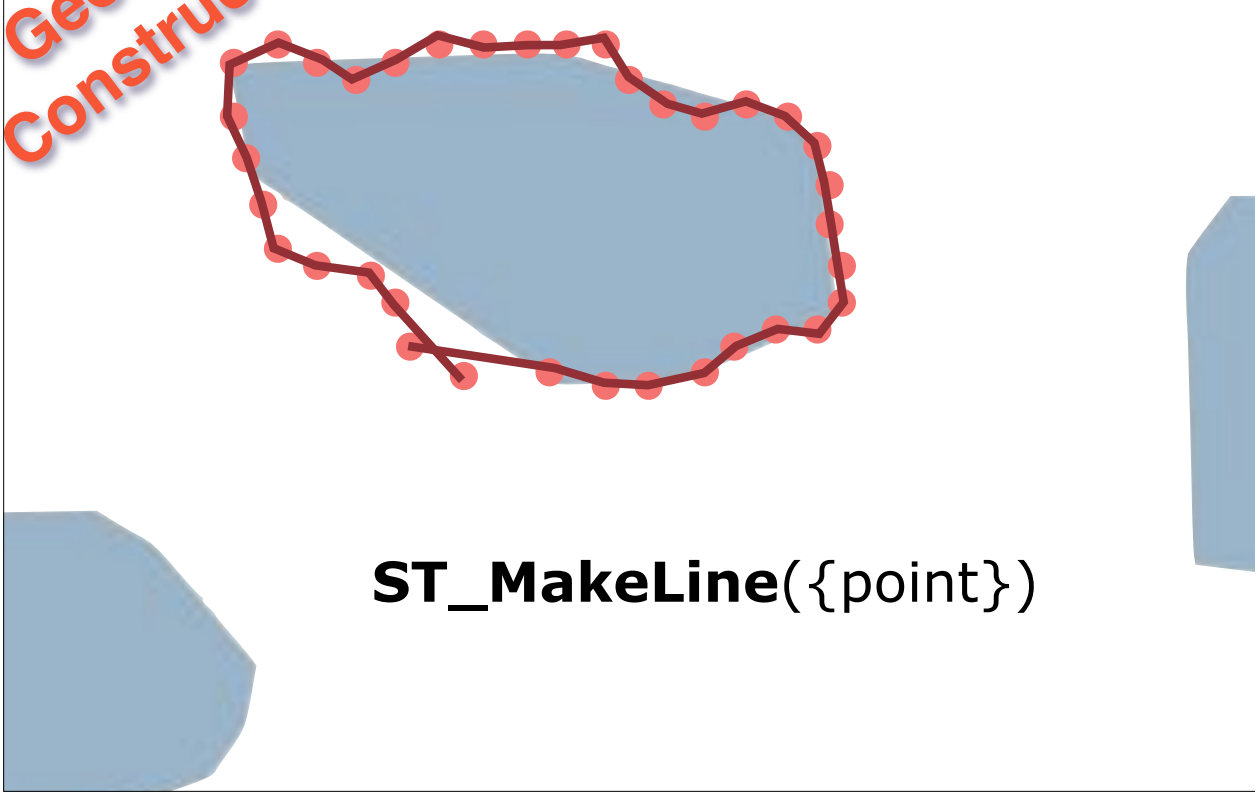
ST_MakeLine()

ST_Polygonize()

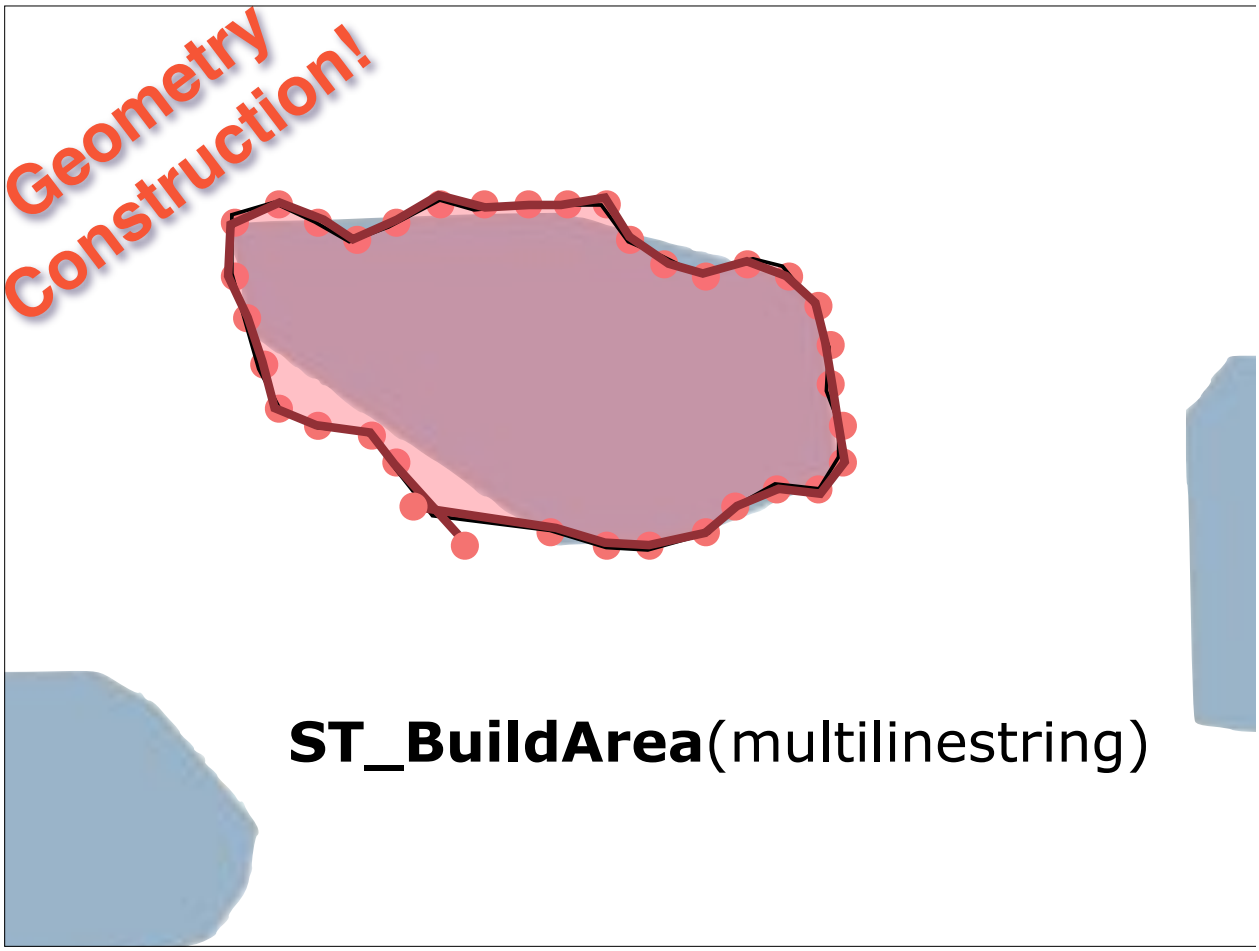
ST_BuildArea()

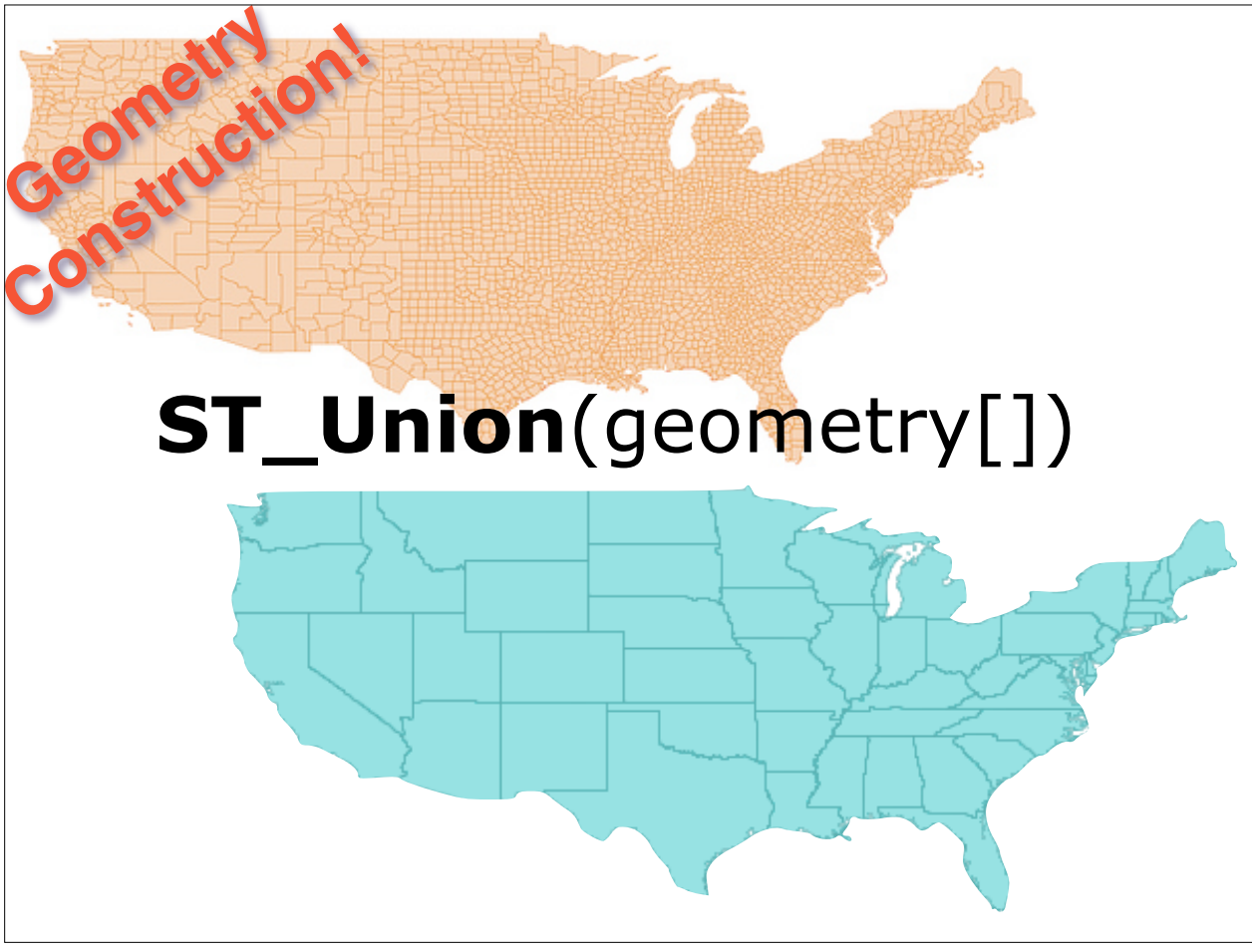
ST_Union()

Geometry Construction!



ST_MakeLine({point})


Geometry Construction!
ST_BuildArea(multilinestring)


Geometry Construction!
ST_Union(geometry[])

Cascaded Union!

Not just union!

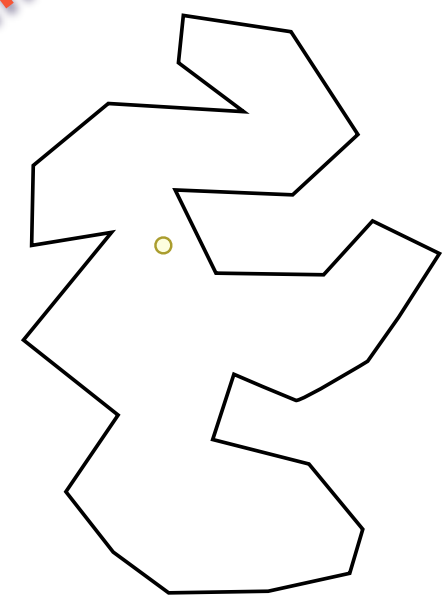
Cascaded Union!

Prepared Geometry!

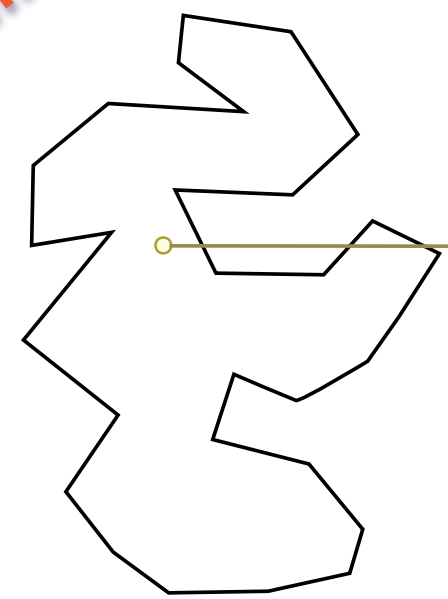
Not just Intersects!

```
SELECT ...
FROM points, polygons
WHERE ST_Intersects (
    polygons.geom,
    points.geom
  )
```

Prepared Geometry!

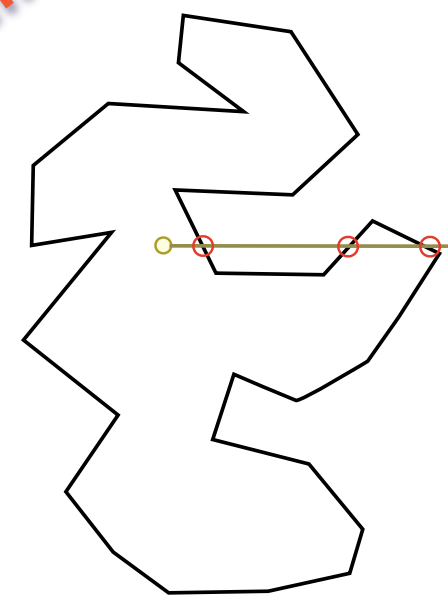


Prepared Geometry!



Point in Polygon

Prepared Geometry!



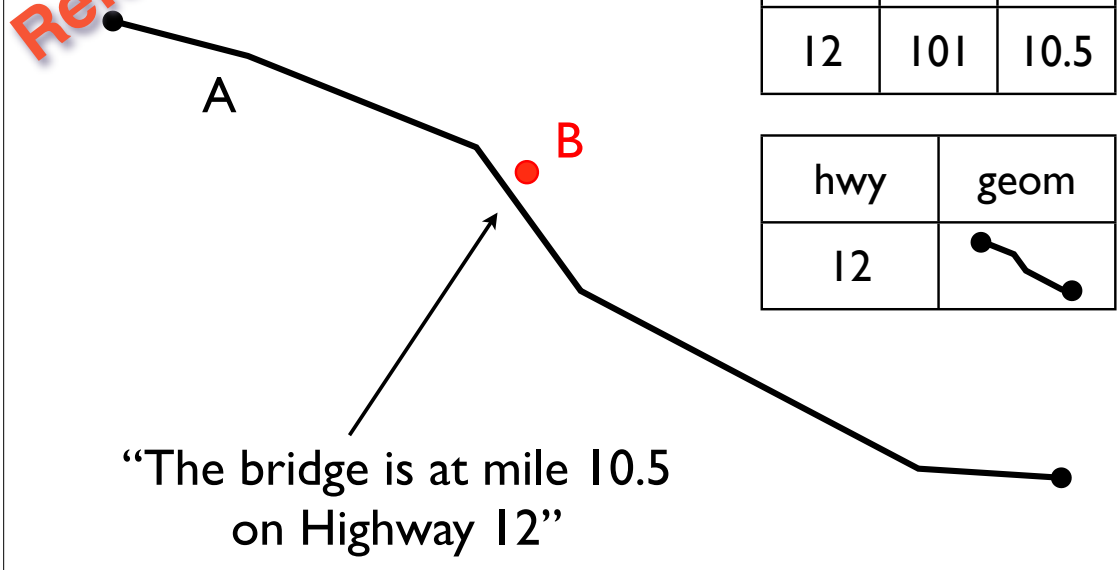
Point in Polygon = O(n)

Prepared Geometry!

Prepared geometry makes repeated tests on large geometries very fast.

(ST_Intersects, ST_Contains)

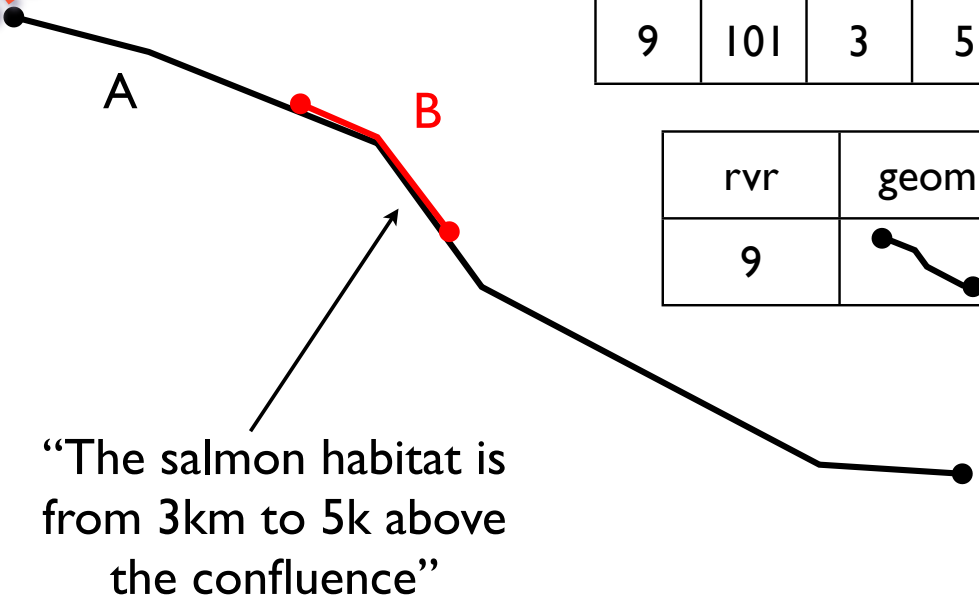
Linear Referencing!



hwy	brdg	loc
12	101	10.5

hwy	geom
12	

Linear Referencing!



rvr	fsh	from	to
9	101	3	5

rvr	geom
9	

"The salmon habitat is from 3km to 5k above the confluence"

Linear Referencing!

ST_LocateAlong()

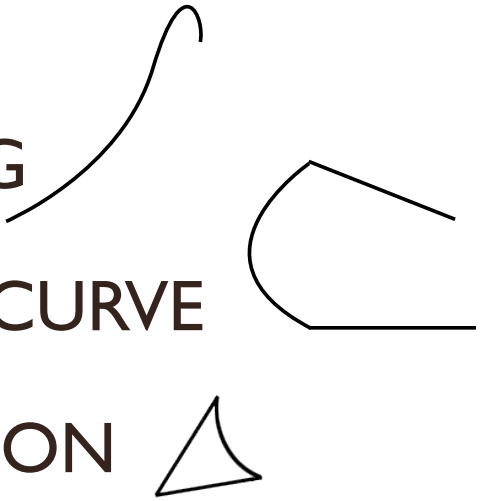
ST_LocateBetween()

ST_AddMeasure()

ST_Line_Locate_Point()

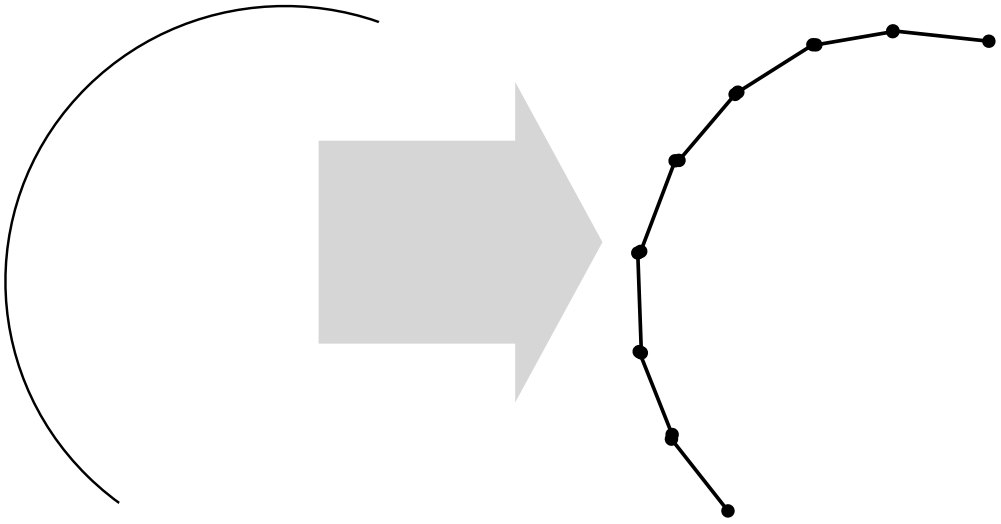
Curves!

- CURVESTRING
- COMPOUNDCURVE
- CURVEPOLYGON
- ST_CurveToLine()
- **ST_LineToCurve**()



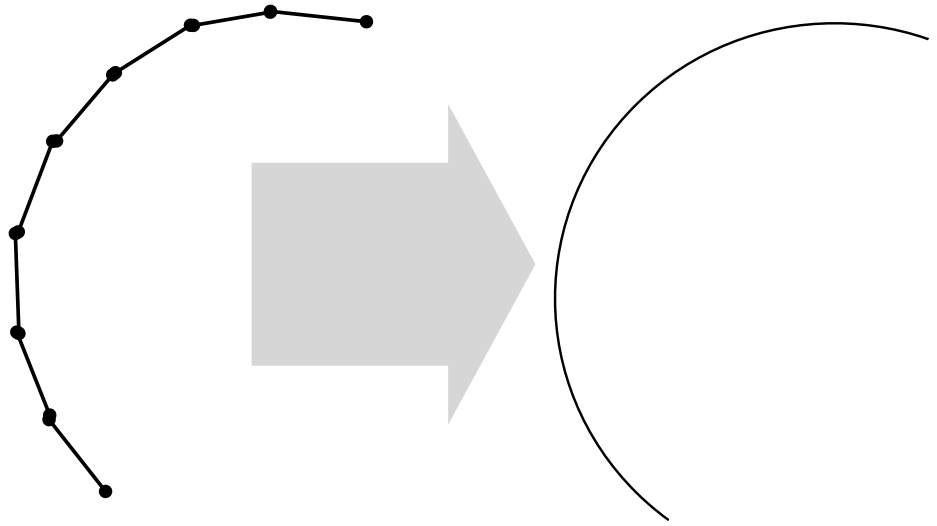
Curves!

ST_Curve ToLine()!



Curves!

ST_LineTo Curve()!



Reprojection

ST_Transform()

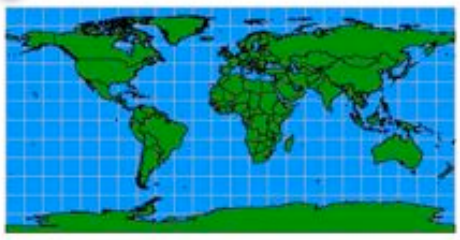


Plate Carree Projection

Sinusoidal Projection

Behrmann Projection

Albers Equal Area Conic Projection

Reprojection

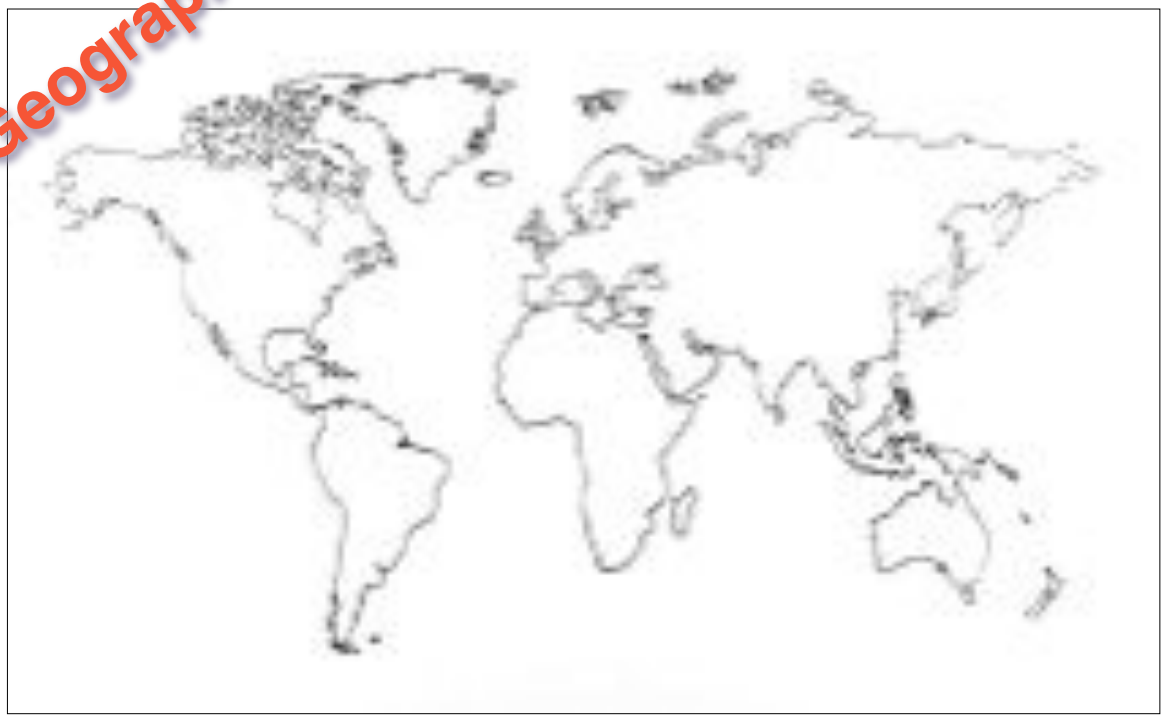
- Albers
- Lambert
- Mercator
- Sinusiodal
- Stereographic
- UTM
- Gnomic
- Orthographic

ST_Transform()

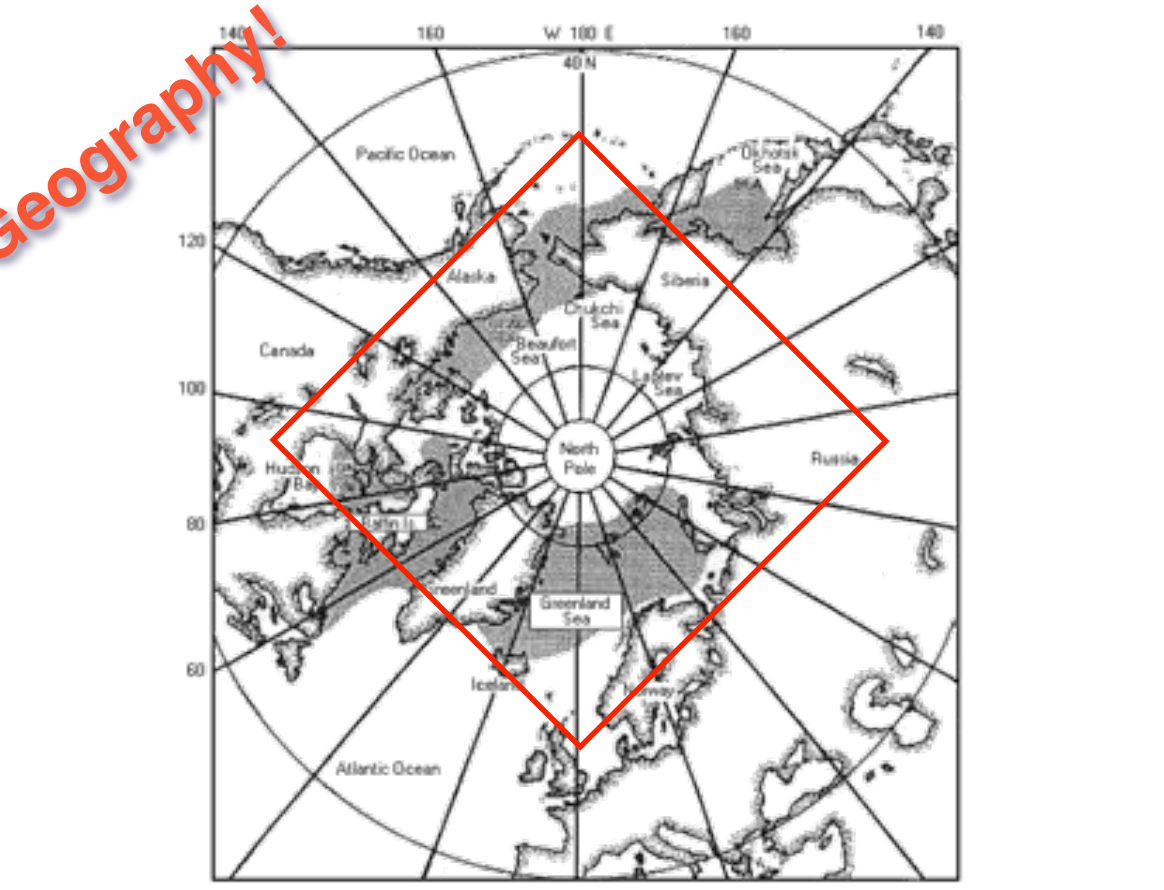
- Robinson
- Miller
- Krovak
- Azimuthal Equidistant
- **And more...**

Geography!

Geography!



Geography!

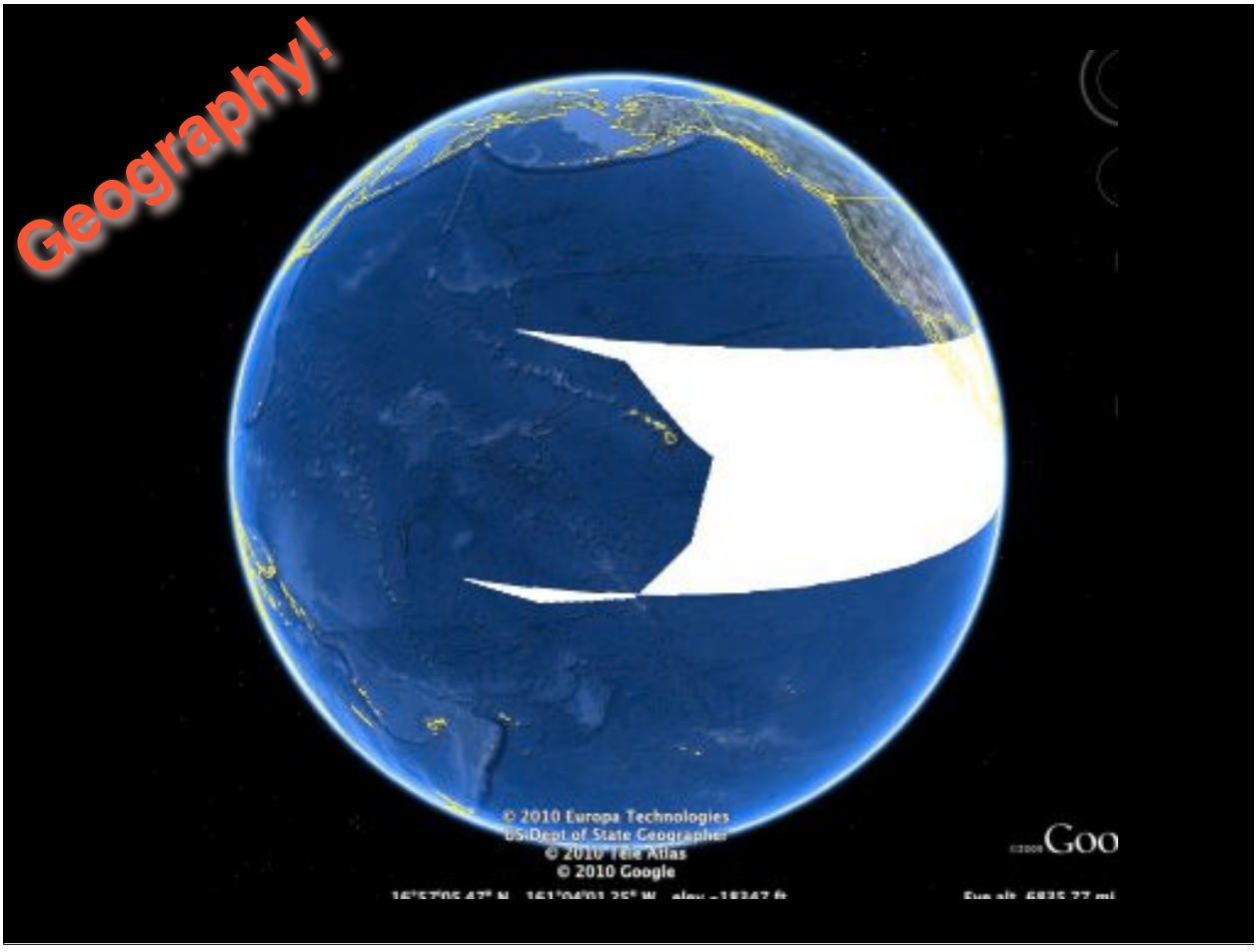


Geography!

Geography!

Geography!

Geography!



Geography!

GeoNewbies

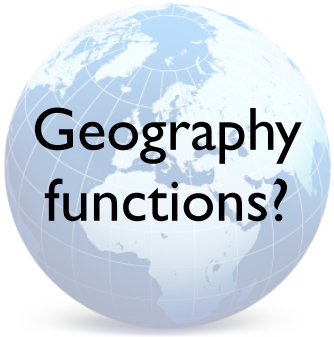
"I want to find all the address points within one mile!
My data is in lat/lon!
Google Maps rocks!"

Geography!

GeoHugies

"Yeah, I own a freaking satellite, you got a problem with that?"

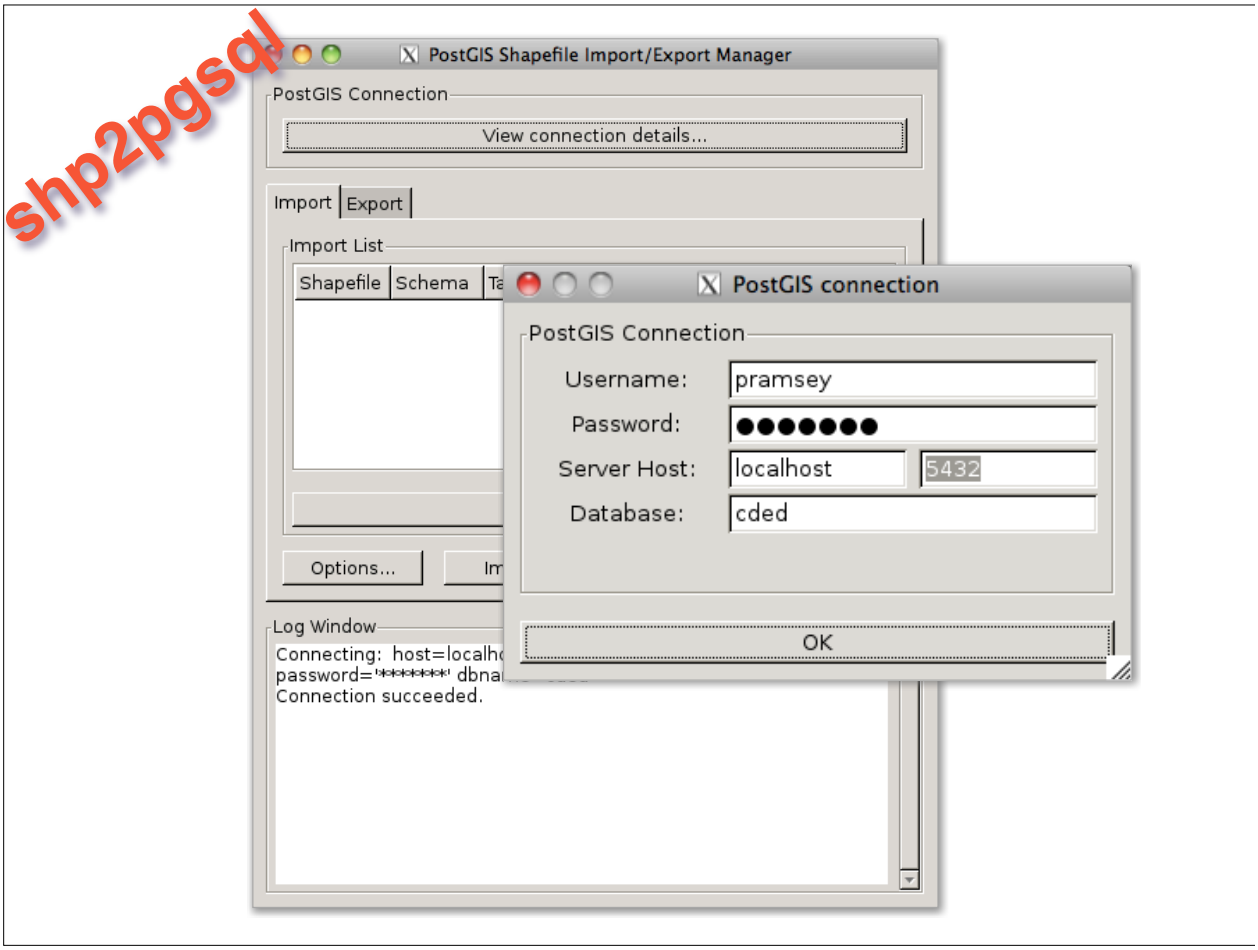
Geography!

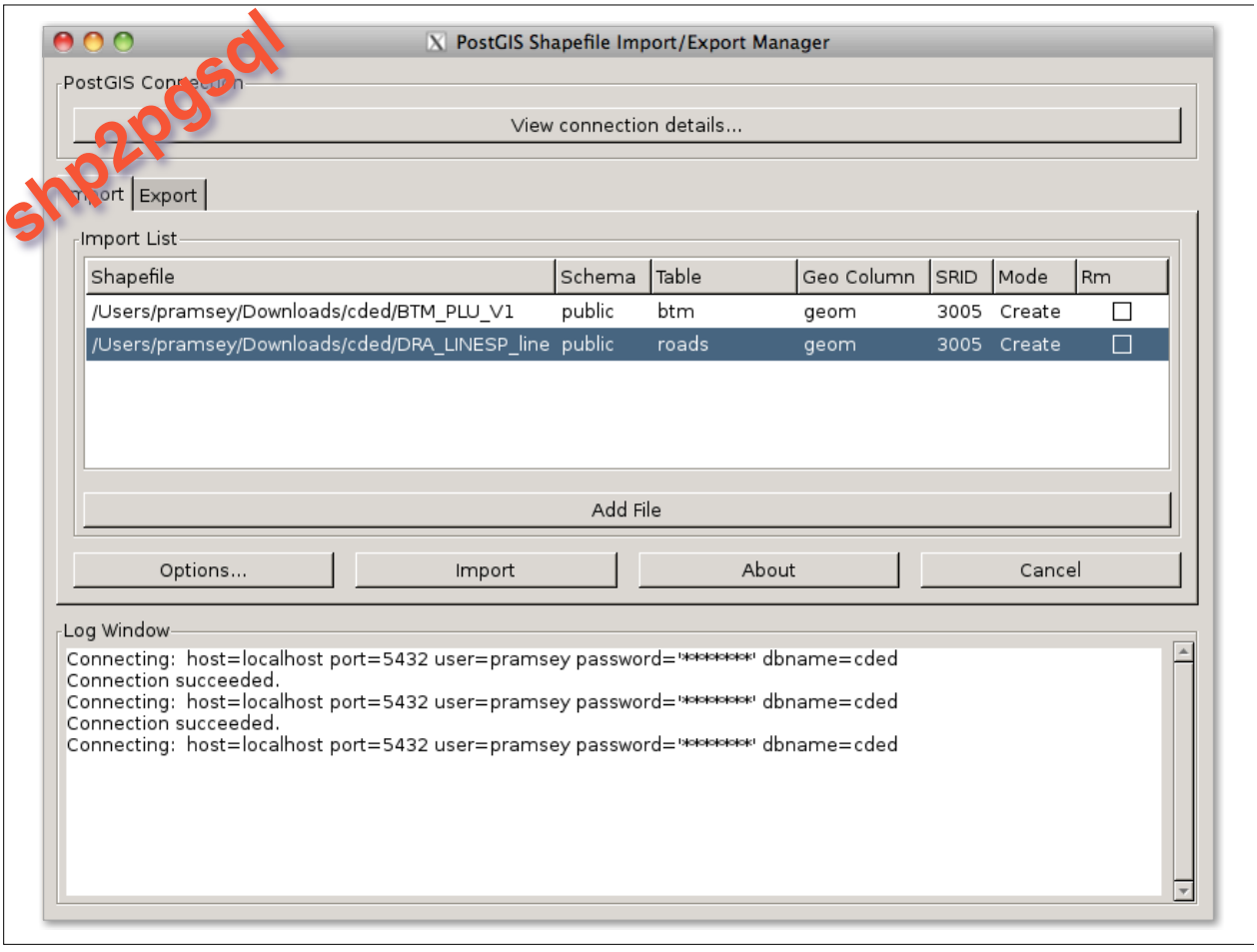


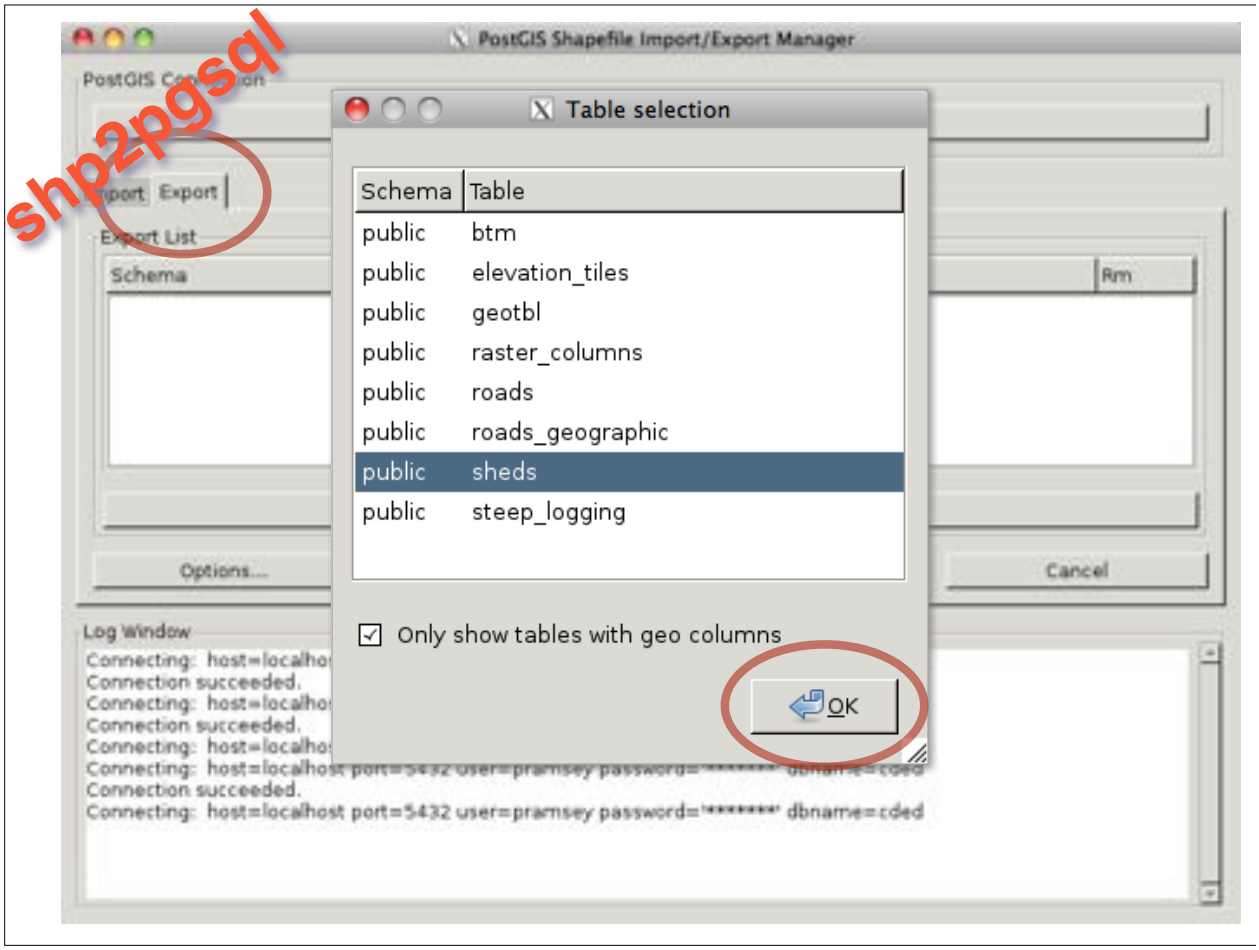
- Indexes spherical data
- ST_Intersects()
- ST_Distance()
- ST_DWithin()
- ST_Area()
- **Casts** to/from **GEOMETRY**

shp2pgsql

```
shp2pgsql -D -s 4326 \
          -i \
          countries.shp \
          countries \
| psql -U pramsey \
       -d geodatase
```


shp2pgsql
PostGIS Shapefile Import/Export Manager
PostGIS Connection
View connection details...
Import
Export
Import List
Shapefile
Schema
Options...
Log Window
Connecting: host=localhost
password='*******' dbname=cded
Connection succeeded.
PostGIS connection
PostGIS Connection
Username: pramsey
Password: ●●●●●●●
Server Host: localhost 5432
Database: cded
OK


shp2pgsql
PostGIS Shapefile Import/Export Manager
View connection details...
Export
Import List
Shapefile
Schema
Table
Geo Column
SRID
Mode
Rm
/Users/pramsey/Downloads/cded/BTM_PLU_V1
public
btm
geom
3005
Create
/Users/pramsey/Downloads/cded/DRA_LINESP_line
public
roads
geom
3005
Create
Add File
Options...
Import
About
Cancel
Log Window
Connecting: host=localhost port=5432 user=pramsey password='*******' dbname=cded
Connection succeeded.
Connecting: host=localhost port=5432 user=pramsey password='*******' dbname=cded
Connection succeeded.
Connecting: host=localhost port=5432 user=pramsey password='*******' dbname=cded



Typmod

New to 2.0!

```
CREATE TABLE
 my_spatial_table
 (
  id INTEGER,
  name VARCHAR(64),
  geom Geometry(Point,26910)
 );
```

Typmod

New to 2.0!

```
ALTER TABLE my_spatial_table
   ALTER COLUMN geom
   SET DATA TYPE
     Geometry(Point, 4326)
   USING
     ST_Transform(geom, 4326)
```

Typmod

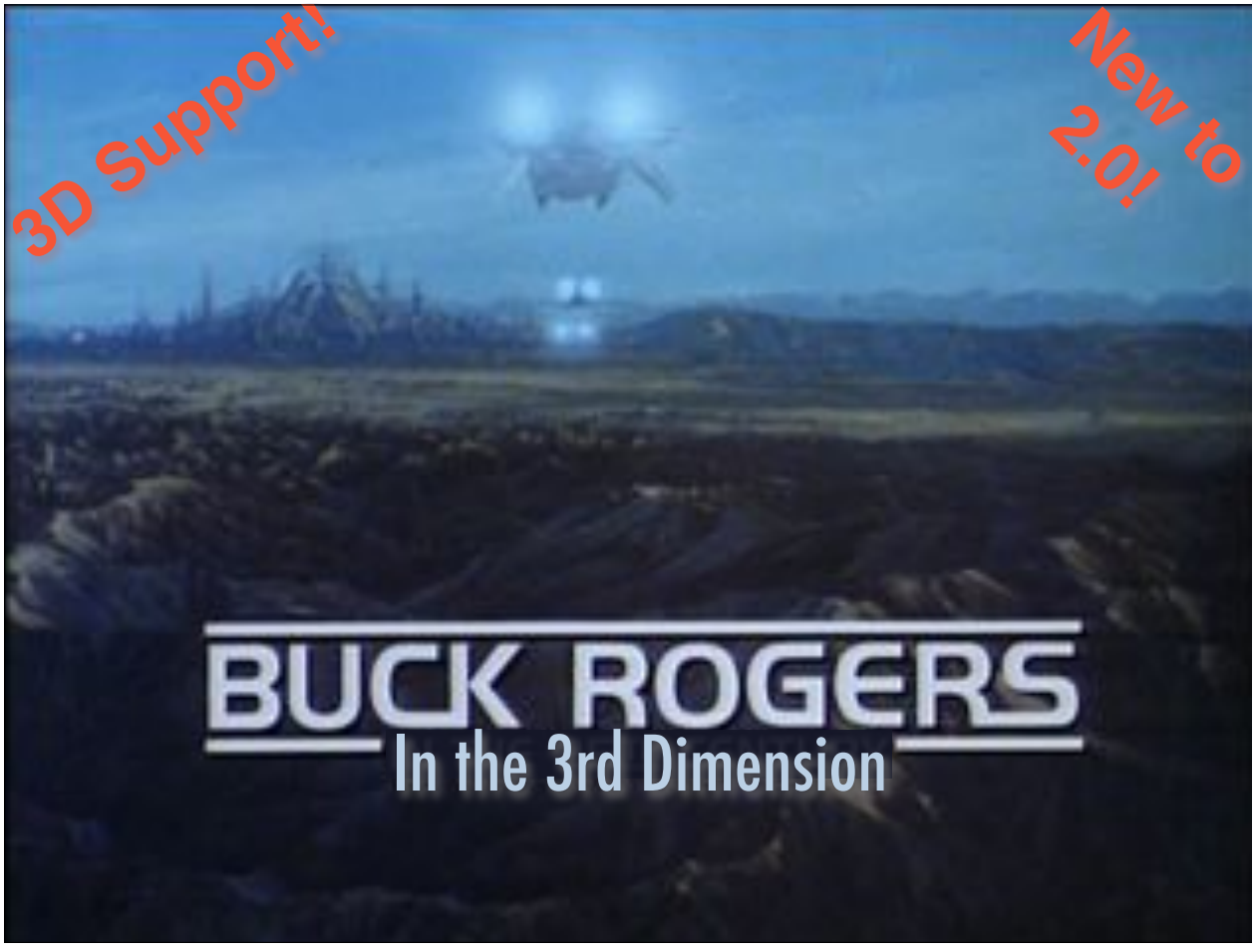
New to 2.0!

```
SELECT *
FROM geometry_columns
WHERE
 f_table_name =
 'my_spatial_table'
```

Typmod

New to 2.0!

```
-------------------+-----------------
f_table_catalog    | my_database
f_table_schema     | public
f_table_name       | my_spatial_table
f_geometry_column  | geom
coord_dimension    | 2
srid               | 4326
type               | POINT
```



3D Support!

New to 2.0!

- ST_3dDistance(geom, geom)
- ST_3dLength(geom)
- ST_3dClosestPoint(geom, geom)
- ST_3dPerimeter(geom)
- ST_3dIntersects(geom, geom)
- ST_3dDWithin(geom, geom, tolerance)

3D Support!

New to 2.0!

New 3D Types!

- TRIANGLE
- TIN
- POLYHEDRALSURFACE

3D Support!

New to 2.0!

New 3D Formats!

- ST_AsX3D(geom)
- ST_AsGML(3, ...)
- Also...
  - ST_AsText(geom)
  - ST_AsBinary(geom)

ND-Index!

New to 2.0!

```
CREATE INDEX my_index
ON my_spatial_table
USING GIST (
   geom
   gist_nd_geometry_ops
);
```

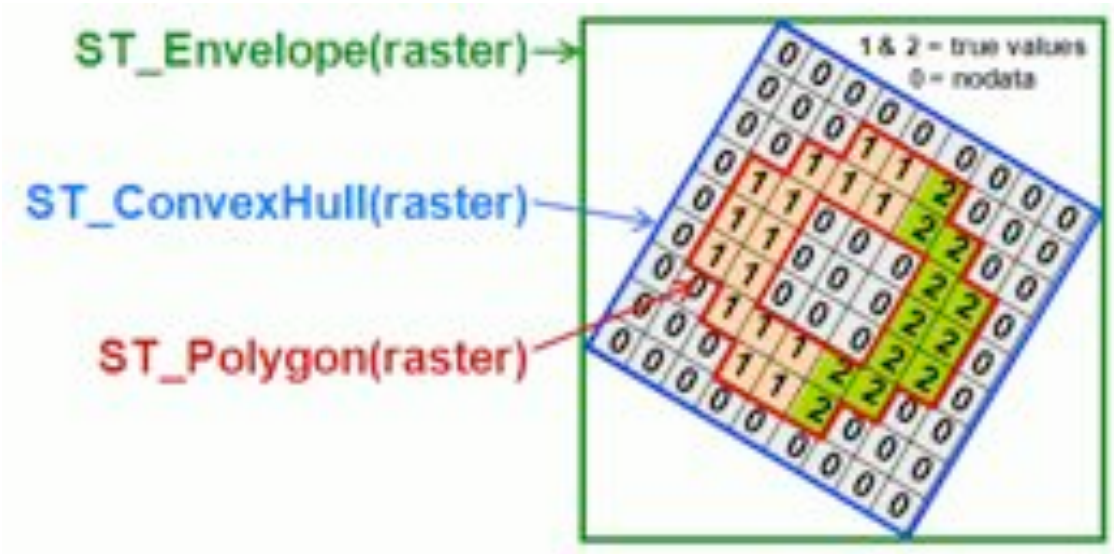
ND-Index!

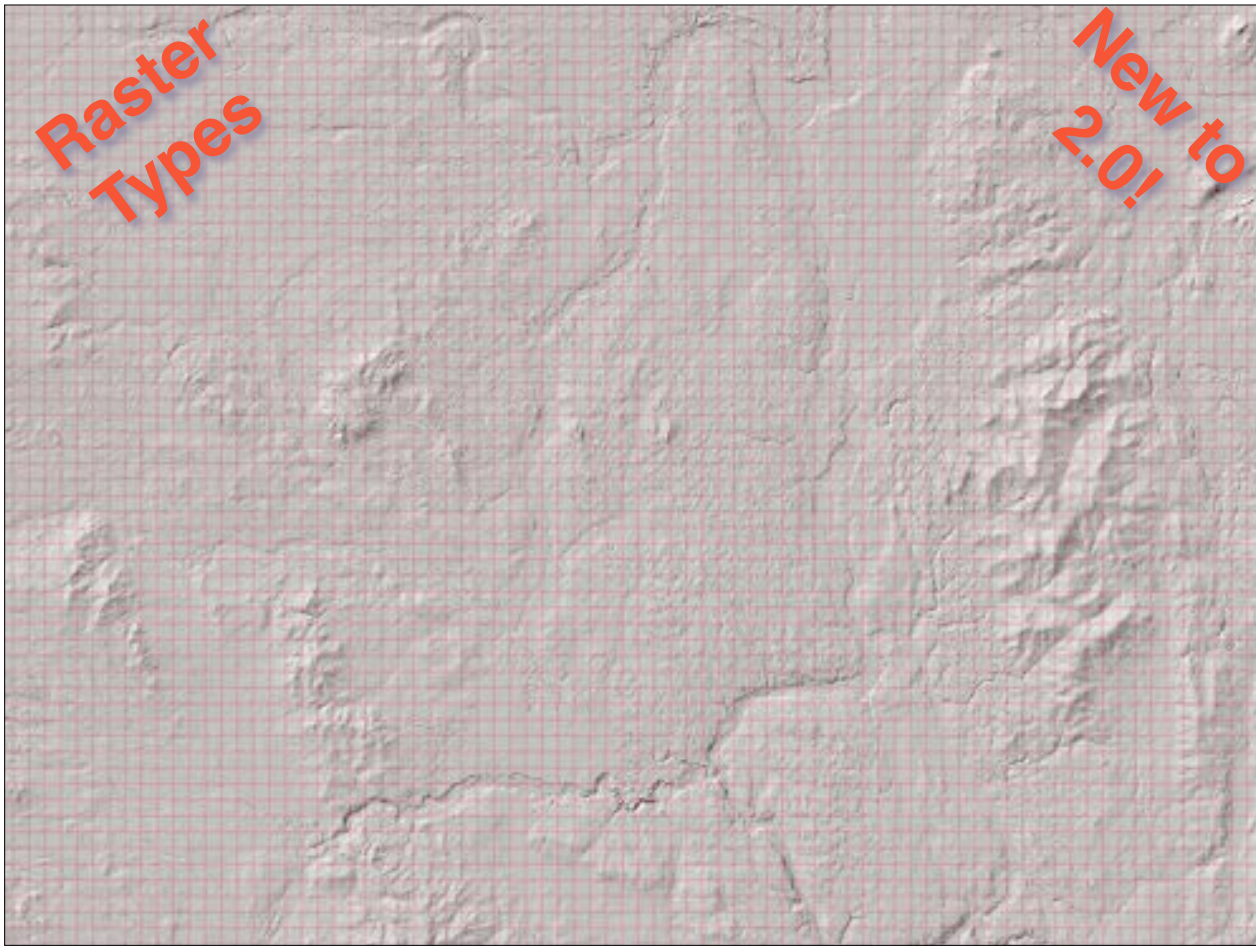
New to 2.0!

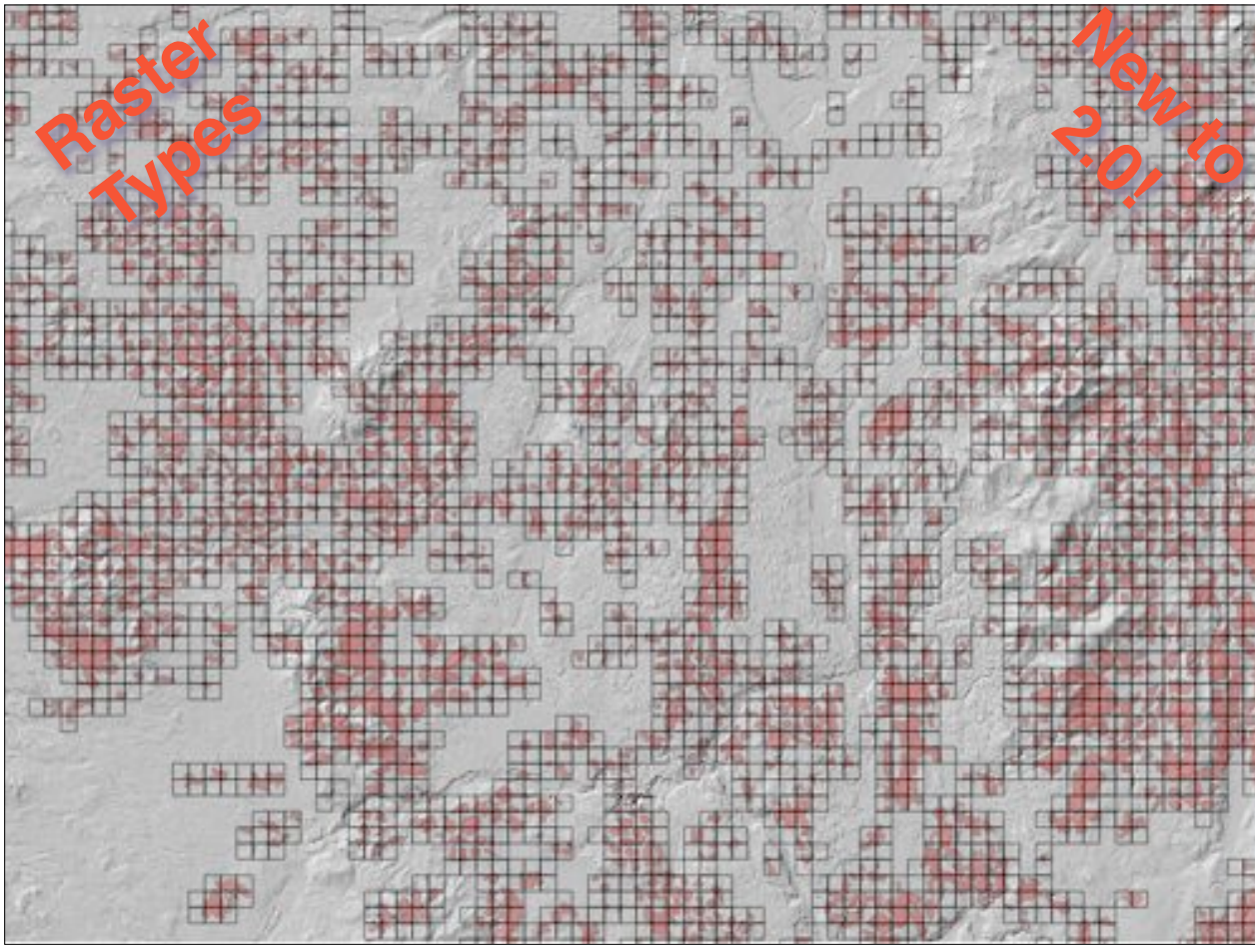
```
SELECT *
FROM my_spatial_table
WHERE
   geom &&&
    'LINESTRING Z
     (0 0 0, 10 10 10)'
```

Raster Types

New to 2.0!

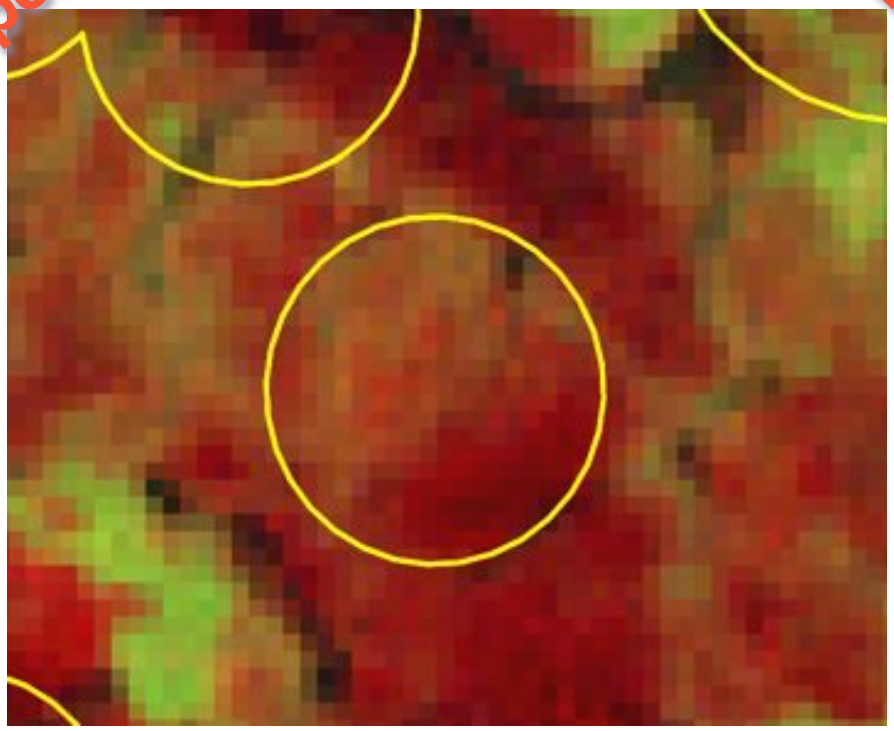



Raster Types
New to 2.0!


Raster Types
New to 2.0!

Raster Types

New to 2.0!



Raster Types

New to 2.0!

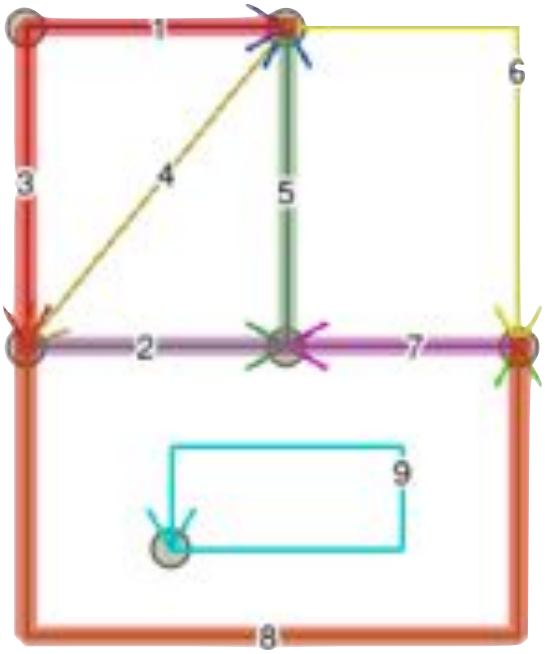
Raster Types

New to 2.0!

Topology!

New to 2.0!

Topology



Topology!

New to 2.0!

Topology!

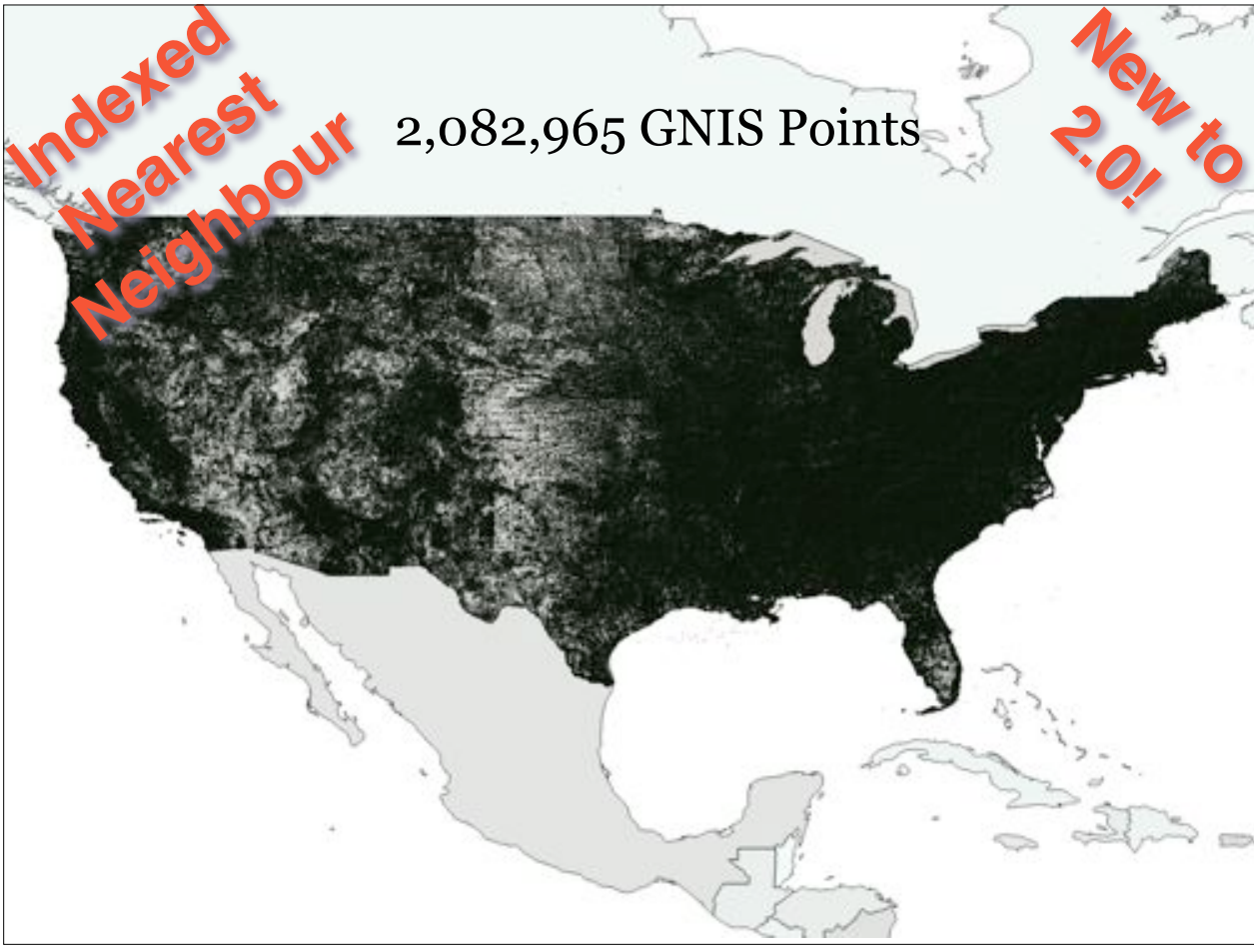
New to 2.0!

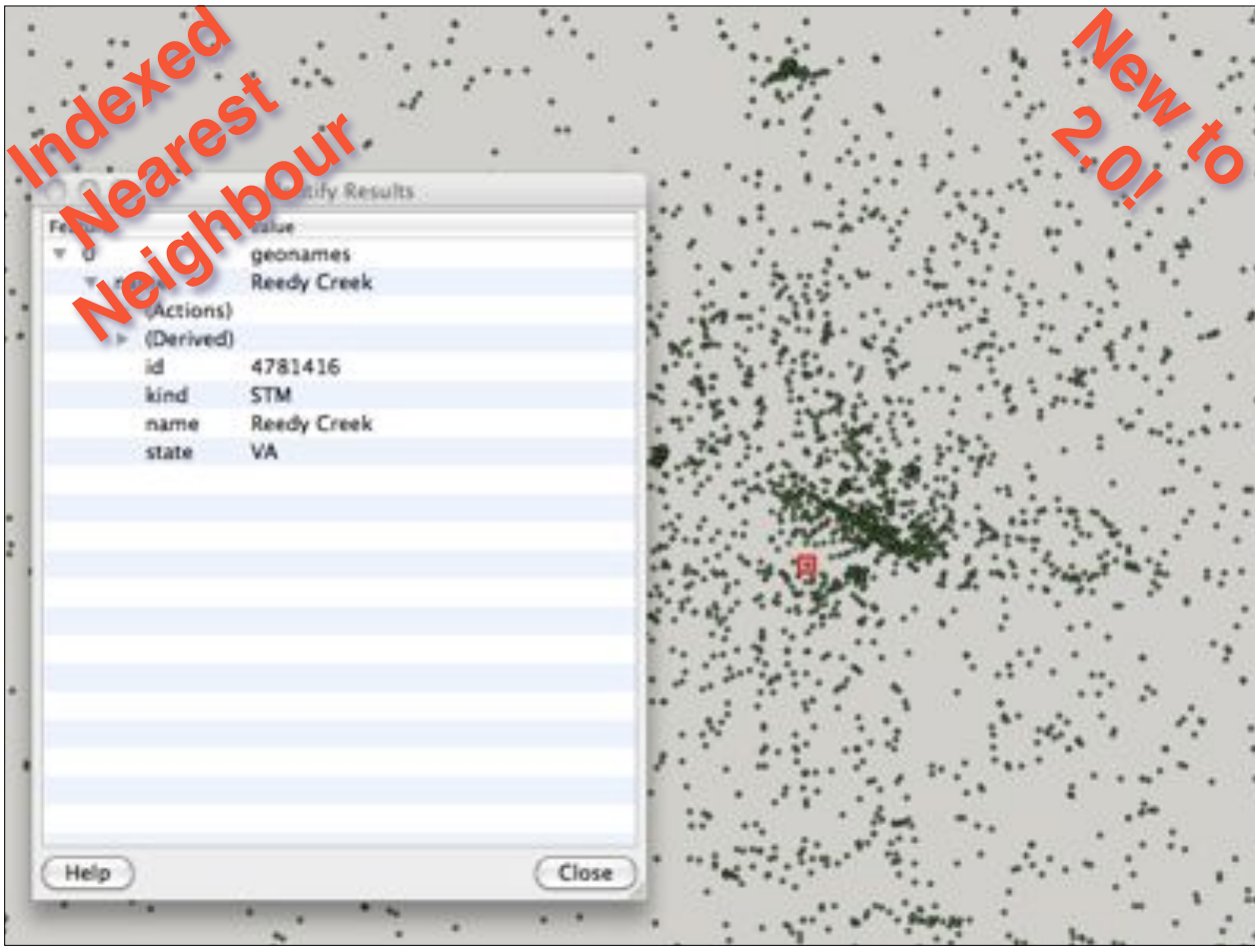
Indexed Nearest Neighbour

Indexed KNN

New to 2.0!

- KNN = K Nearest Neighbour
- Index-based tree search
- Restricted to index keys (a.k.a. bounding boxes)
  - Points: exact answer
  - Others: box-based answer





Indexed Nearest Neighbour

New to 2.0!

```
SELECT id, name, state, kind
FROM geonames
ORDER BY
    geom <->
    (SELECT geom FROM geonames
      WHERE id = 4781416)
LIMIT 10
```

Indexed Nearest Neighbour

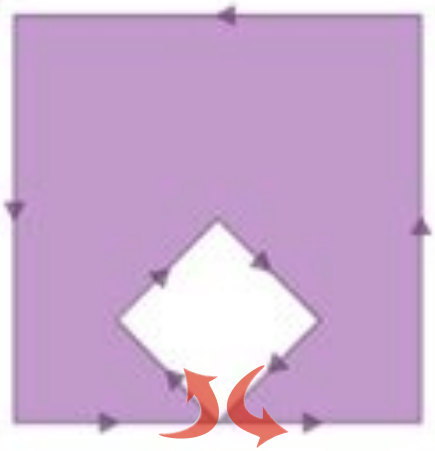
New to 2.0!

id	name	state	kind
4281416	Reedy Creek	VA	STM
4794583	Woodland Heights Baptist Church	VA	CH
4759577	Forest Hill Park	VA	PRK
6495576	Fairfield Inn And Stes Rich Nw	VA	HTL
7239038	Greater Brook Road Baptist Church	VA	CH
4778121	Patrick Henry Elementary School	VA	SCH
4746788	Berryman United Methodist Church	VA	CH
4794519	Woodland Park	VA	PPL
4780425	Progressive Holiness Church	VA	CH
4774149	Mount Calvary Cemetery	VA	CMTY

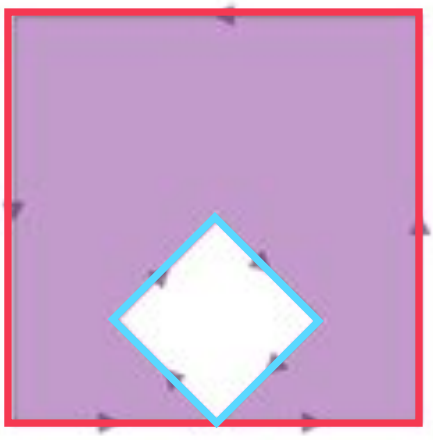
(10 rows)

Time: 9.723 ms

Validity Reporting



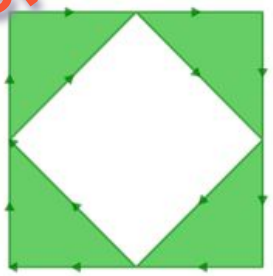
ESRI



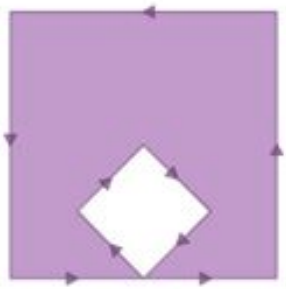
OGC

Validity Reporting

ST_IsValidReason()
ST_IsValid()



Interior is disconnected[-2 0]

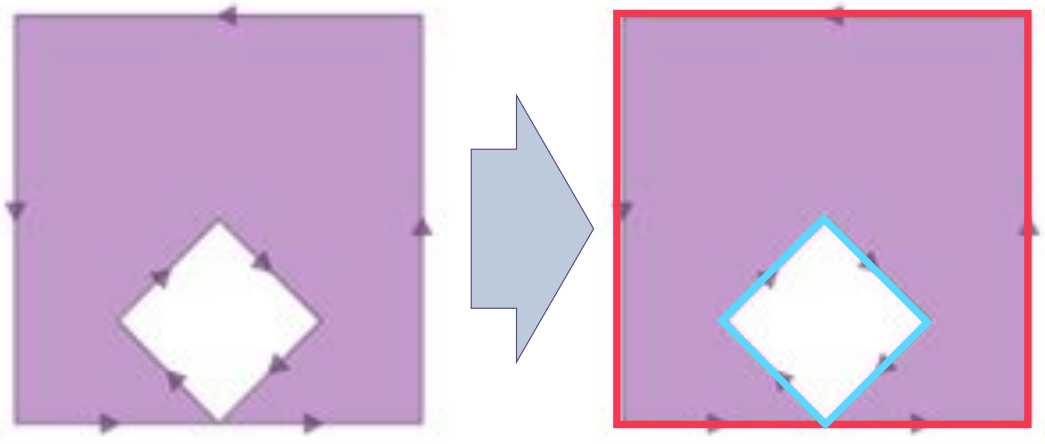


Ring Self-intersection[2 0]

Validity Repair

New to 2.0!

ST_MakeValid()



Curve Distance!

Coming in 2.1!

```
SELECT ST_Distance(
   'CIRCULARSTRING(...)',
   'CURVEPOLYGON(...)'
);
```

More Performance!!

Coming in 2.1!

- Raster ST_Union(), native implementation, **10x faster**
- Geography ST_Distance() and ST_DWithin (), internally indexed, **20-30x faster**
- ST_DumpPoints(), native implementation, **10x faster**
- New R-Tree splitter, **20-30% faster**
- New N-D and geography statistics calculations, **20-30% faster**


Do you have questions about...?
4 FRUIT
COMBO
+4
PostGIS
Feature Frenzy!!!