Paul Ramsey presents....

PostGIS Feature Frenzy!!!

frenzy because “state of postgis” talks only covered leading edge, nothing covering all the good features already there
frenzy because our standard 20-25 minute talk barely has enough time to handle introductory material
frenzy because everyone likes to rack up lots of points really fast
but... invited talk!
one HOUR???

All glory to the Hypnotoad!

Thursday, March 12, 15

Start off with some context about how I get paid to do all this fun frenzy work...

All glory to the Hypnotoad.

CARTODB

Geospatial on the cloud

CARTODB

Geospatial on the cloud

Powered by PostGIS

PostGIS

is a

“Spatial Database”

Database

database provides
random access data storage system
types, indexes, functions for STANDARD TYPES

Database

- **Types**
  - string, float, date

database provides
random access data storage system
types, indexes, functions for STANDARD TYPES

Database

- **Types**
  - string, float, date
- **Indexes**
  - b-tree, hash

database provides
random access data storage system
types, indexes, functions for STANDARD TYPES

Database

- **Types**
  - string, float, date
- **Indexes**
  - b-tree, hash
- **Functions**
  - strlen(string), pow(float, float), now()

Thursday, March 12, 15

database provides
random access data storage system
types, indexes, functions for STANDARD TYPES

SPATIAL Database

- **Spatial Types**
  - geometry, geography
- **Spatial Indexes**
  - r-tree, quad-tree, kd-tree
- **Spatial Functions**
  - ST_Length(geometry), ST_X(geometry)

spatial database provides
random access data storage system
SPATIAL types, SPATIAL indexes, SPATIAL functions

Spatial Database

spatial databases your likely to run into are
 PostGIS, Oracle and SQL Server.
Others include
 DB2 Spatial, Informix Spatial, Netezza, Teradata, SpatialLite.
Spatial functionality has gone from being a rarity to being a standard feature.

What does PostGIS do?

That is interesting, but... what does it do?
PostGIS answers questions that have a spatial component.
It can answer questions on very large collections of spatial data,
and against very complex data models.

“What parcels are within 1km of this fire?”

That sounds like a GIS question!
How many lines of code should it take to solve it?

```
SELECT owner_phone
FROM parcels
WHERE ST_DWithin(
        geom,
        ‘POINT()’,
        1000 );
```

One line!
Using one spatial function, a coordinate,
and a table of parcel data, we can generate
a classic GIS “alert list” of
people to phone about the fire.

“How far did the bus travel last week?”

```
SELECT
  Sum(ST_Length(geom))
FROM
  vehicle_paths
WHERE
  id = 12
AND
  date > Now() - ‘7d’;
```

One SQL statement to answer a location services query.

“How much land is there in each zoning category?”

```
SELECT
  category,
  Sum(ST_Area(geom))
FROM
  zoning
GROUP BY
  category;
```

We can use SQL to summarize our whole data set, with just one query!

SQL in a database is very powerful, **more powerful** than desktop GIS in terms of amount of code required and the size of datasets than can be queried.

History!

Thursday, March 12, 15

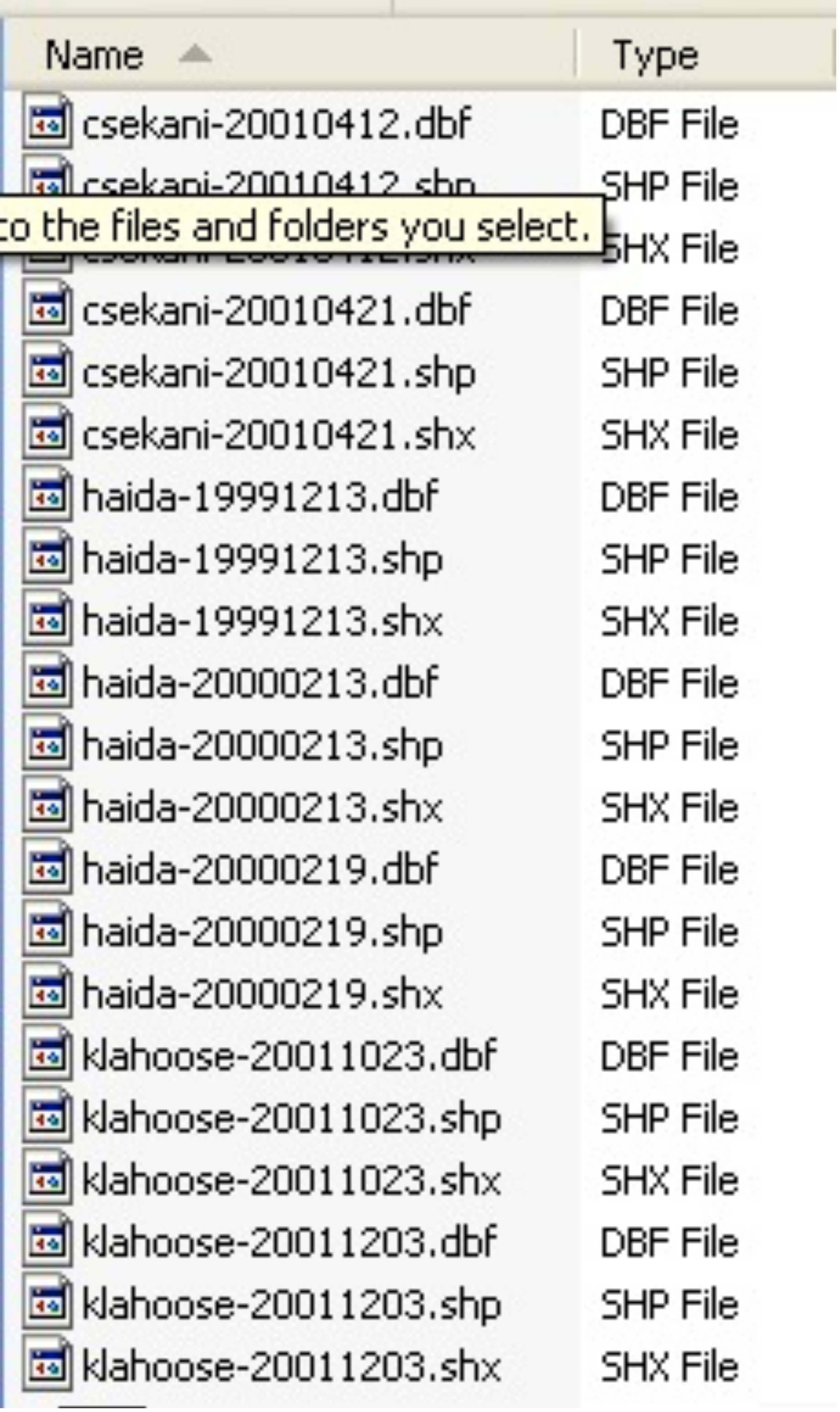
I didn’t write PostGIS. I’ve been a major developer since 2008.
The first versions (0.1 to 0.7) were done by Dave Blasby on my left.
I did the build system, documentation and Java components.
This picture is actually a little too new, at the time of PostGIS 0.1
(around 2001) there were only 5 people in company.

Thursday, March 12, 15

I didn’t write PostGIS. I’ve been a major developer since 2008.
The first versions (0.1 to 0.7) were done by Dave Blasby on my left.
I did the build system, documentation and Java components.
This picture is actually a little too new, at the time of PostGIS 0.1
(around 2001) there were only 5 people in company.

“Managing changing data in shape files is a pain in the ______!”

PostGIS was the answer to the question “how are we going to manage this changing data?!”



o the files and folders you select.

Name	Type
csekani-20010412.dbf	DBF File
csekani-20010412.shp	SHP File
csekani-20010412.shx	SHX File
csekani-20010421.dbf	DBF File
csekani-20010421.shp	SHP File
csekani-20010421.shx	SHX File
haida-19991213.dbf	DBF File
haida-19991213.shp	SHP File
haida-19991213.shx	SHX File
haida-20000213.dbf	DBF File
haida-20000213.shp	SHP File
haida-20000213.shx	SHX File
haida-20000219.dbf	DBF File
haida-20000219.shp	SHP File
haida-20000219.shx	SHX File
klahoose-20011023.dbf	DBF File
klahoose-20011023.shp	SHP File
klahoose-20011023.shx	SHX File
klahoose-20011203.dbf	DBF File
klahoose-20011203.shp	SHP File
klahoose-20011203.shx	SHX File

We had a collection of “areas of interest” for treaty negotiations that were passed through modeling software. The negotiators would provide ever changing versions they wanted analyses on, so knowing what the history was, was important.

<X>

Database model could provide history, easy publication distribution, and automation of analysis runs. (That was the theory, in fact we never got around to using the database for this purpose.)

Table	Date	Geometry
Haida	19991213	POLYGON()
Haida	20000213	POLYGON()
Haida	20000219	POLYGON()
Carrier	20010412	POLYGON()
Carrier	20010421	POLYGON()
Klahoose	20011023	POLYGON()
Klahoose	20011203	POLYGON()

We had a collection of “areas of interest” for treaty negotiations that were passed through modeling software. The negotiators would provide ever changing versions they wanted analyses on, so knowing what the history was, was important.

<X>

Database model could provide history, easy publication distribution, and automation of analysis runs.

(That was the theory, in fact we never got around to using the database for this purpose.)

Database Advantages

- Query all areas at once
- Publish in one step
- Manage in one place

More History!!!

That was the beginning, but there is more.

Thursday, March 12, 15

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

2001, May	0.1	Objects / Indexes

Thursday, March 12, 15

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

2001, May	0.1	Objects / Indexes
2001, July	0.5	Functions

Thursday, March 12, 15

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

2001, May	0.1	Objects / Indexes
2001, July	0.5	Functions
2003, November	0.8	OGC SFSQL

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

2001, May	0.1	Objects / Indexes
2001, July	0.5	Functions
2003, November	0.8	OGC SFSQL
2005, April	1.0	Lightweight Geometry

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

2001, May	0.1	Objects / Indexes
2001, July	0.5	Functions
2003, November	0.8	OGC SFSQL
2005, April	1.0	Lightweight Geometry
2010, February	1.5	Geography

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

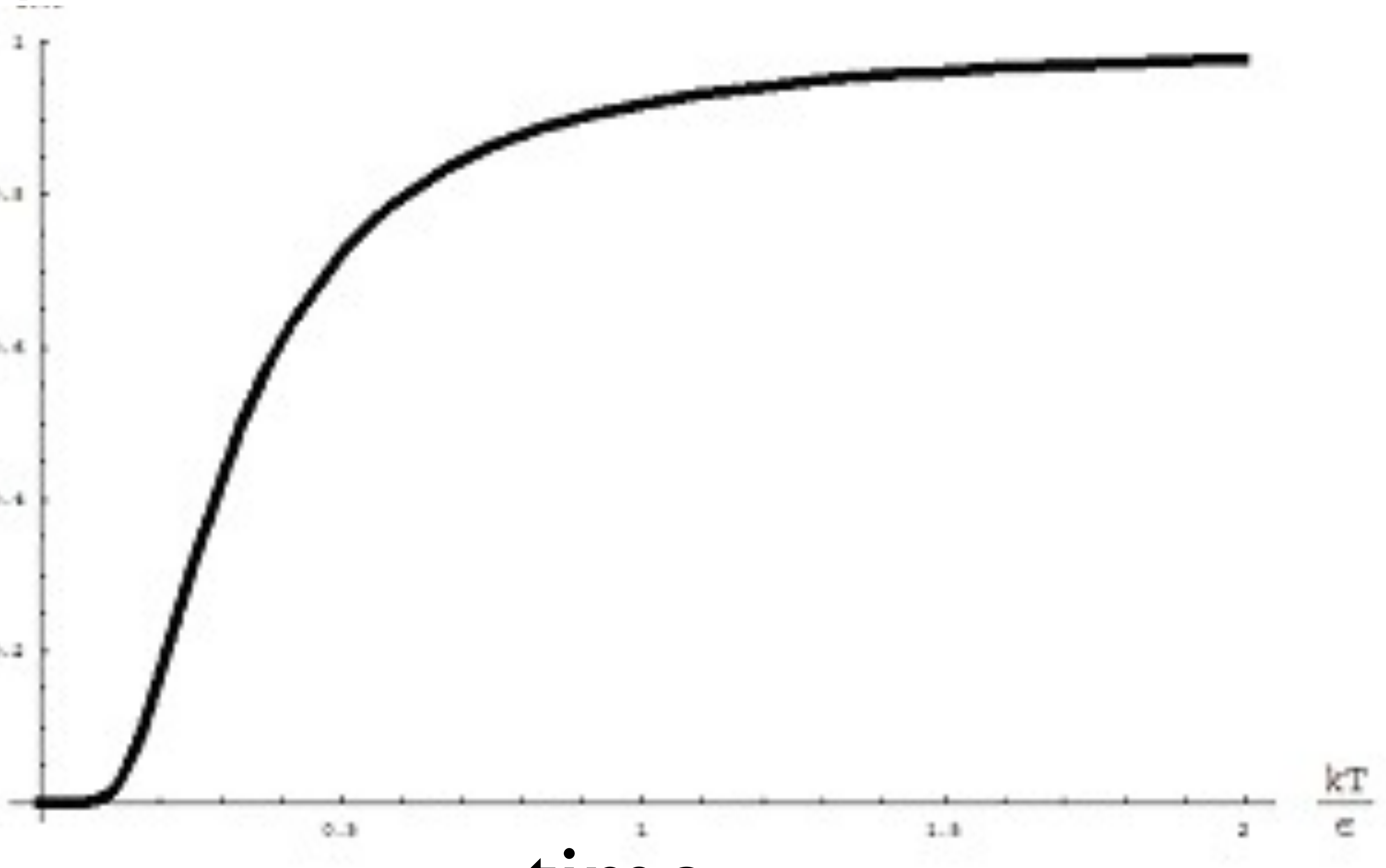
2001, May	0.1	Objects / Indexes
2001, July	0.5	Functions
2003, November	0.8	OGC SFSQL
2005, April	1.0	Lightweight Geometry
2010, February	1.5	Geography
2012, April	2.0	Raster

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

2001, May	0.1	Objects / Indexes
2001, July	0.5	Functions
2003, November	0.8	OGC SFSQL
2005, April	1.0	Lightweight Geometry
2010, February	1.5	Geography
2012, April	2.0	Raster
2013, August	2.1	Speed/polish

overview of release history
release tempo is slowing
are developers getting tired
is community drifting away

perfection



time

asymptotically approaching perfection

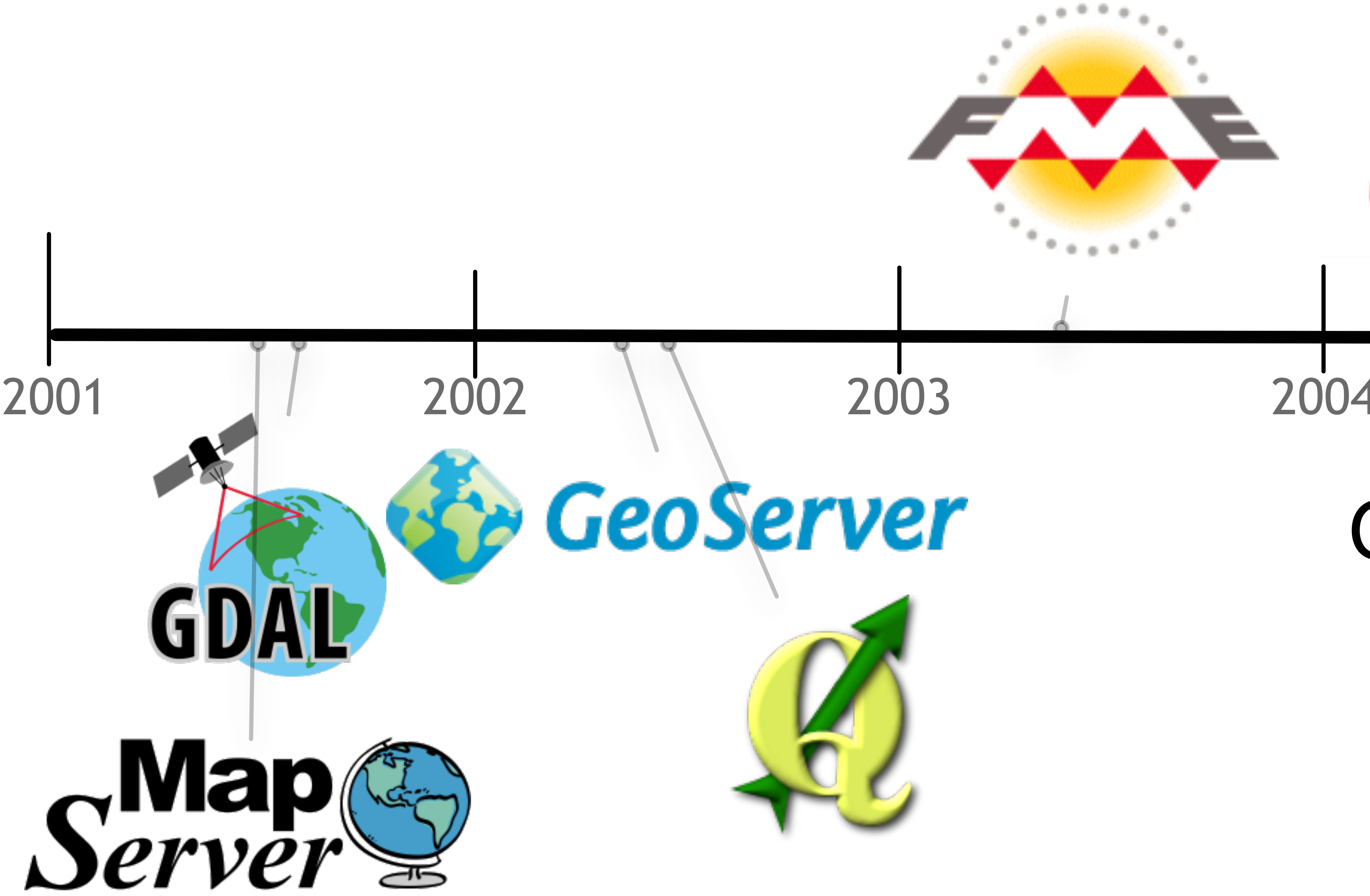
PostGIS 3.0?

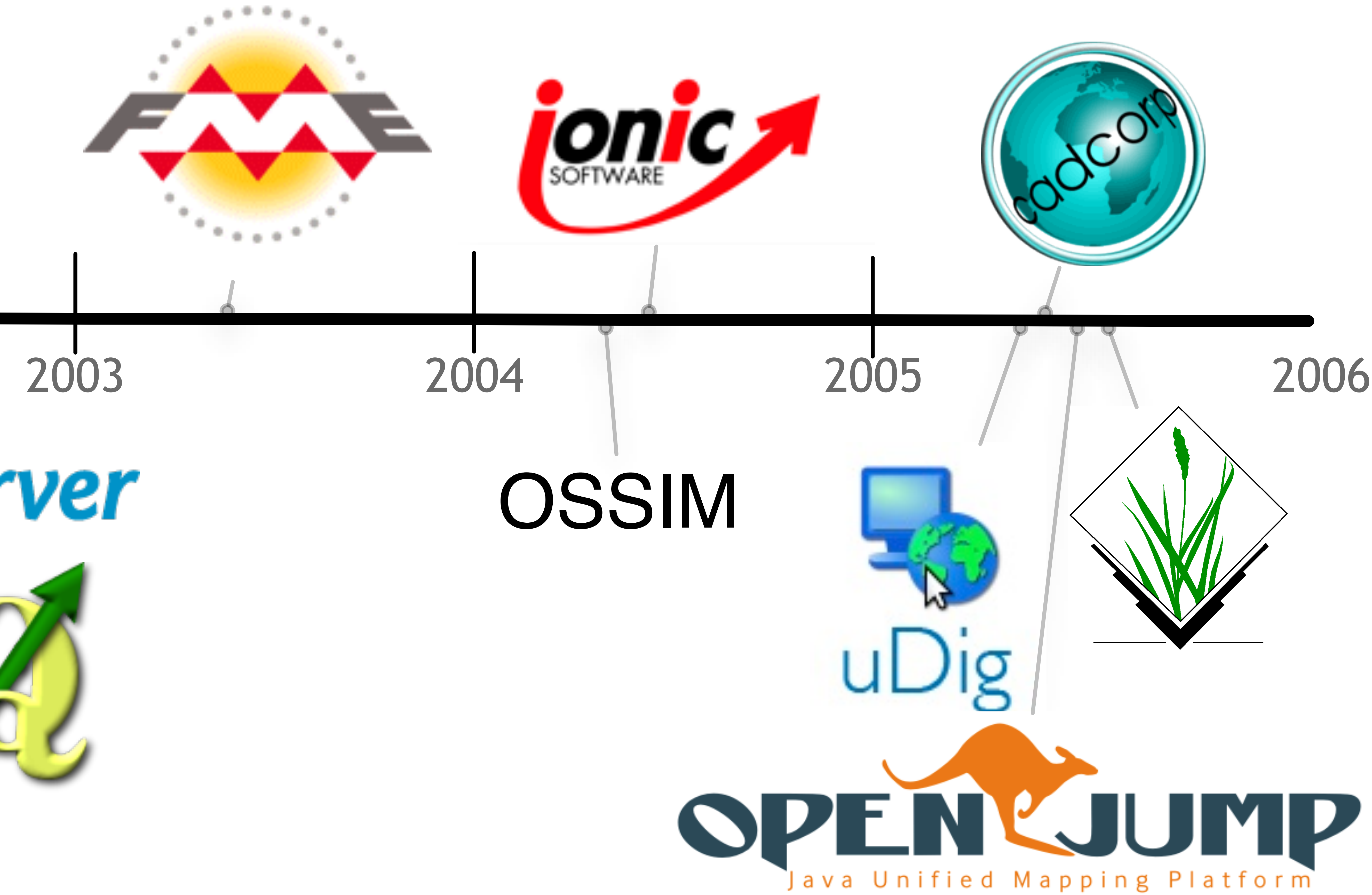
Thursday, March 12, 15

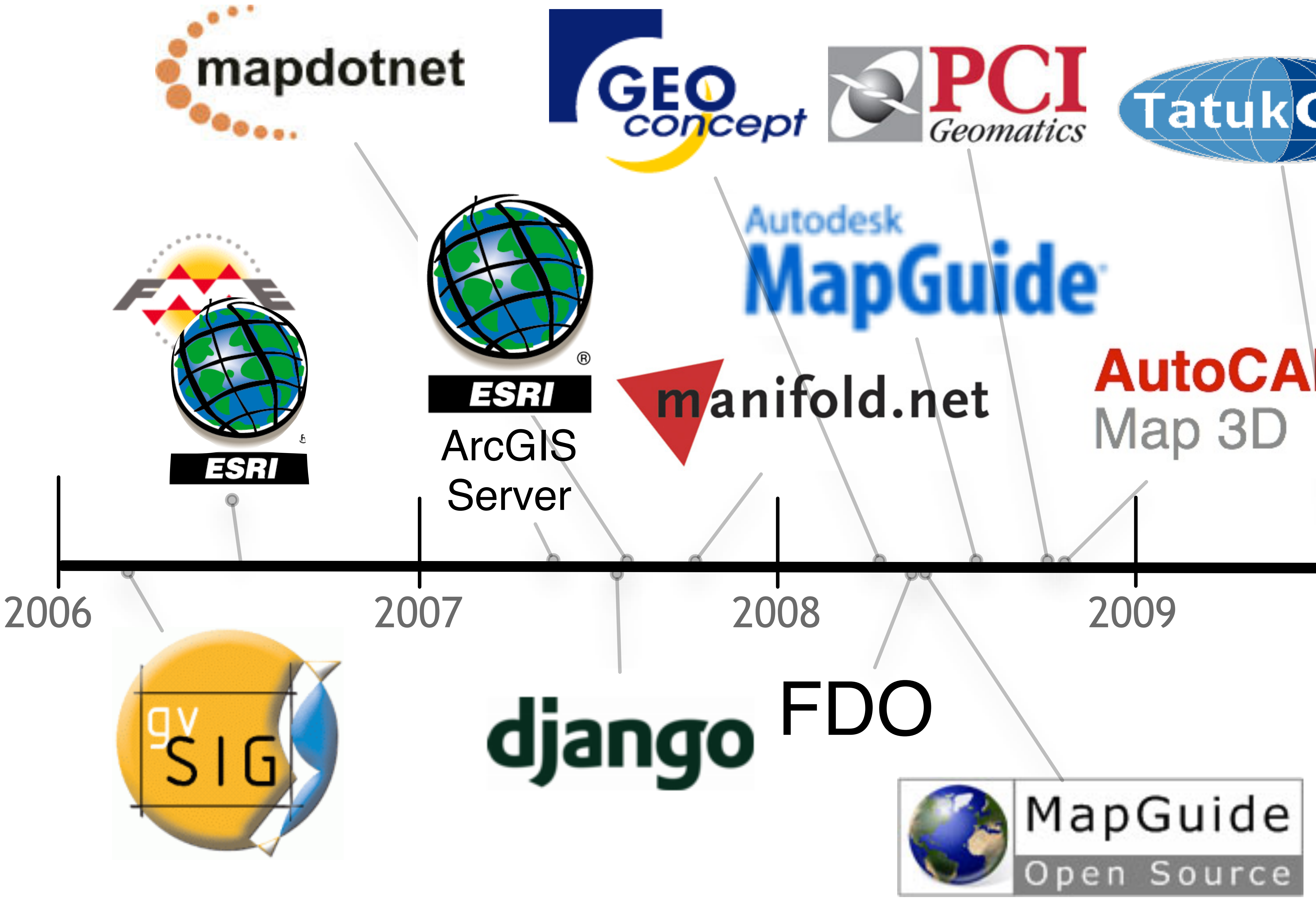
so when will postgis 3 arrive?
never

Yet More History!!!

More to history than just software features.
PostGIS has gone from curiosity to industry standard.


FME
2001
2002
2003
2004
GDAL
GeoServer
Map Server


FME
ionic
SOFTWARE
cadcorp
2003
2004
2005
2006
rver
OSSIM
uDig
OPEN JUMP
Java Unified Mapping Platform



ArcGIS Server

FDO

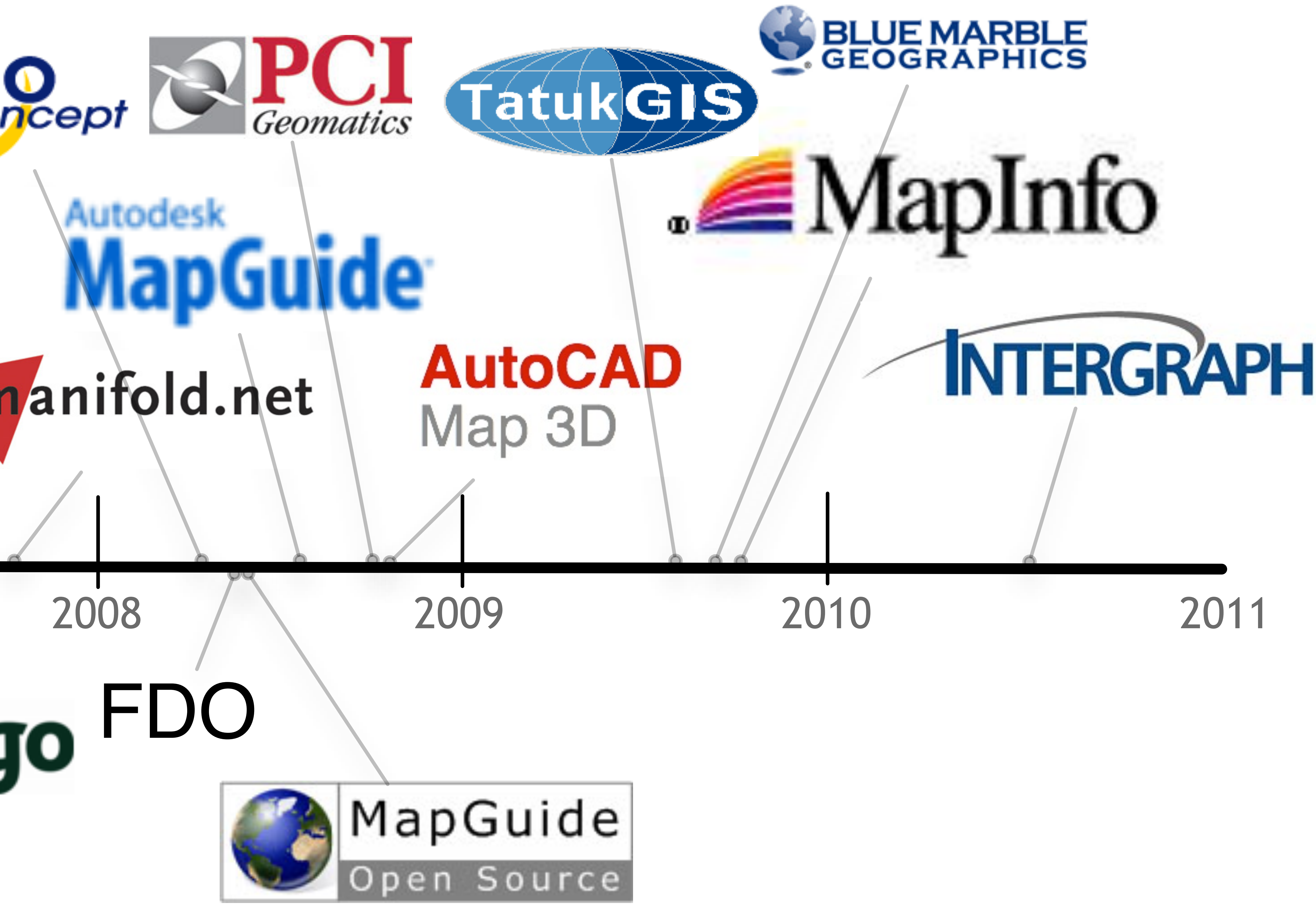
2006

2007

2008

2009

lots of proprietary companies at the end of this timeline!



Thursday, March 12, 15

lots of proprietary companies at the end of this timeline!

Widely Supported

Widely Supported

uDig
MapServer
MapGuide
Open Source
GeoServer
gvSIG
django
GDAL

Widely Supported

ESRI
FME
AutoCAD
Map 3D
uDig
cadcorp
MapServer
MapInfo
MapGuide
Open Source
INTERGRAPH
GeoServer
gvSIG
GDAL
django

Widely DEPLOYABLE

OPENSHIFT

amazon
web services™

heroku

“Why are these companies supporting PostGIS?”

Thursday, March 12, 15

because it makes them money

first as DIFFERENTIATOR
GeoConcept and CadCorp both supported PostGIS because IGN (national mapping agency of France) wanted PostGIS-enabled tools.
It gave them access to more business.

Thursday, March 12, 15

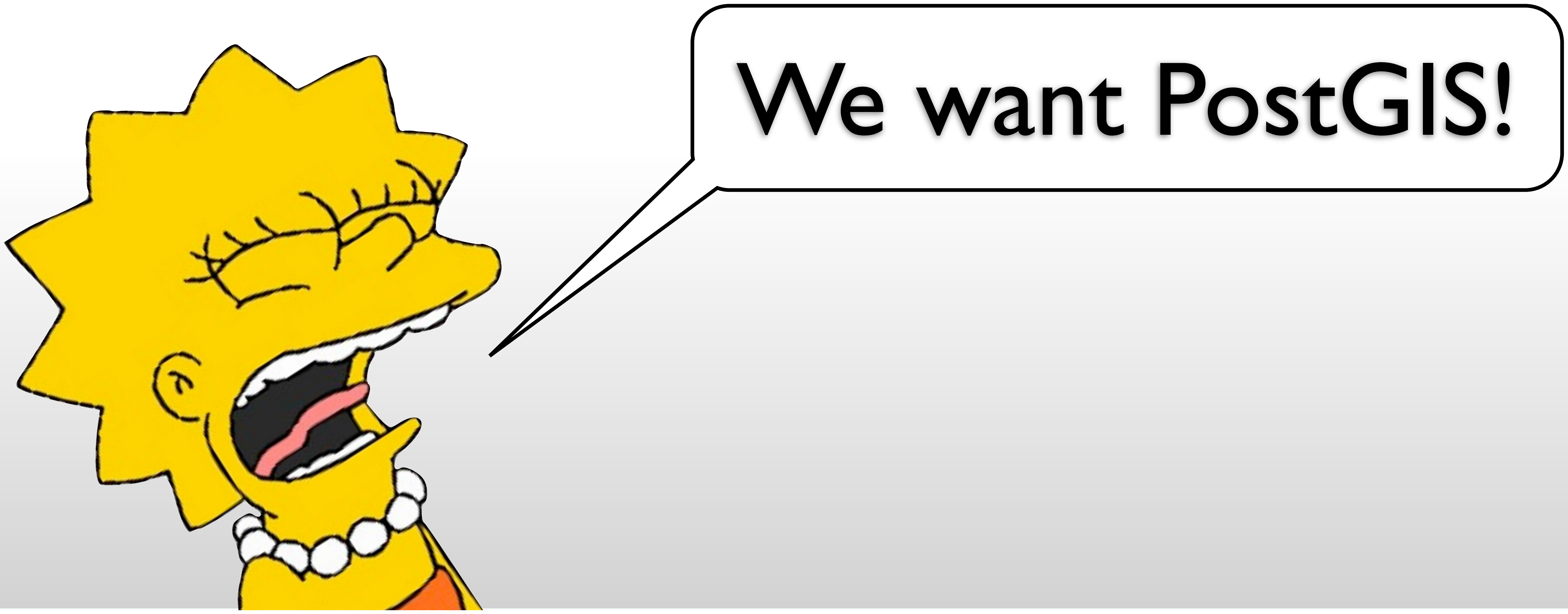
first as DIFFERENTIATOR
GeoConcept and CadCorp both supported PostGIS because IGN (national mapping agency of France) wanted PostGIS-enabled tools.
It gave them access to more business.

Thursday, March 12, 15

Then as required BASIC FEATURE
Even the big three
ESRI, MapInfo and InterGraph have come on board
because their customers have been saying
they want it.

Then as required BASIC FEATURE
Even the big three
ESRI, MapInfo and InterGraph have come on board
because their customers have been saying
they want it.

ESRI MapInfo INTERGRAPH



Then as required BASIC FEATURE
Even the big three
ESRI, MapInfo and InterGraph have come on board
because their customers have been saying
they want it.

Who is using PostGIS?

And who are those customers?
They are legion.

Corporations

wsi
CALTHORPE ASSOCIATES
URBAN DESIGNERS, PLANNERS, ARCHITECTS
infoterra
an EADS Astrium company
Google™
SAIC®
From Science to Solutions
DIGITALGLOBE®
Ball
Ball Aerospace
& Technologies Corp.
REDFIN
The New York Times
BAE SYSTEMS
ZONAR
INSPECT • TRACK • KNOW
ERICSSON
URS

On the private sector side,
WSI also known as the Weather channel
The New York Times, running PostGIS behind their Django instance.
InfoTerra in the UK, managing all of Ordnance Survey in their PostGIS
DigitalGlobe, managing their web-based image delivery
SAIC, BAE, Ball Systems, all doing defence systems.
RedFin and Zonar, startup companies,

Governments

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE
INSTITUTO GEOGRÁFICO PORTUGUÊS
THE GOVERNMENT OF WESTERN AUSTRALIA
SAINT PAUL
The Most Livable City in America
Sécurité publique Québec
Landgate
NREL
National Renewable Energy Laboratory
Pierce County
THE GREAT SEAL OF THE STATE OF MINNESOTA
L'ETOILE DU NORD
1858
NAV CANADA
Natural Resources Canada
Ressources naturelles Canada
Canada

On the government side,
National levels in France and Portugal and Canada
Regional levels like MN, and Quebec
Local levels like Pierce County, the City of St Paul, and many more
and quasi-governmental outfits like NREL and NavCanada all use PostGIS.

Governments

INSTITUT GEOGRAPHIQUE NATIONAL
IGN
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE
THE GOVERNMENT OF WESTERN AUSTRALIA
INSTITUTO GEOGRÁFICO PORTUGUÊS
SAINT PAUL
The Most Livable City in America
Sécurité publique Québec
Landgate
NREL
National Renewable Energy Laboratory
Pierce County
THE GREAT SEAL OF THE STATE OF MINNESOTA
L'ETOILE DU NORD
1858
NAV CANADA
Natural Resources Canada
Ressources naturelles Canada
Canada

On the government side,
National levels in France and Portugal and Canada
Regional levels like MN, and Quebec
Local levels like Pierce County, the City of St Paul, and many more
and quasi-governmental outfits like NREL and NavCanada all use PostGIS.

Start the Frenzy!!!

Thursday, March 12, 15

I have three #1 features of PostGIS
THREE
first...

Freedom!

Liberty!

As you all know, since you’re here at FOSS4G,
the most important feature of PostGIS is freedom!
A free-as-in-freedom spatial database can be extended, deployed and shared any way you want.

NoSQL?

You have heard of the “NoSQL” movement, perhaps?
The number one feature of PostGIS is that it
puts the power of hundreds of complex,
performance optimized
spatial functions at your fingertips
through powerful declarative language we call
SQL

SQL Goodness!

Thursday, March 12, 15

You have heard of the “NoSQL” movement, perhaps?
The number one feature of PostGIS is that it
puts the power of hundreds of complex,
performance optimized
spatial functions at your fingertips
through powerful declarative language we call
SQL

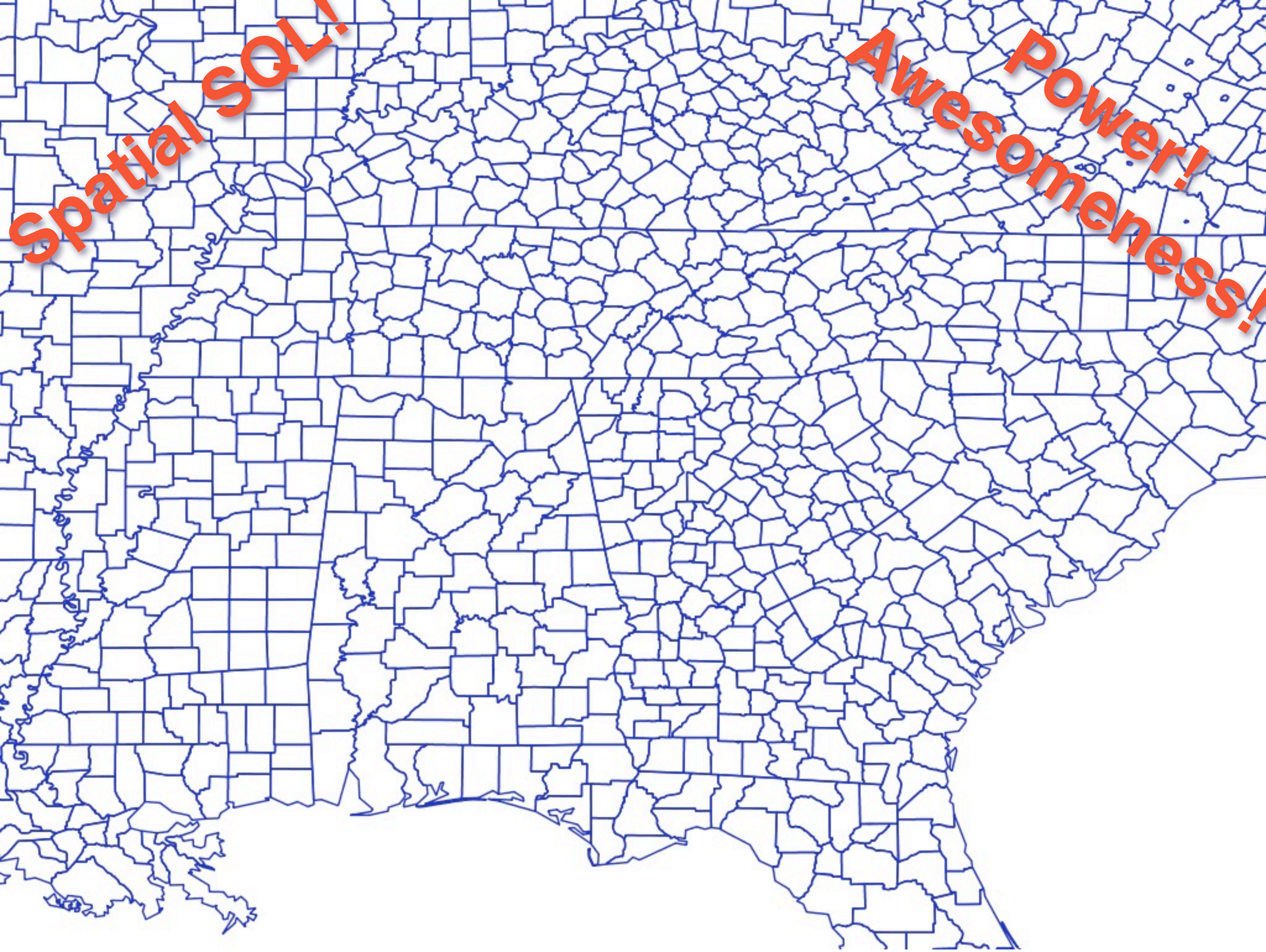
Thursday, March 12, 15

SPATIAL SQL IS #1 FEATURE!!!

For example, suppose you had a big customer database, millions of customers, with geocoded addresses, suppose you're Walmart.

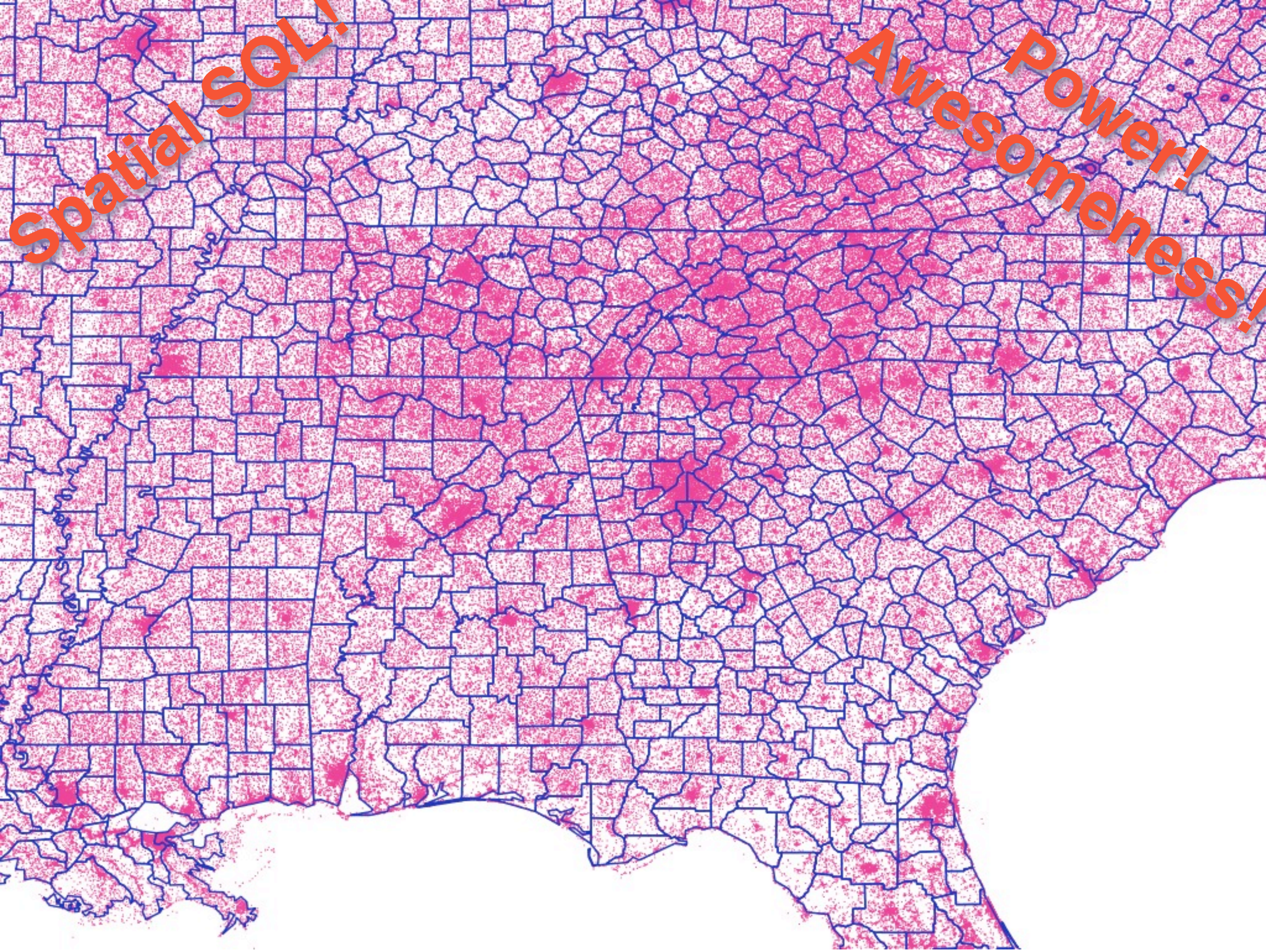
And suppose that, for marketing purposes, you want to know the income and education level of your customers.

You can't get that from the POS records. You could run a survey, but that would be expensive, and it would also be duplicative, because a survey has already been run a huge expensive survey...



Thursday, March 12, 15

called the US Census. But your customer information doesn’t have a foreign key, linking it to census tracts, how do you get census information onto your customer records?



Thursday, March 12, 15

You use the universal key, location! Every customer has a location, and that location falls within a census tract. So you can join customers to census using a spatial relationship condition!
How many lines of code should this take???

Spatial SQL!

Power! Awesomeness!

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

Thursday, March 12, 15

something like this,
which would spit out a table that added census attributes to customers,
and you could pipe into a statistical analysis to get your answer.

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 spatial type is fully integrated into the query planner.
Even the PostgreSQL gurus at PgCon didn’t know this.
(So I added this slide)
That means that complex multi-table queries like this one execute efficiently.

The Basics!

ST_Length(A)

ST_Distance(A, B)

ST_DWithin(A, B, r)

ST_Area(A)

ST_Intersects(A, B)

You can use

The Basics!

ST_AsText(A)

ST_AsBinary(A)

ST_GeomFromText()

ST_GeomFromBinary()

Fun Formats!

ST_AsGeoJSON()

ST_AsGML()

ST_AsKML()

ST_GeomFromGML()

ST_GeomFromKML()

Thanks to the work of Oslandia, PostGIS supports a veritable
zoo of XML and other hipster formats like JSON
for both output and input.

Geometry Construction!

ST_Buffer()

ST_MakeLine()

ST_Polygonize()

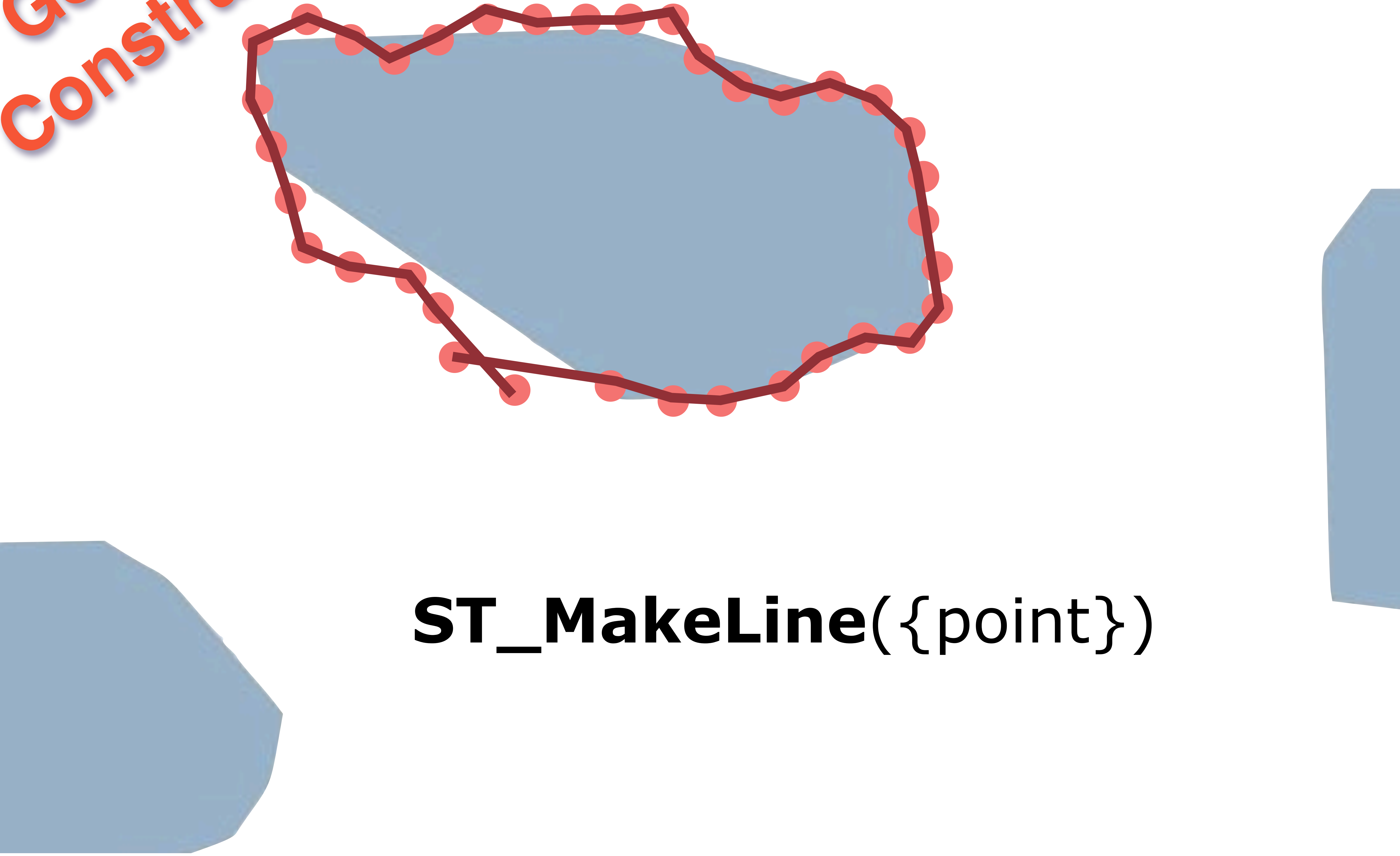
ST_BuildArea()

ST_Union()

Geometry Construction!

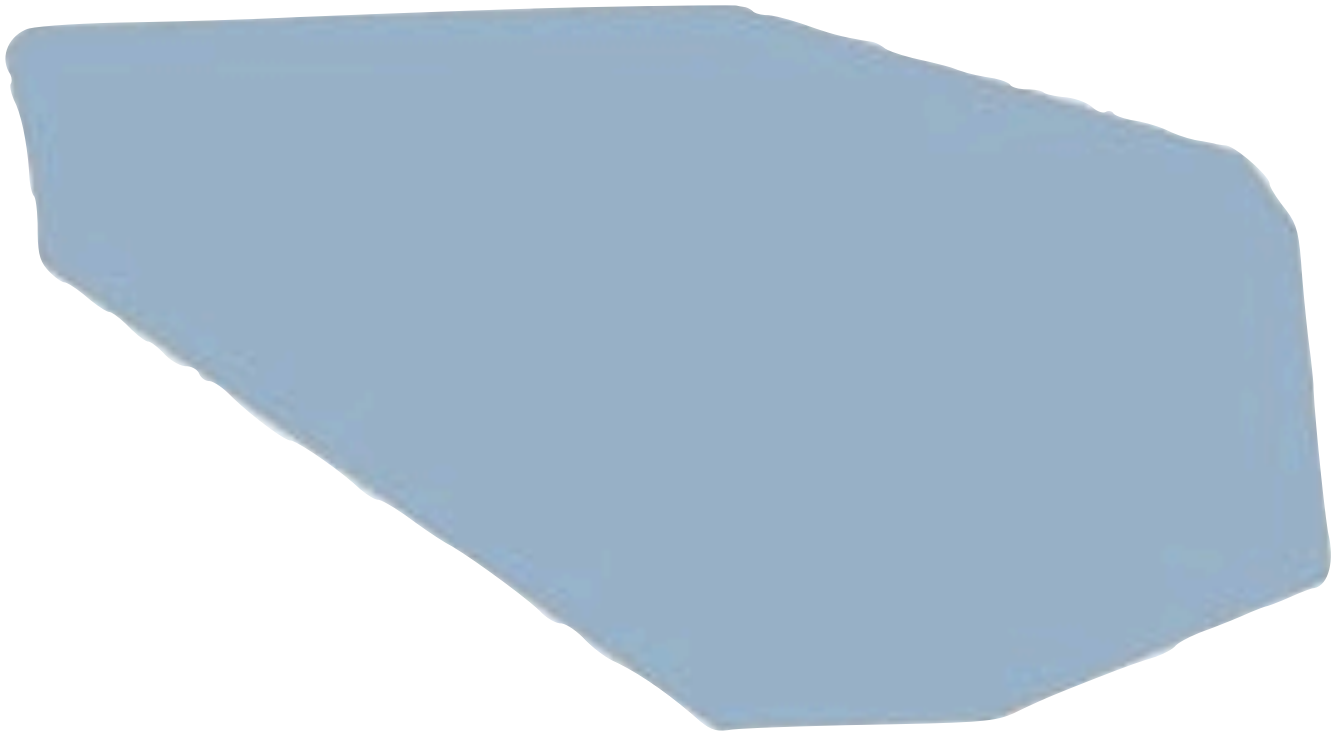
ST_MakeLine({point})

Geometry Construction!



ST_MakeLine({point})

Geometry Construction!

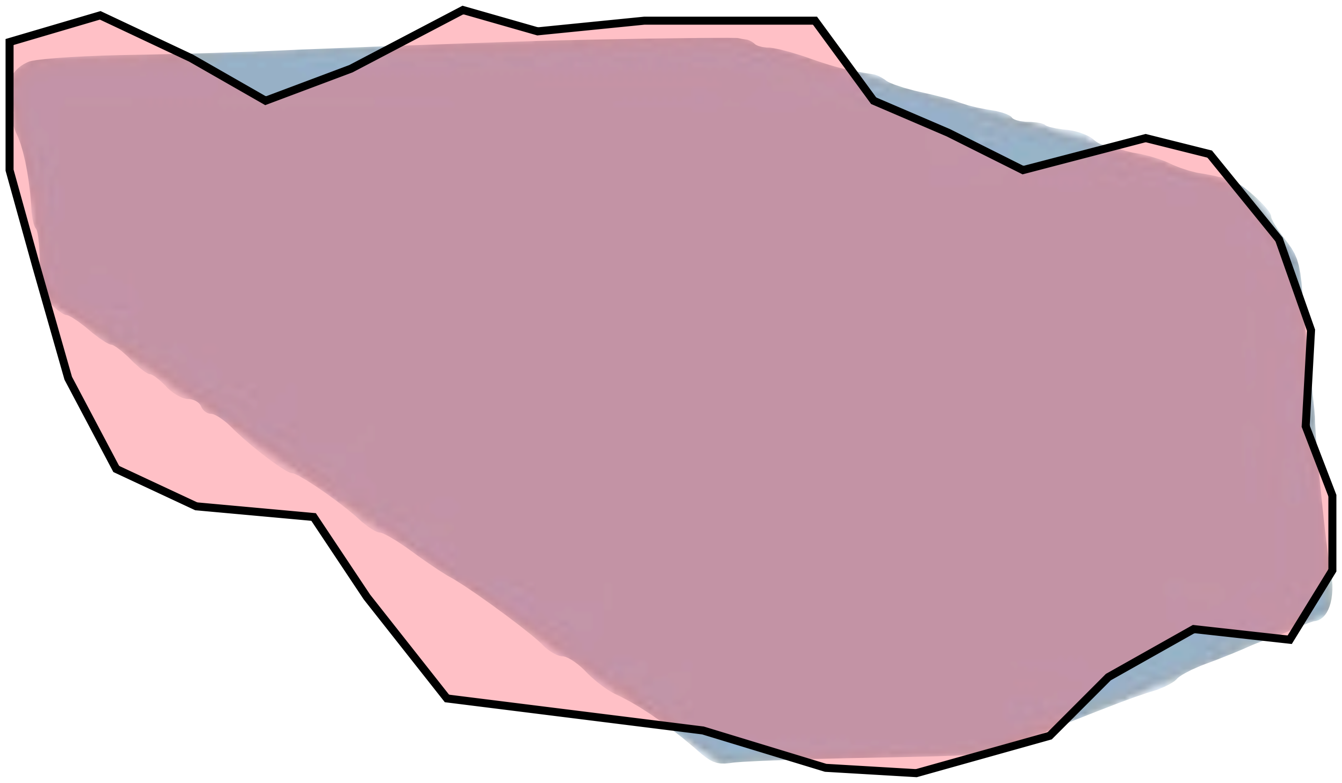


ST_BuildArea(multilinestring)

Thursday, March 12, 15

The ST_BuildArea() function takes in a multilinestring and builds the best area it can from it. Interior partitions are removed, and interior rings are respected as holes.

Geometry Construction!

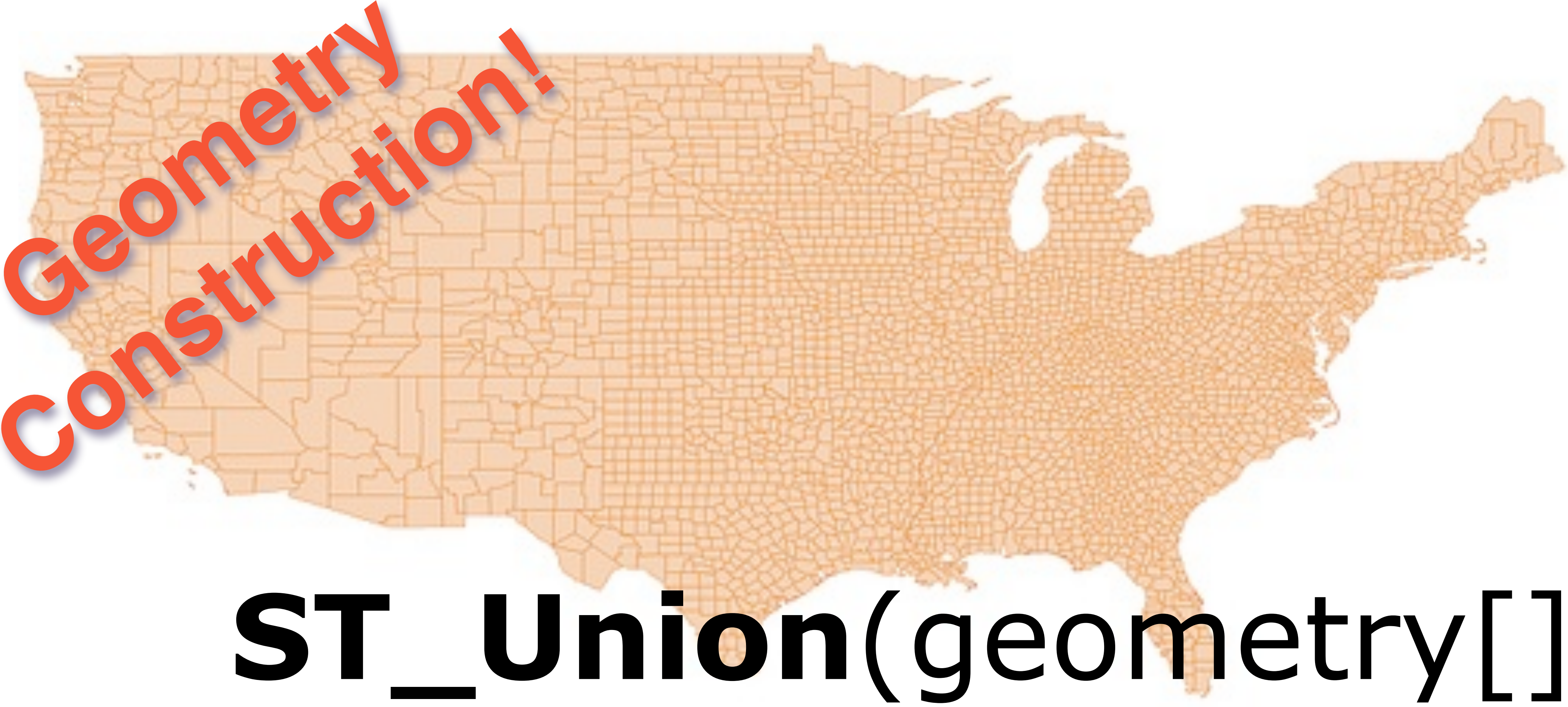


ST_BuildArea(multilinestring)

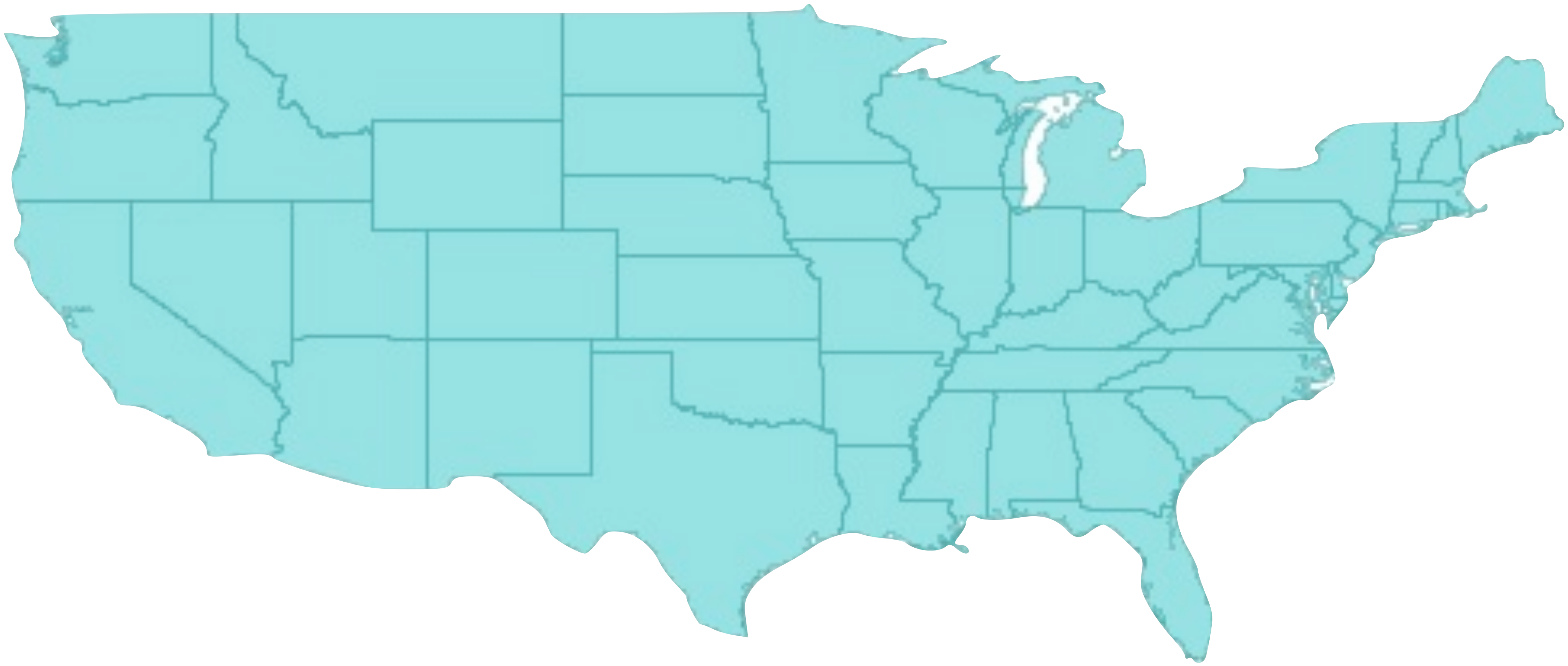
Thursday, March 12, 15

The ST_BuildArea() function takes in a multilinestring and builds the best area it can from it. Interior partitions are removed, and interior rings are respected as holes.

Geometry Construction!

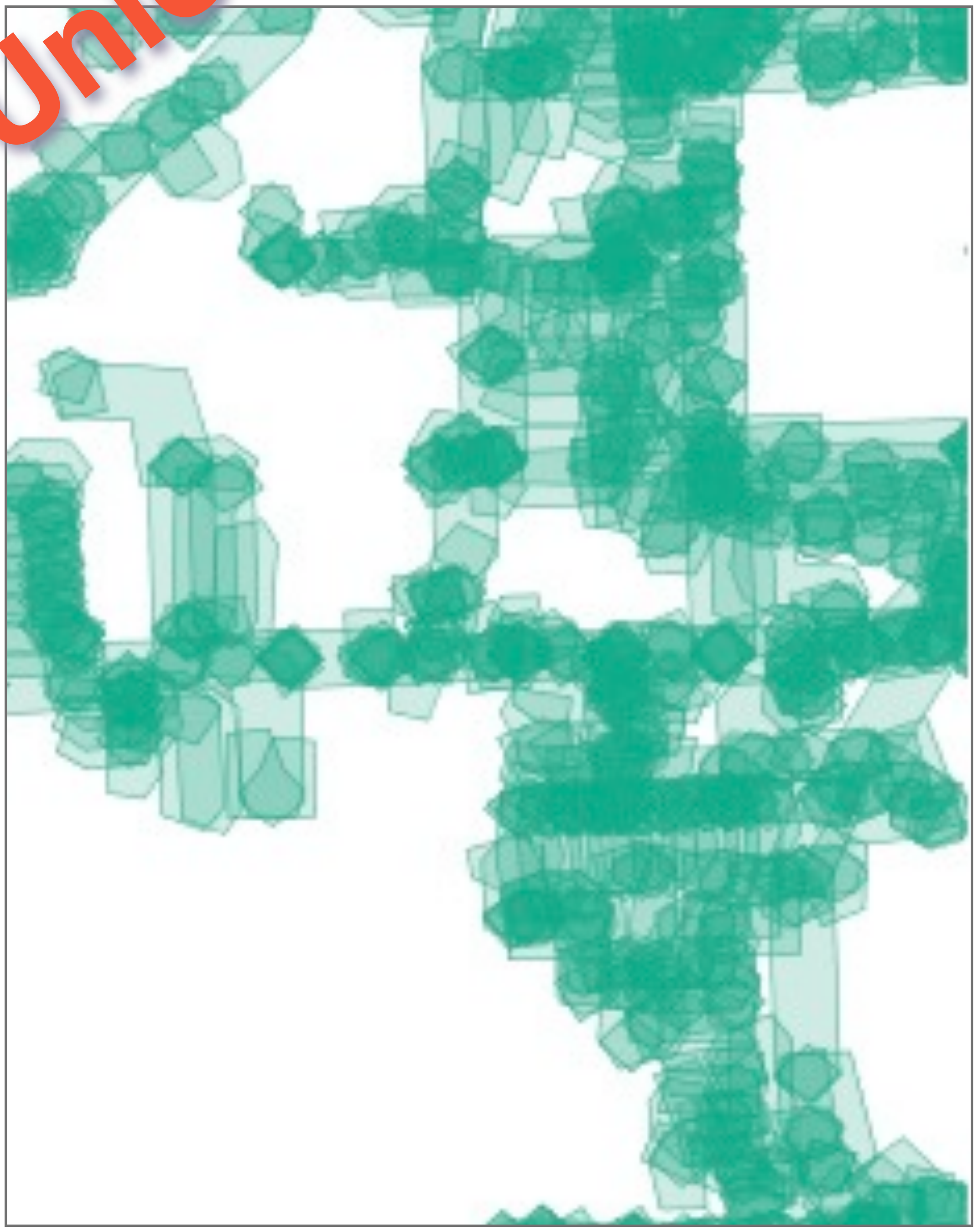


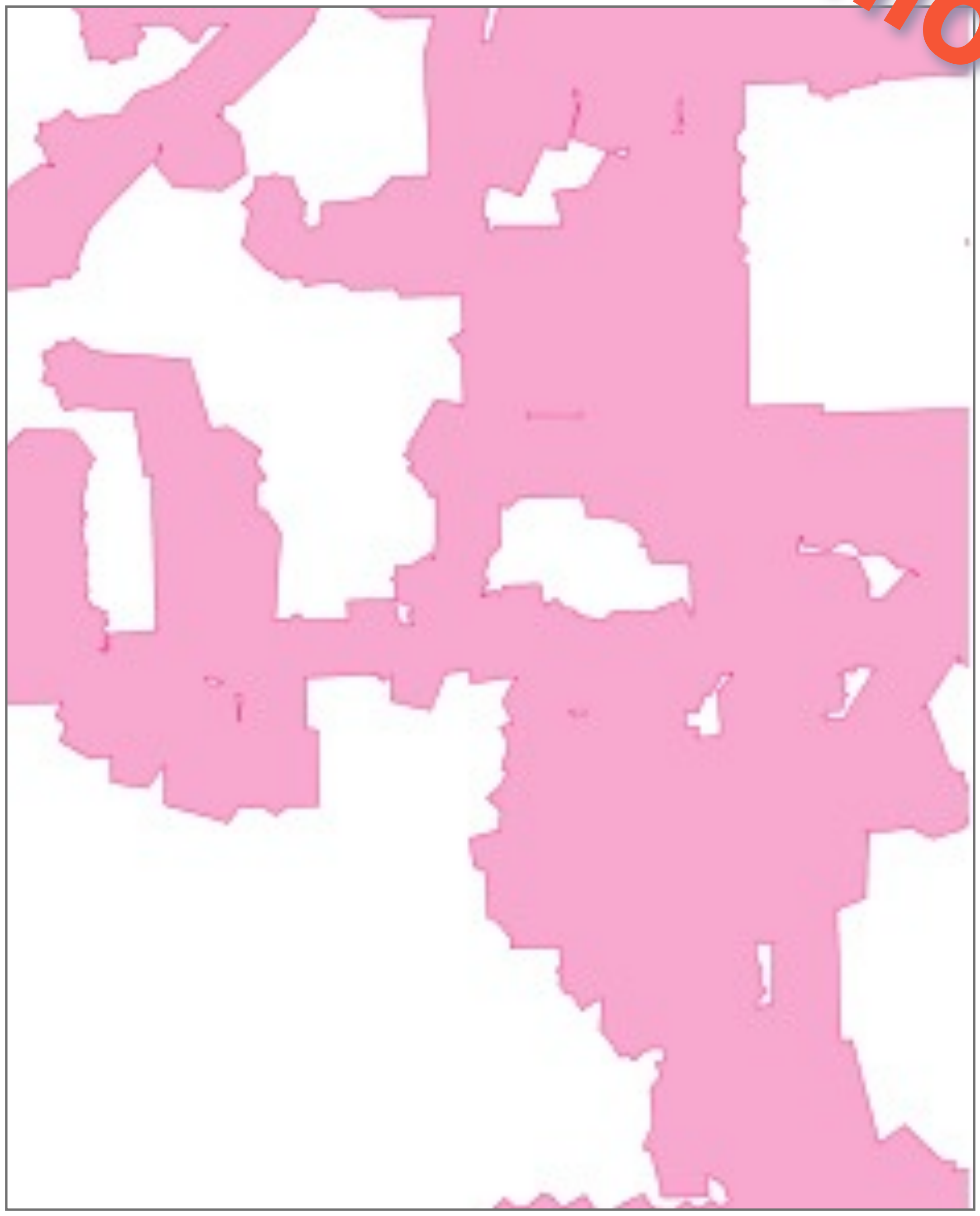
ST_Union(geometry[])



Cascaded Union!

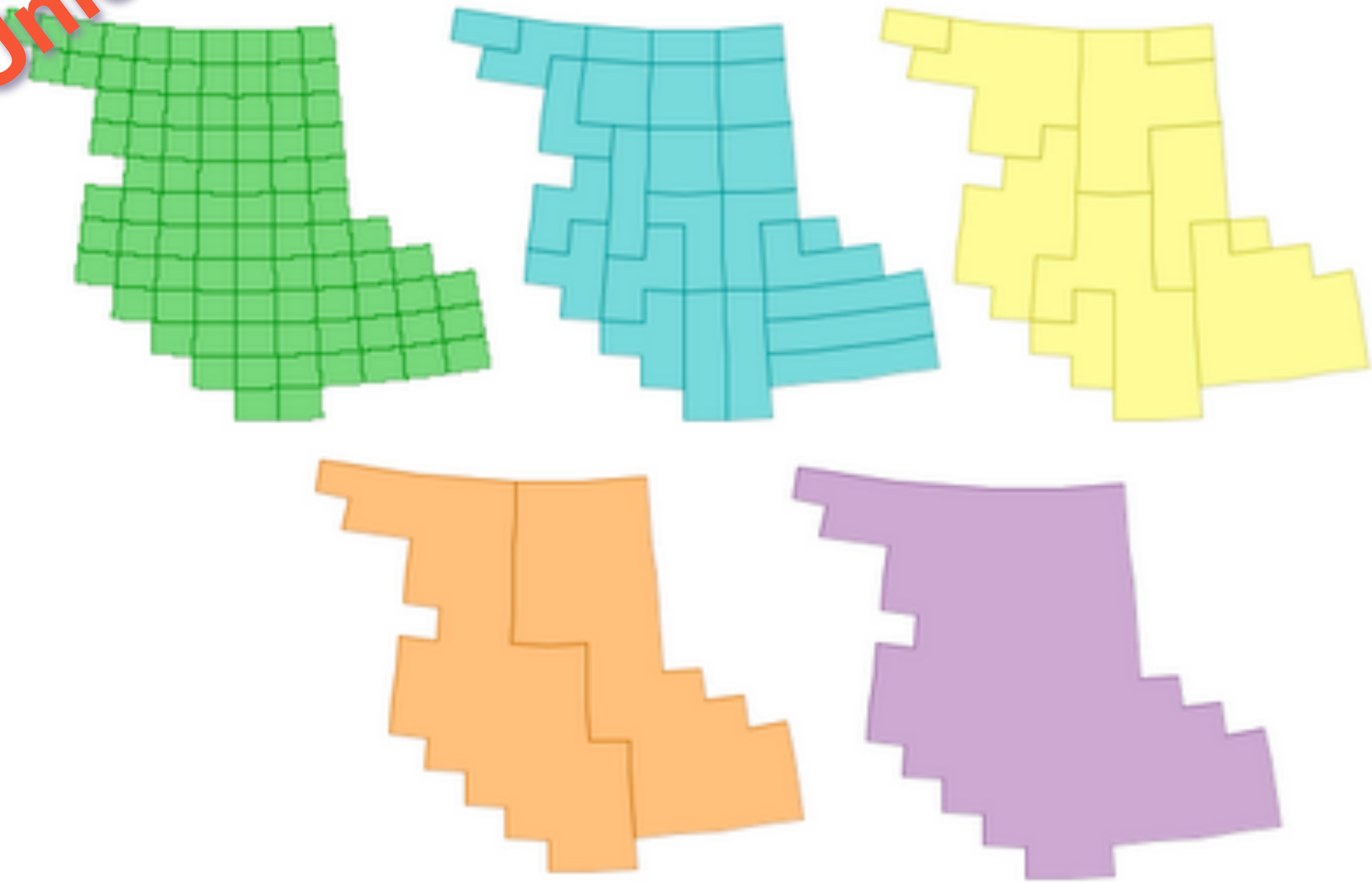
Not just union!





And our standard functions keep getting faster.
The cascaded union improvement came in version 1.4
The picture is the example we were sent:
“why is this operation so slow” he asked
Cascaded union merges polygons in the optimal order.
It made this example 40 times faster.

Cascaded Union!



Cascaded union does that melting process in a particular order
Merging neighbors first
This example became 5 times faster.

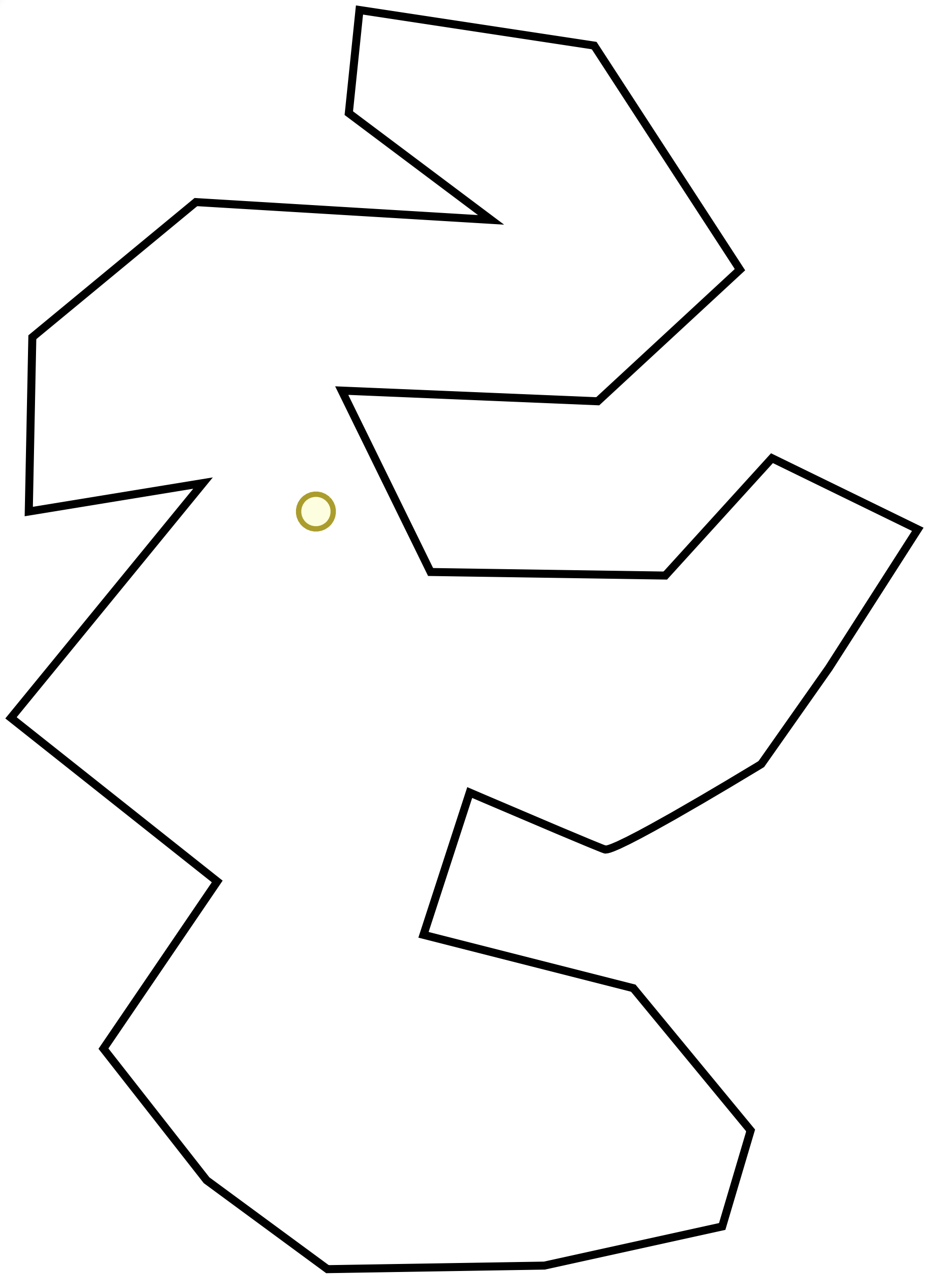
Prepared Geometry!

Not just Intersects!

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

Similarly, prepared geometries make standard spatial join queries faster.

Prepared Geometry!

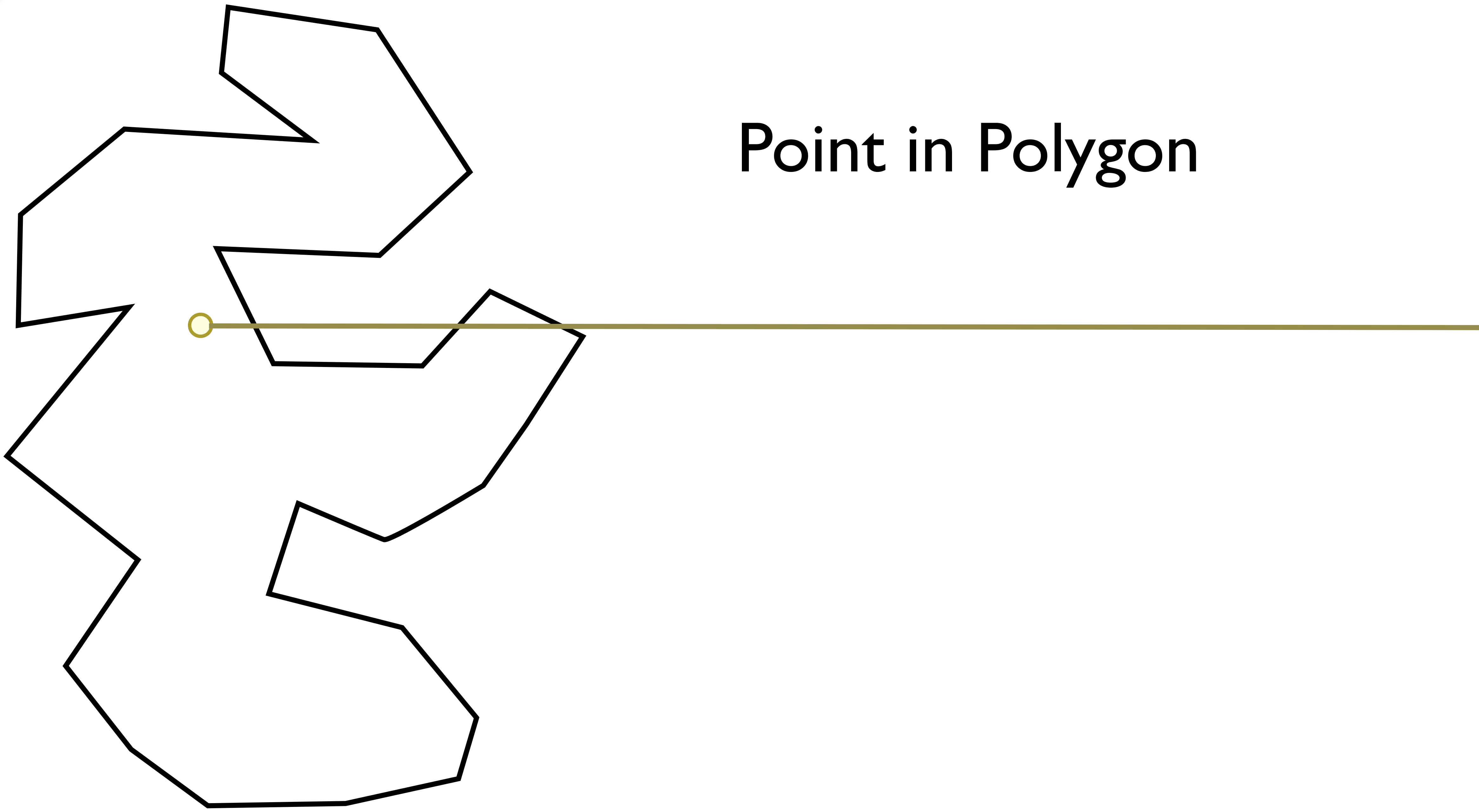


Thursday, March 12, 15

Calculating (for example) a point-in-polygon is quite expensive.

Prepared Geometry!

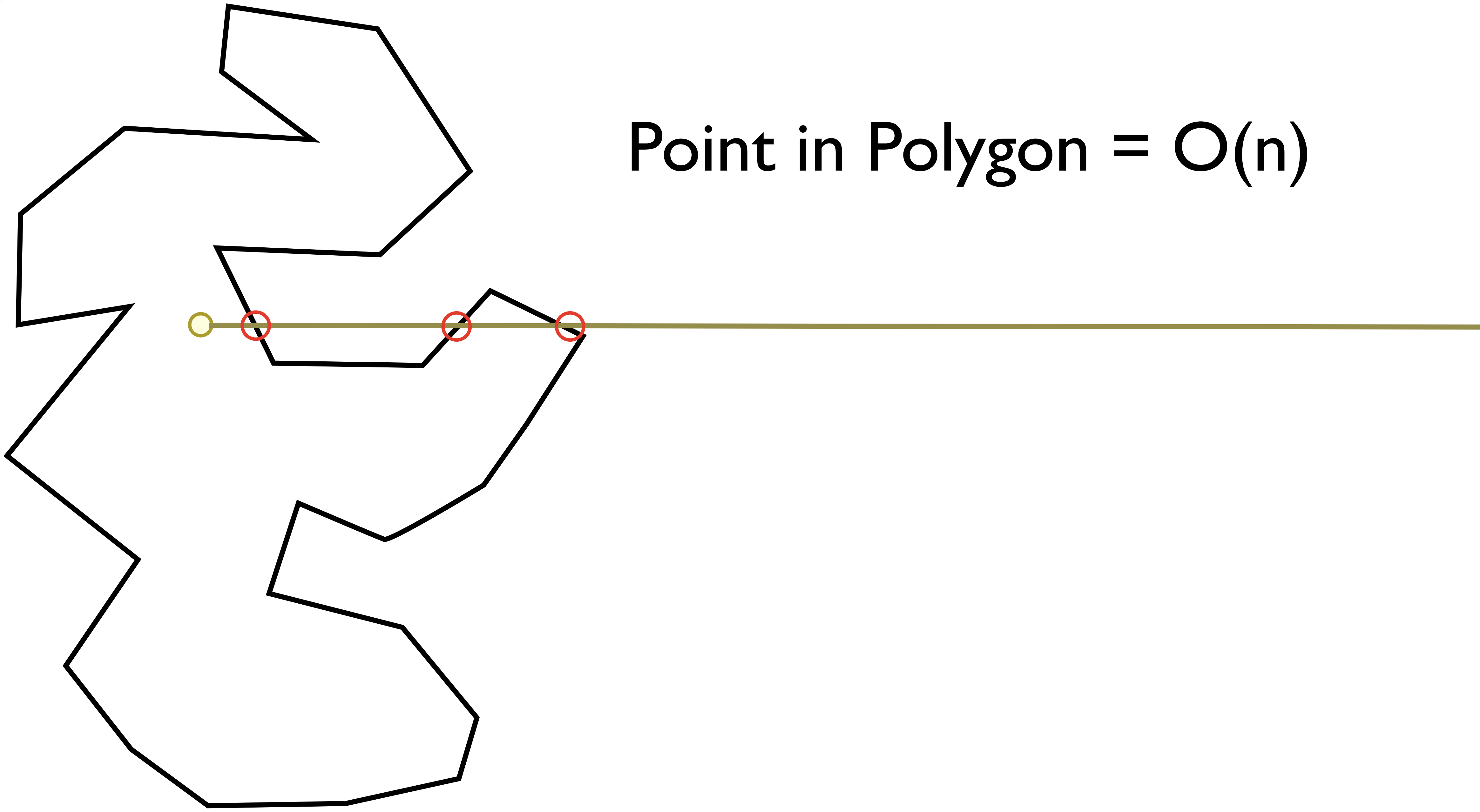
Point in Polygon



Thursday, March 12, 15

First you draw a stab line.
Then you check every edge to see if it hits the line.
Then you count up the number of hits.
Odd => inside.
Even => outside.

Prepared Geometry!



Point in Polygon = O(n)

Thursday, March 12, 15

This point has three hits, so it’s inside.
Calculate cost is proportional to the number of edges.

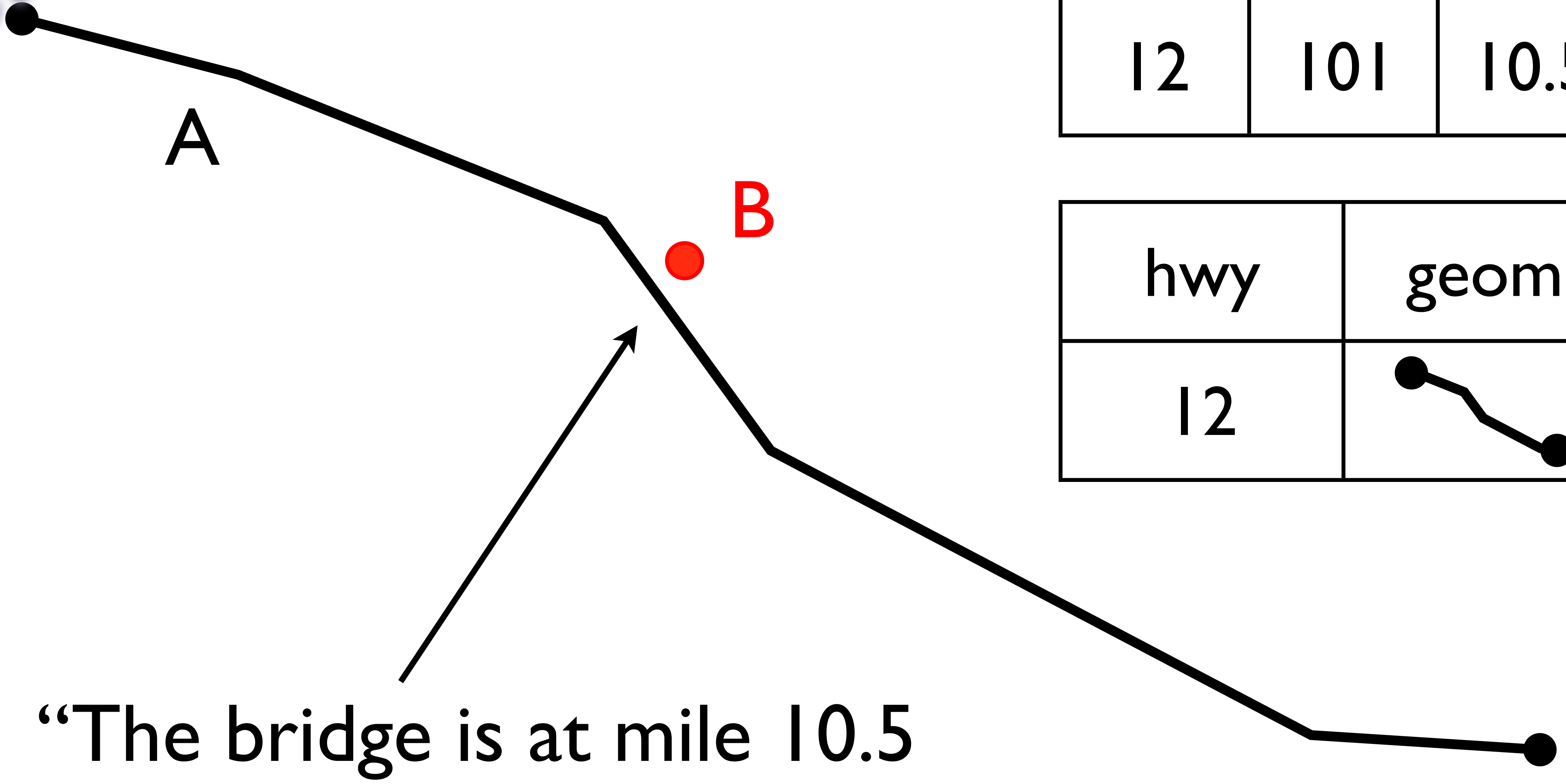
Prepared Geometry!

Prepared geometry makes repeated tests on large geometries very fast.

(ST_Intersects, ST_Contains)

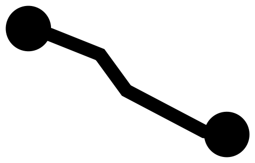
that’s fun geek talk
important to remember...
between 2 and 5 times faster depending on complexity of inputs
(more complex, means better improvement)

Linear Referencing!

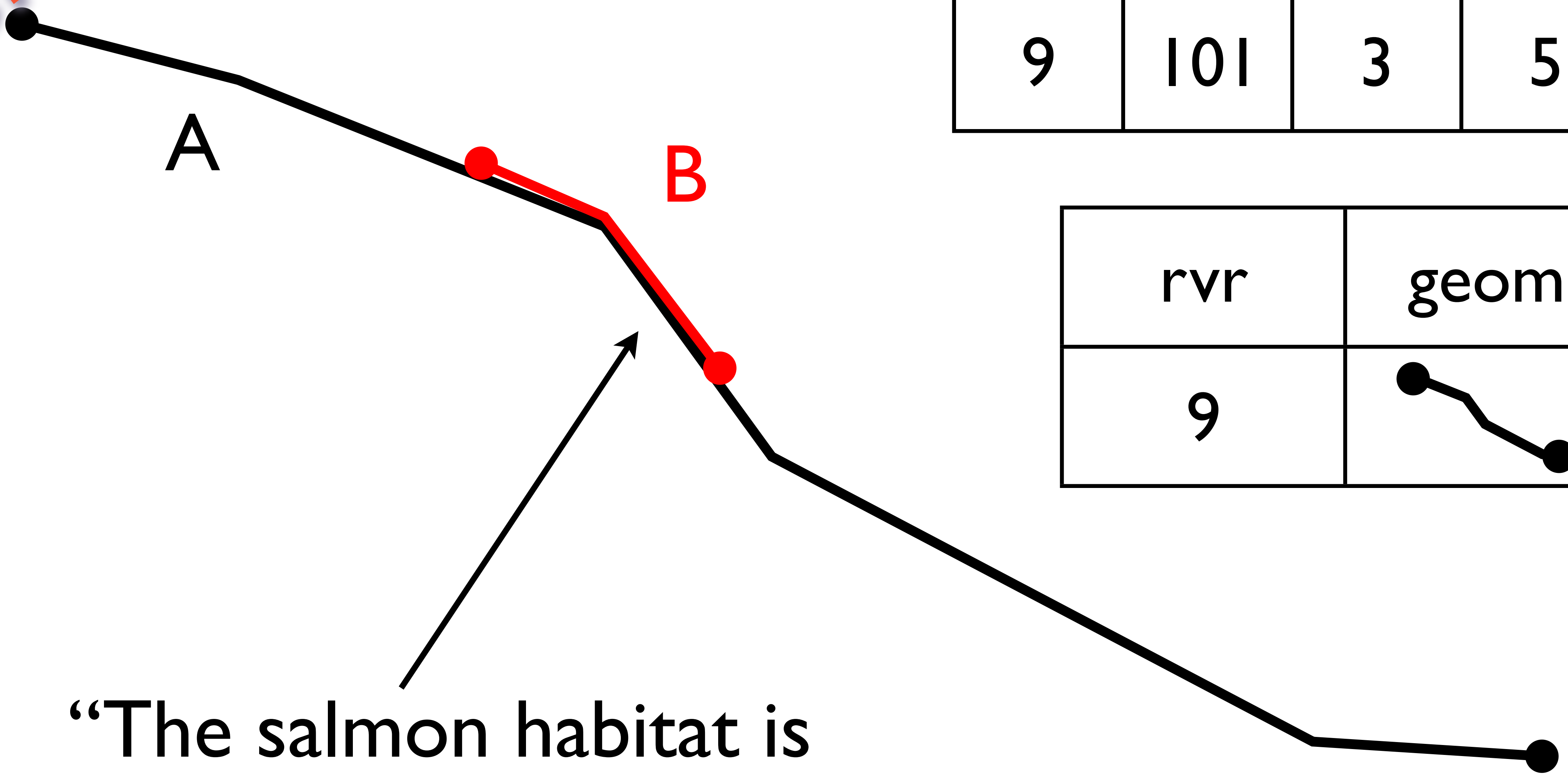


"The bridge is at mile 10.5 on Highway 12"

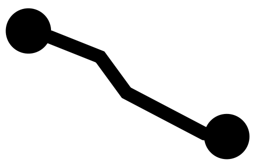
hwy	brdg	loc
12	101	10.5

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()

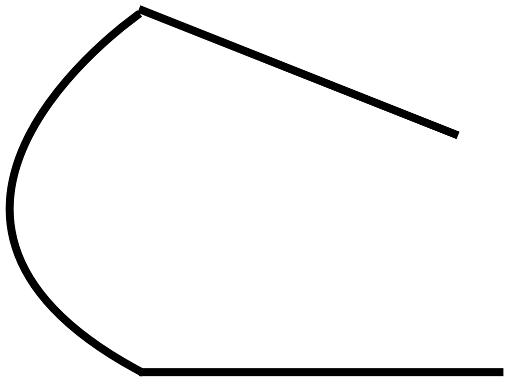
ST_Line_Locate_Point()

There is a set of linear referencing functions so you can build data models that include both measured and proportional linear references.

Curves!

- CURVESTRING

- COMPOUNDCURVE



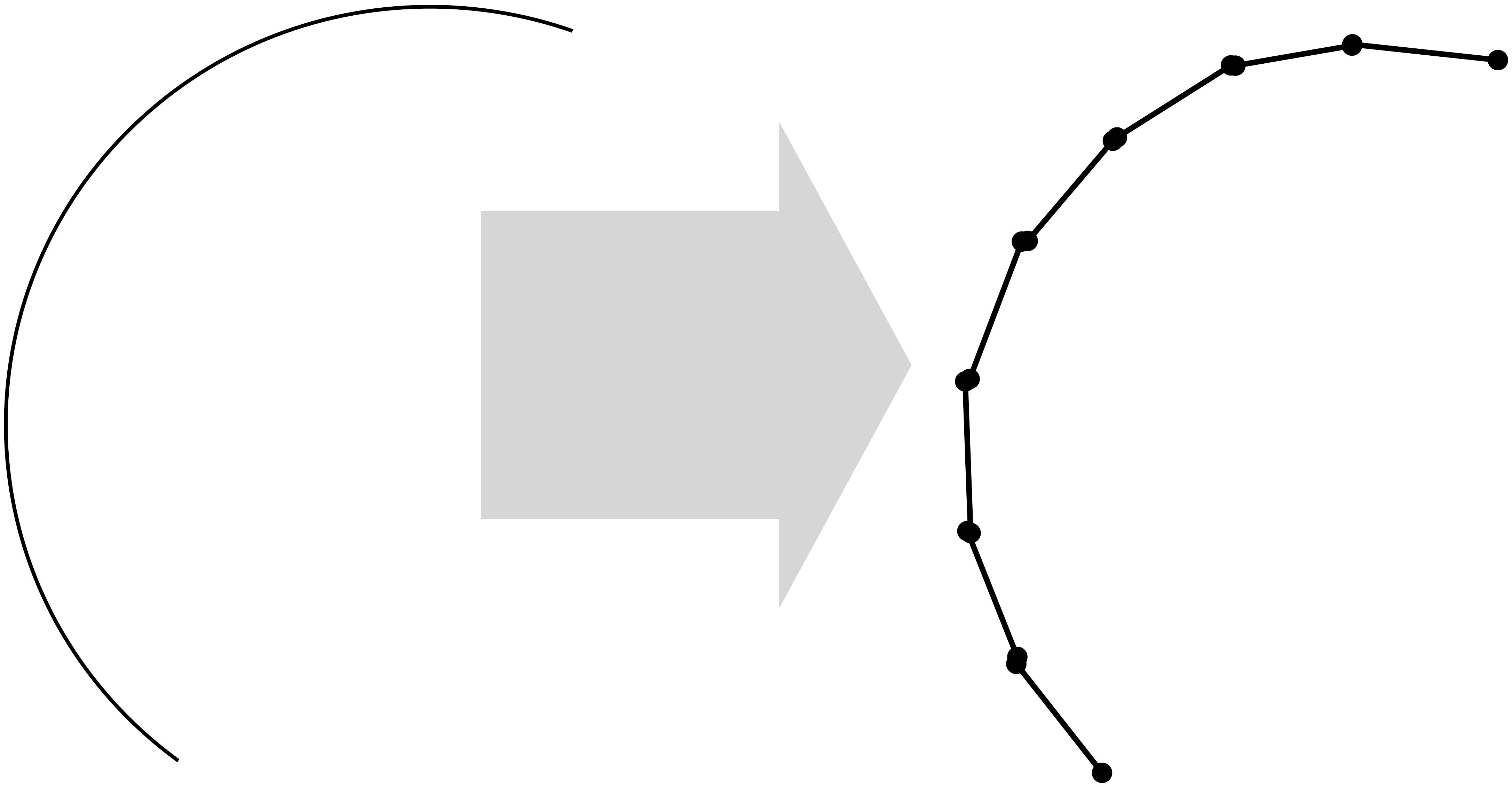
- CURVEPOLYGON

- ST_CurveToLine()
- **ST_LineToCurve()**

Since 1.2, we’ve had curve types
which are part of ISO SQL/MM standard
And our curve support has been getting more complete with each release.
You can convert curves to linestrings, and even convert linestrings to curves!
Curve types are useful for storing CAD data, which uses curves

Curves!

ST_CurveToLine()!

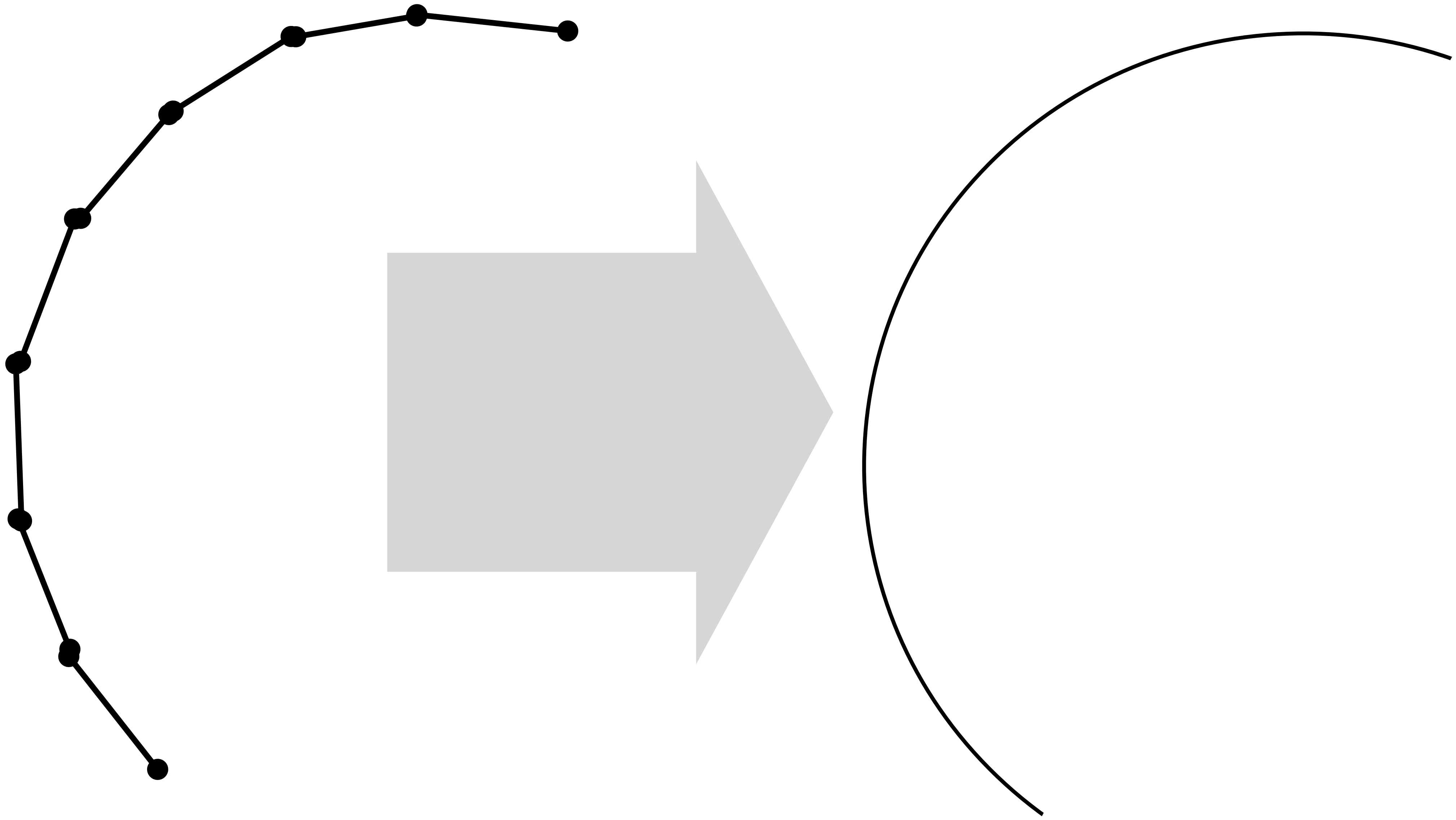


Thursday, March 12, 15

Since 1.2, we’ve had curve types
which are part of ISO SQL/MM standard
And our curve support has been getting more complete with each release.
You can convert curves to linestrings, and even convert linestrings to curves!
Curve types are useful for storing CAD data, which uses curves

Curves!

ST_LineToCurve()!



Thursday, March 12, 15

Since 1.2, we’ve had curve types
which are part of ISO SQL/MM standard
And our curve support has been getting more complete with each release.
You can convert curves to linestrings, and even convert linestrings to curves!
Curve types are useful for storing CAD data, which uses curves

Reprojection

ST_Transform()

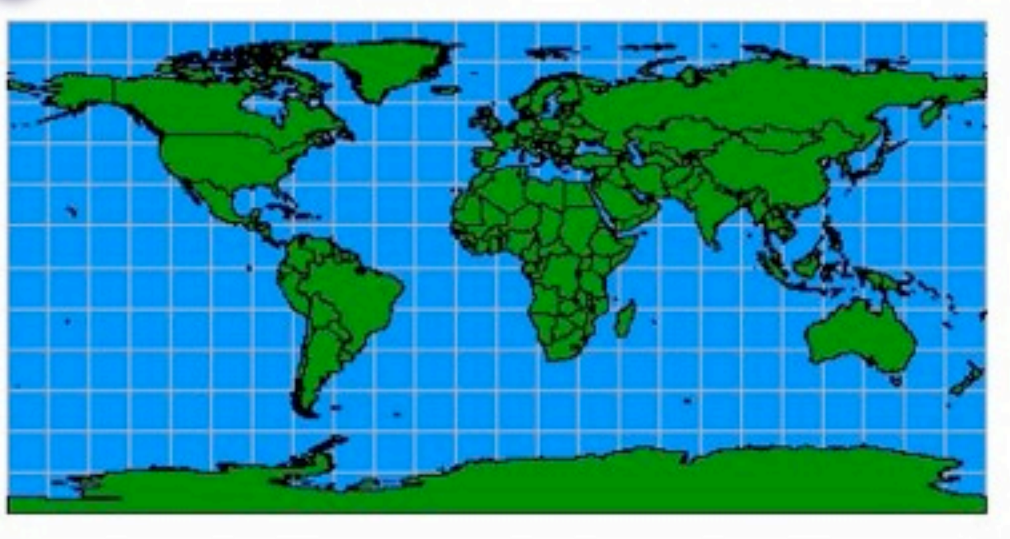
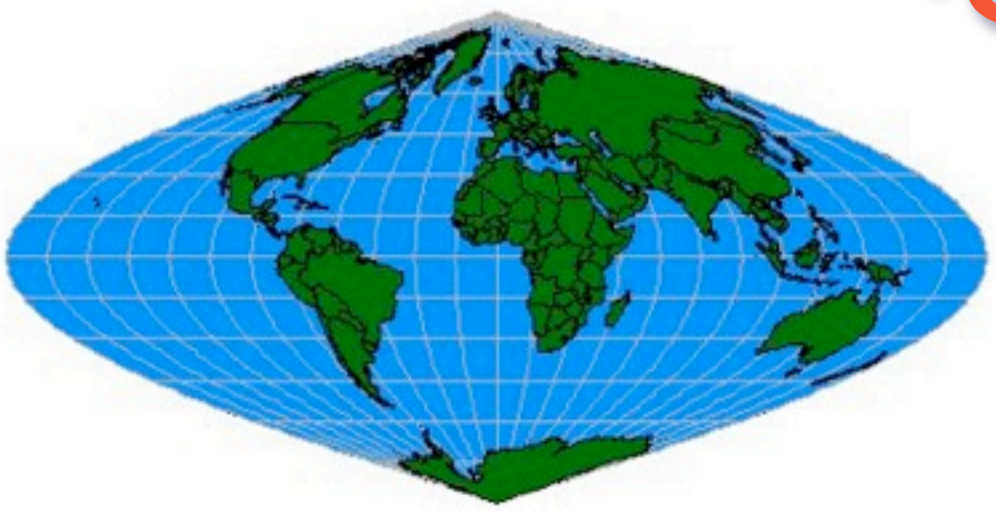
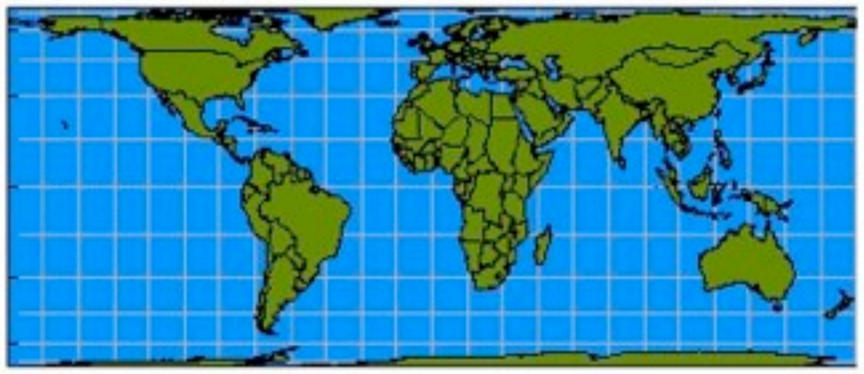


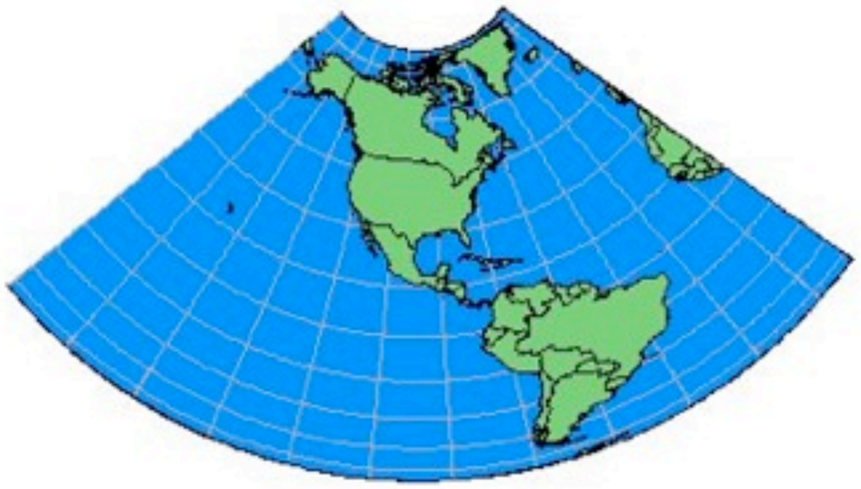
Plate Carree Projection



Sinusoidal Projection



Behrmann Projection



Albers Equal Area Conic Projection

Reprojection

ST_Transform()

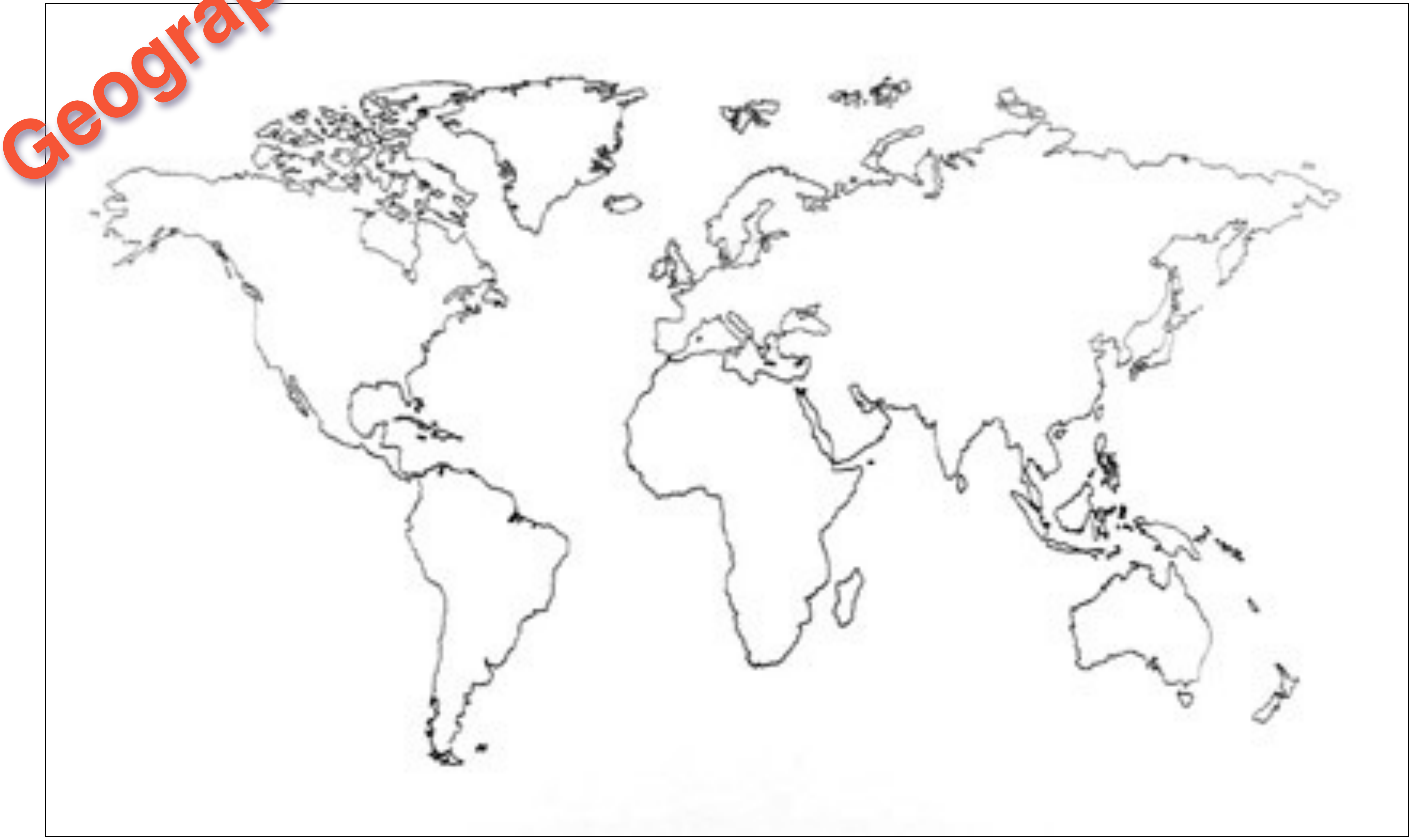
- Albers
- Lambert
- Mercator
- Sinusiodal
- Stereographic
- UTM
- Gnomic
- Orthographic

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

Geography!

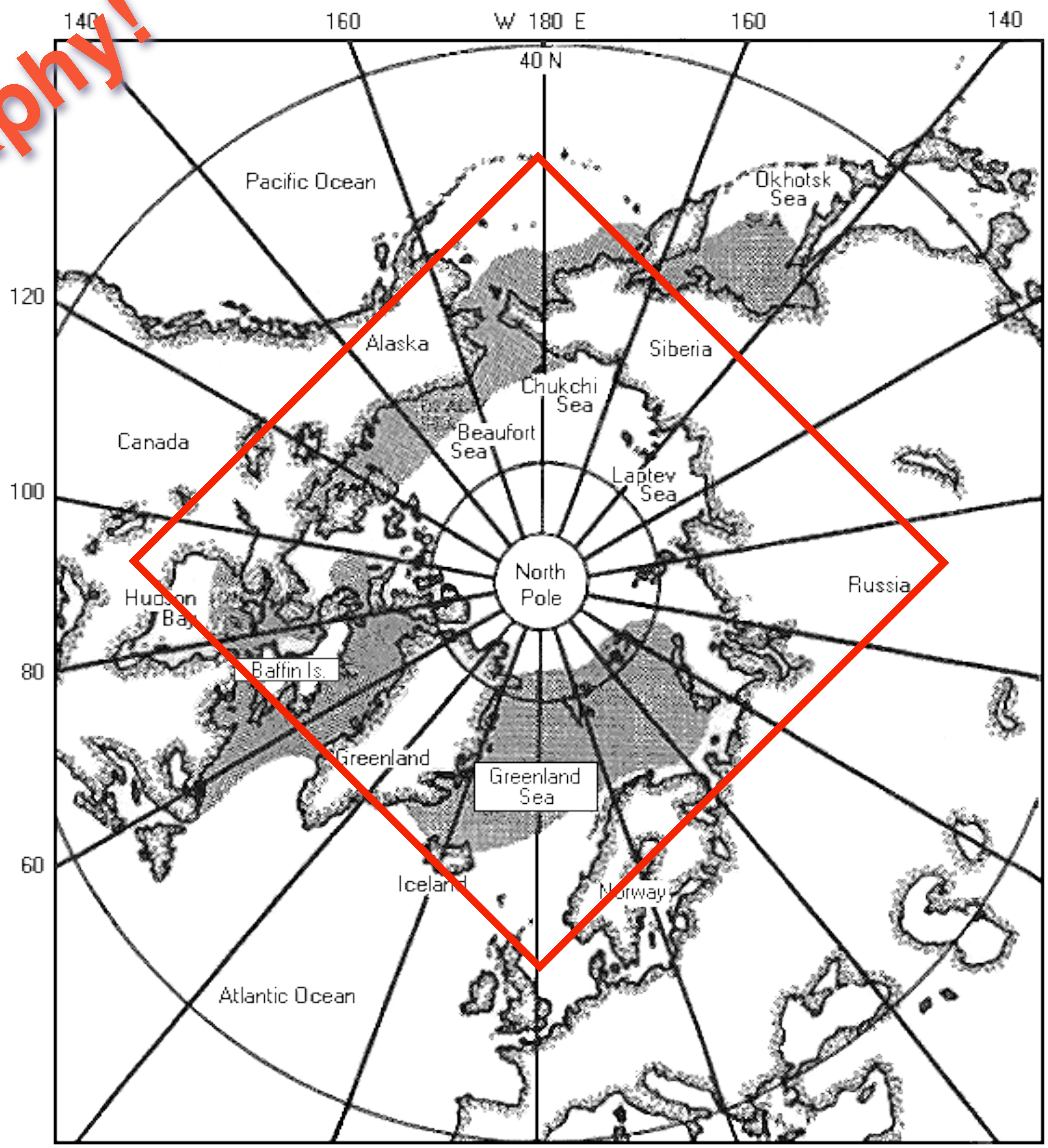
In 1.5, we added the geography type models data on a sphere

Geography!



“geometry” type models data on a plane
but that Simple Plate-Carre view, has lots of problem cases.

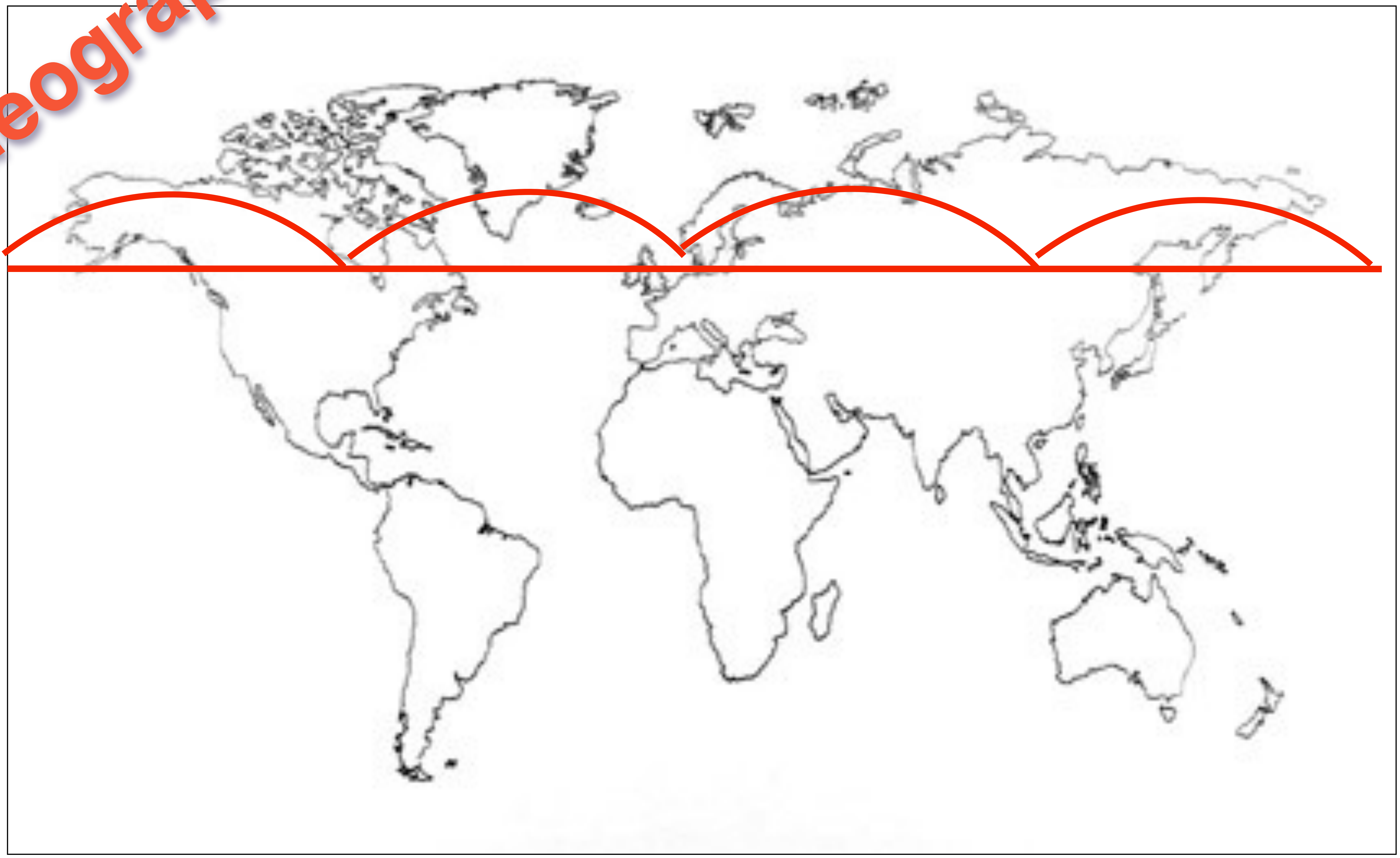
Geography!



Thursday, March 12, 15

a polygon that covers the pol

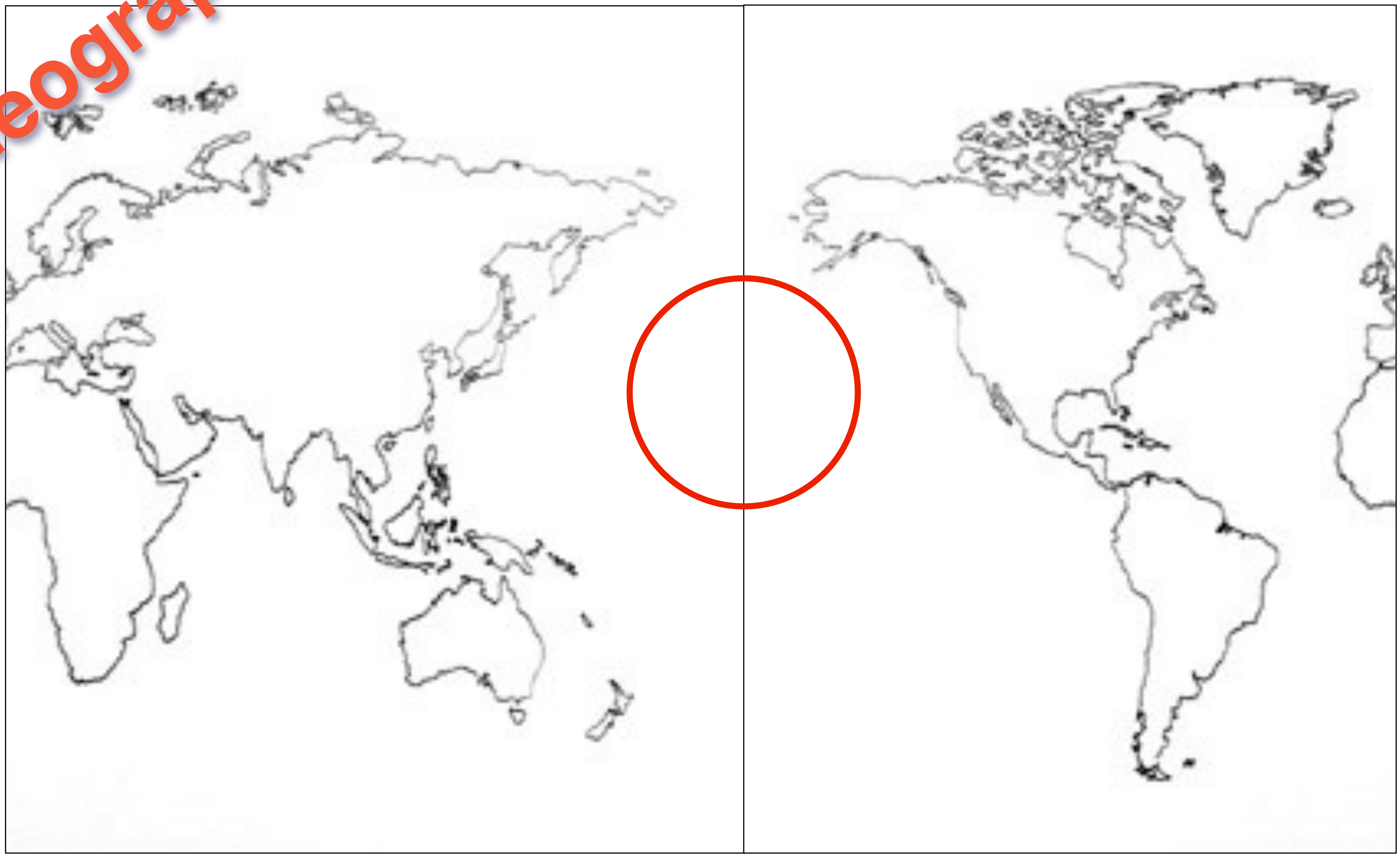
Geography!



Thursday, March 12, 15

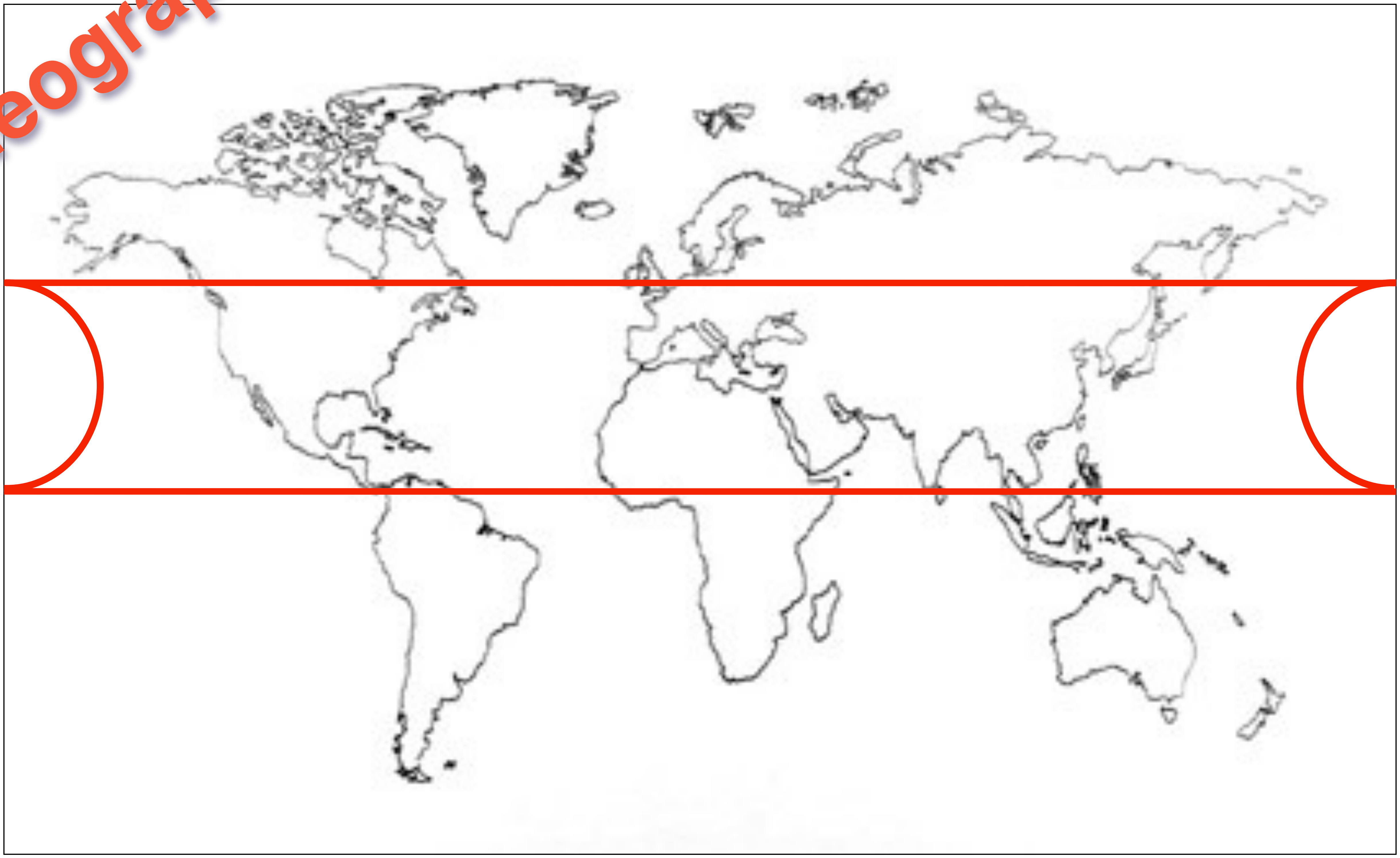
is wrong if you interpret it on the plane

Geography!



a polygon that crosses the dateline

Geography!

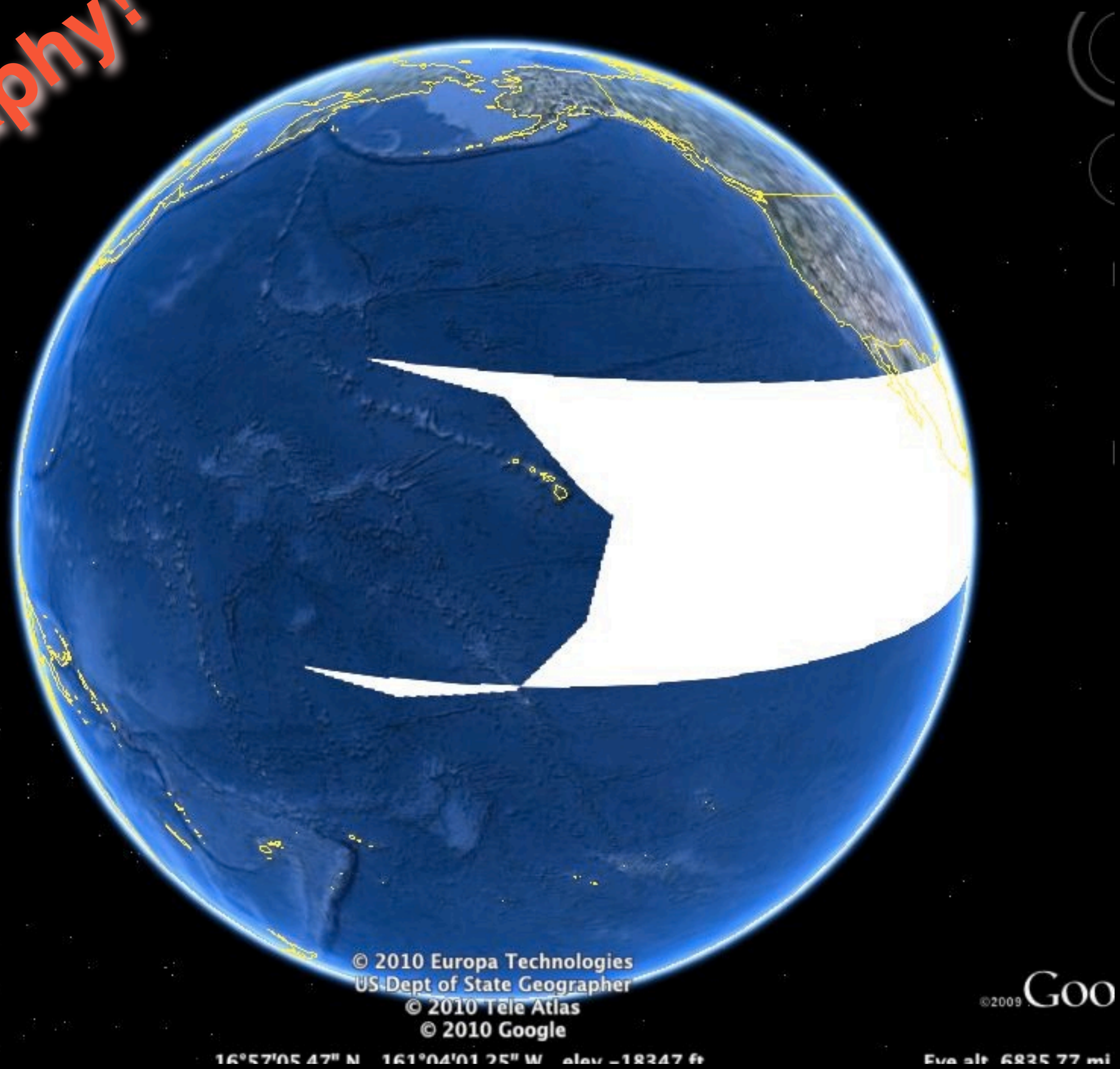


Thursday, March 12, 15

thinks it crosses the whole world if you interpret it on the plane

this wasn’t just a problem for postgis

Geography!


© 2010 Europa Technologies
US Dept of State Geographer
© 2010 Tele Atlas
© 2010 Google
©2009 Goo
16°57'05.47" N 161°04'01.25" W elev -18347 ft
Eye alt 6835.77 mi

even google earth has a hard time with the dateline and polygons
also the poles

Geography!

Who is geography for?

There two kinds of users who find GEOGRAPHY useful
the first is “geonewbies”, users who do not know any GIS or anything about map projections
GEOGRAPHY lets them work with lat/lon data without knowing about projections

Geography!

GeoNewbies

Who is geography for?

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

There two kinds of users who find GEOGRAPHY useful
the first is “geonewbies”, users who do not know any GIS or anything about map projections
GEOGRAPHY lets them work with lat/lon data without knowing about projections

Geography!

Who is geography for?

Thursday, March 12, 15

The “geohugies” are really big organizations that have truly global data, that covers the poles and datelines and everything in between there is no map projection that works for them

Geography!

GeoHugies

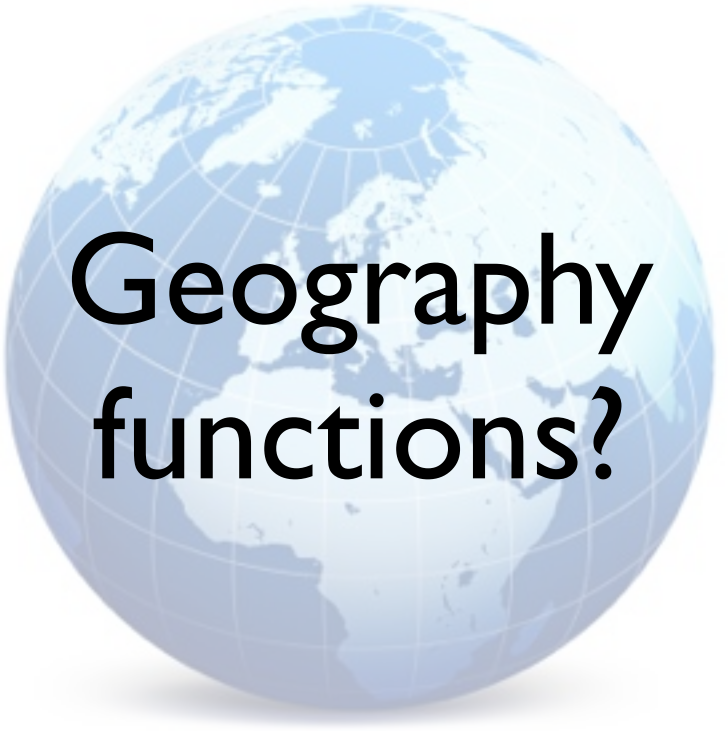
Who is geography for?

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

NASA
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE
NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY
UNITED STATES NAVY

The "geohugies" are really big organizations that have truly global data, that covers the poles and datelines and everything in between there is no map projection that works for them

Geography!

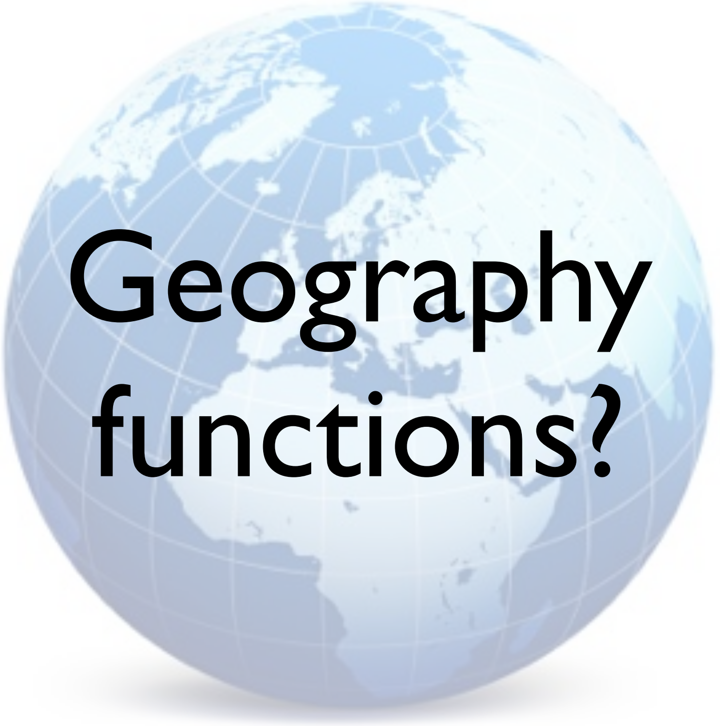


Geography functions?

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!

- Indexes spherical data

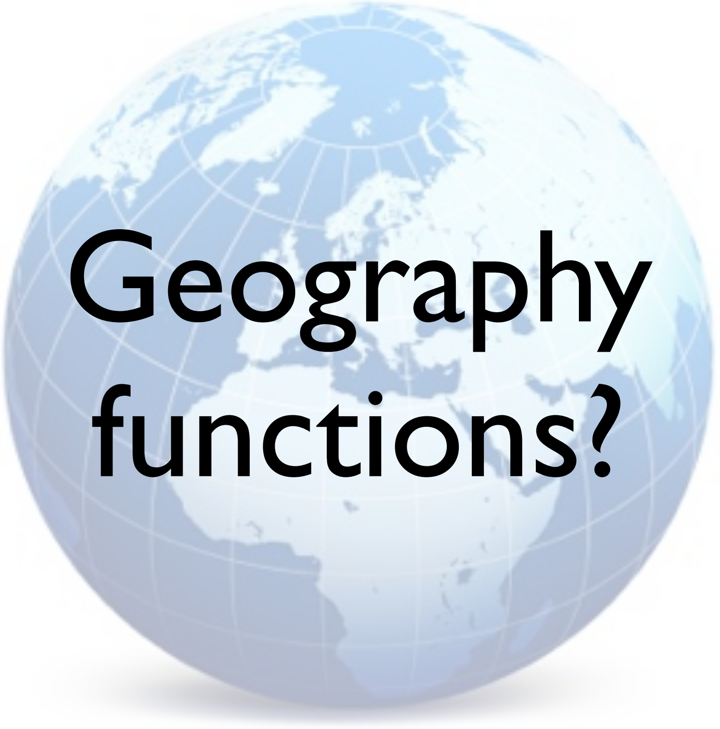


Geography functions?

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!

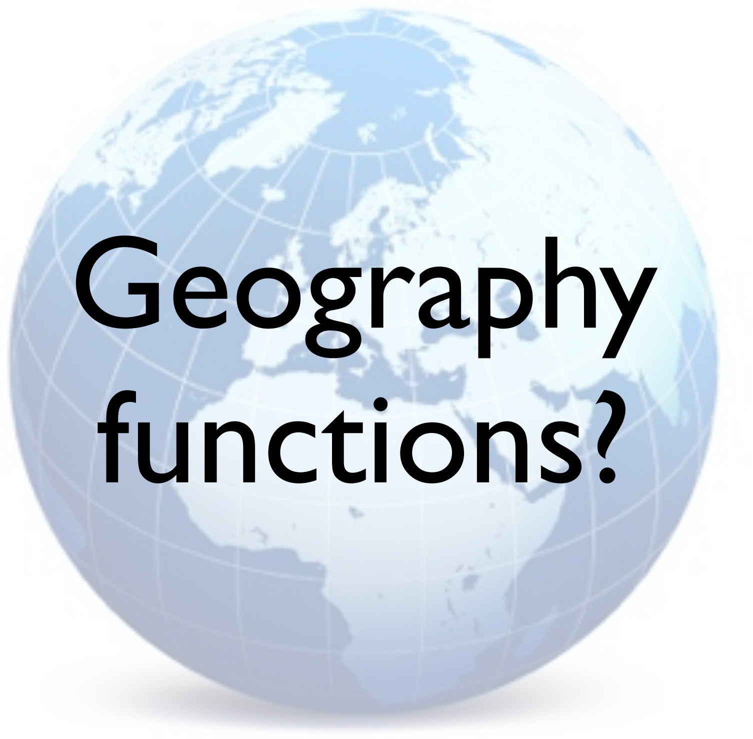
- Indexes spherical data
- ST_Intersects()



Geography functions?

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!

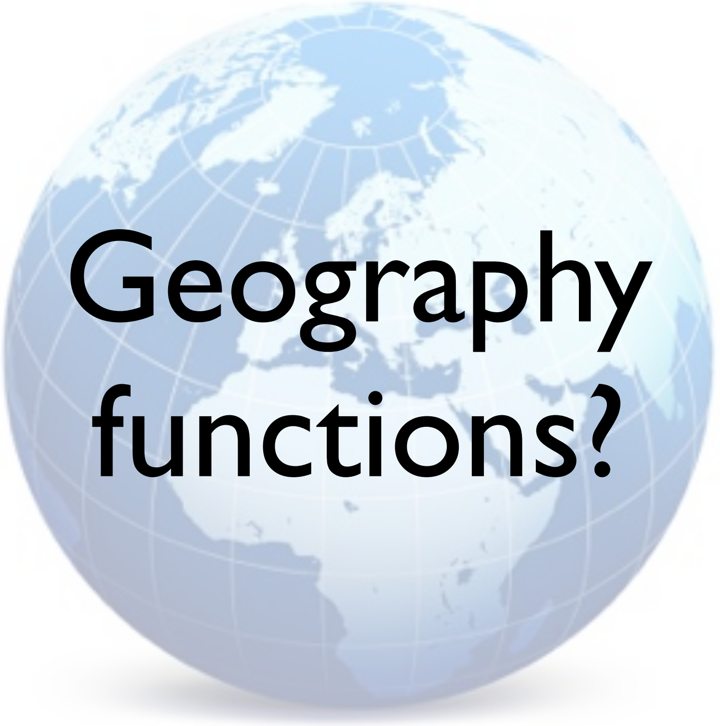


Geography functions?

- Indexes spherical data
- ST_Intersects()
- ST_Distance()

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!

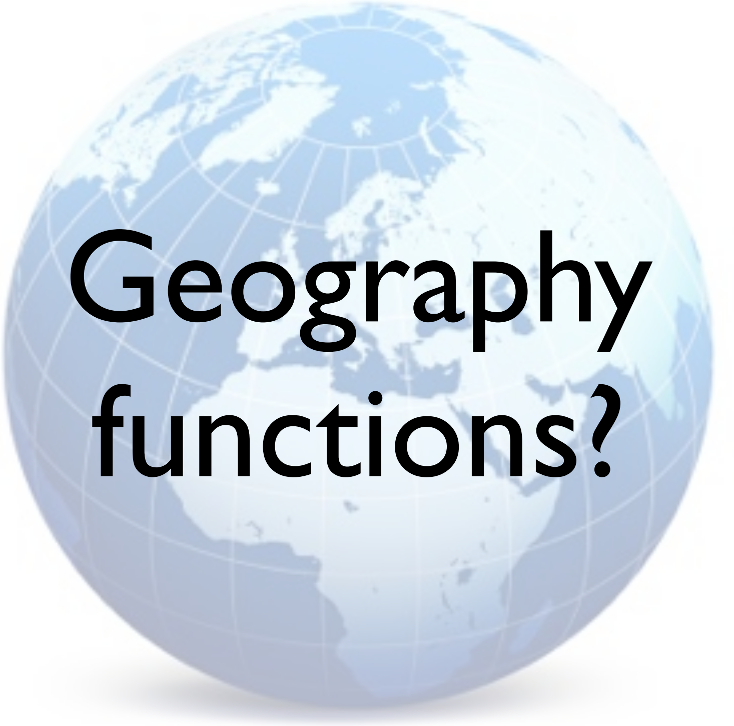


Geography functions?

- Indexes spherical data
- ST_Intersects()
- ST_Distance()
- ST_DWithin()

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!



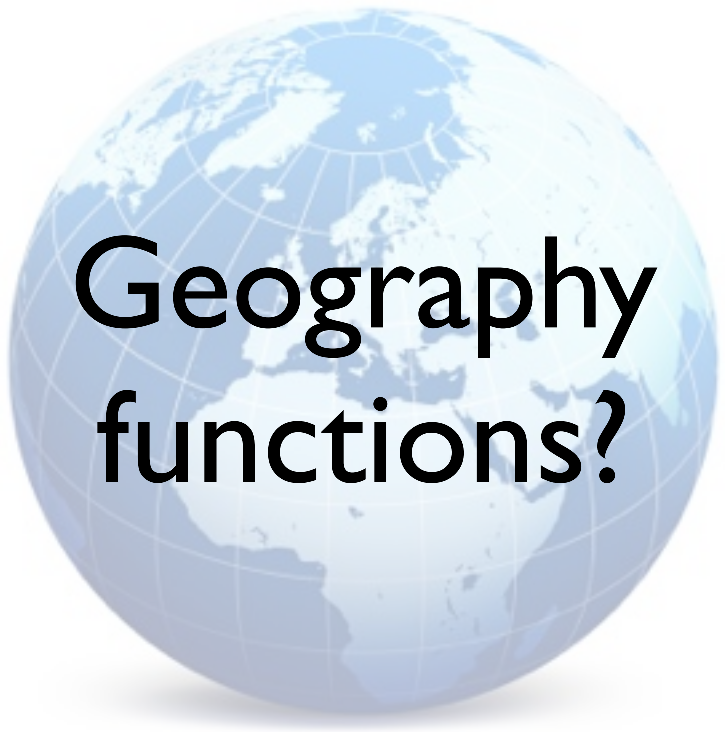
Geography functions?

- Indexes spherical data
- ST_Intersects()
- ST_Distance()
- ST_DWithin()
- ST_Area()

Thursday, March 12, 15

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!



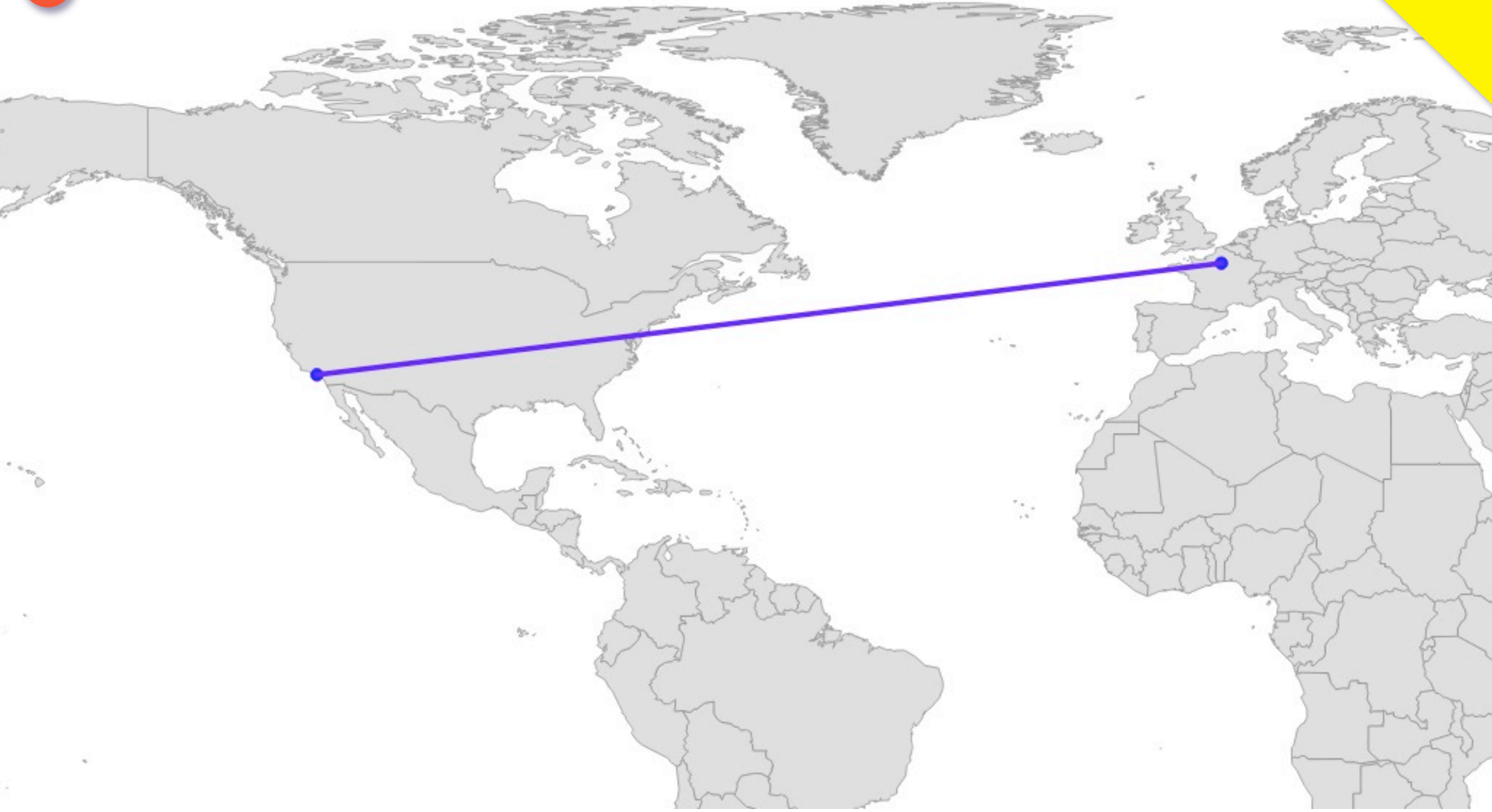
Geography functions?

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

the implementation of GEOGRAPHY currently only has support for a few functions
but you can use casts to convert to GEOMETRY and access all the geometry function

Geography!

New to 2.1!

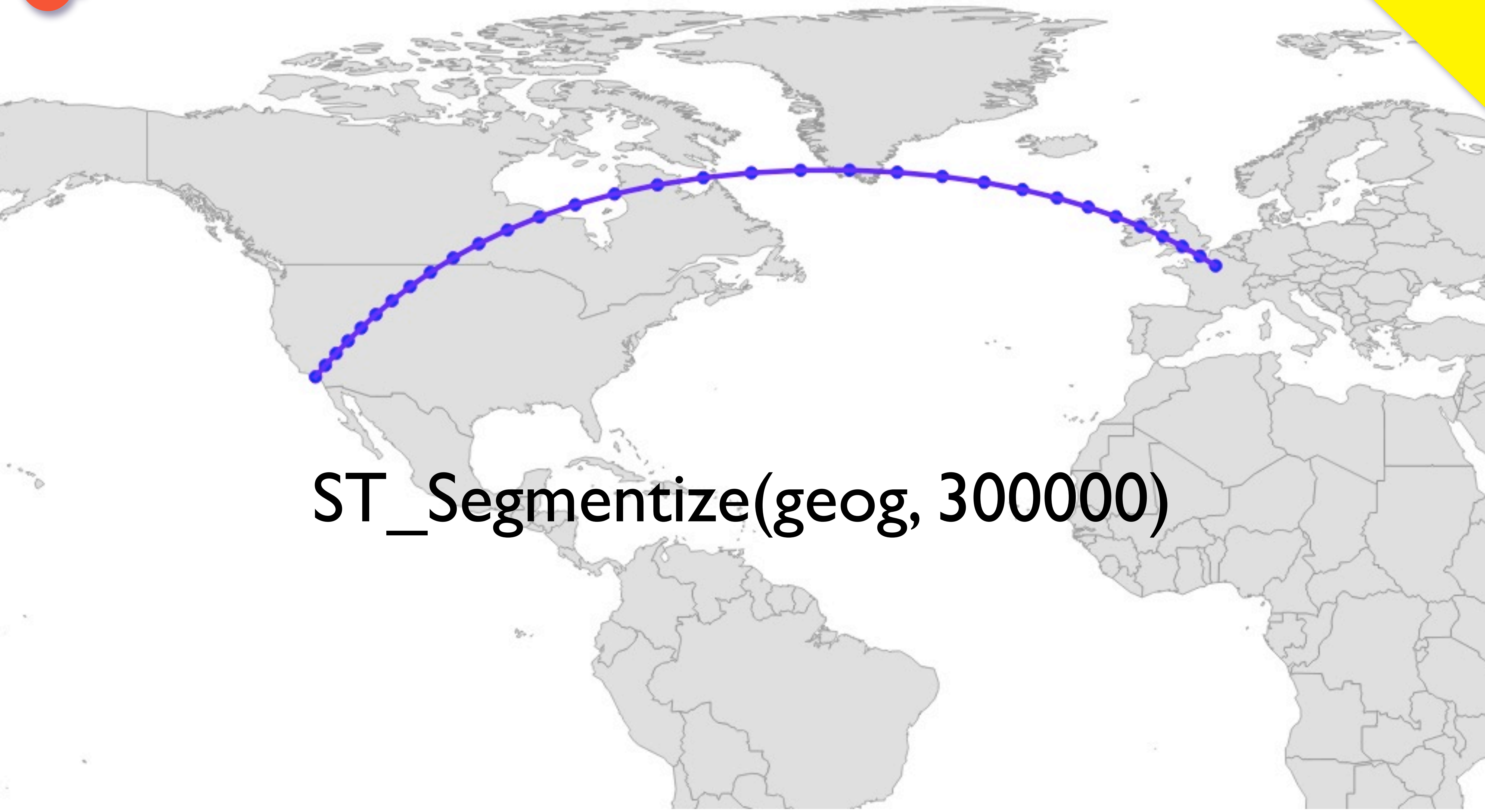


Thursday, March 12, 15

There’s a handy new geography function in 2.1, which takes a line, like this one that only has two vertices, on in Paris and one in Los Angeles,

Geography!

New to 2.1!



ST_Segmentize(geog, 300000)

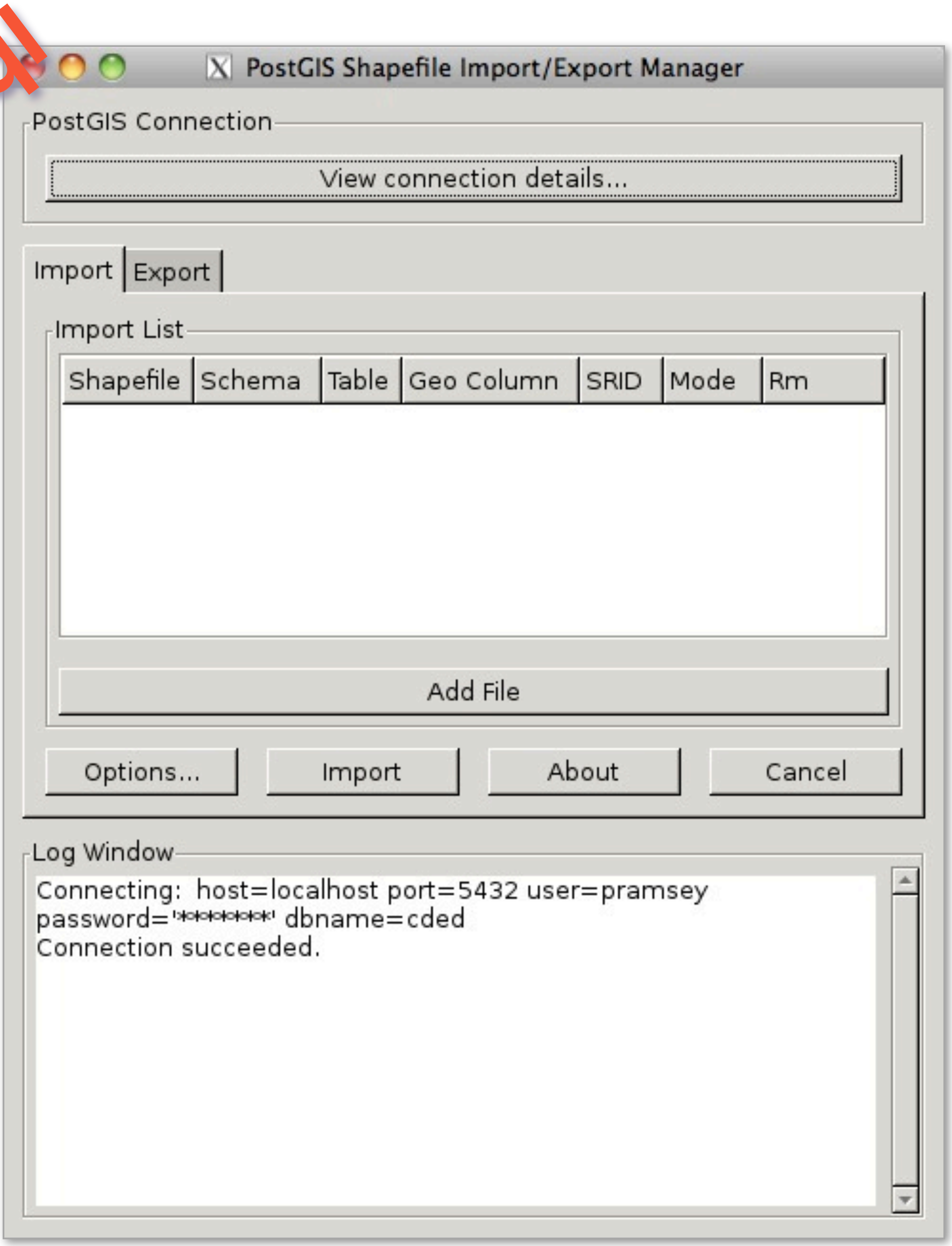
And densifies it in GEOGRAPHIC space, so along the great circle edges between the vertices. This is a 300km densification, and it’s cool to see not only the great circle arc to the north, but the huge distortion in distance in the higher latitudes too. Every vertex here is 300km apart.

shp2pgsql

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

before 1.5, this is how you loaded shapefile data

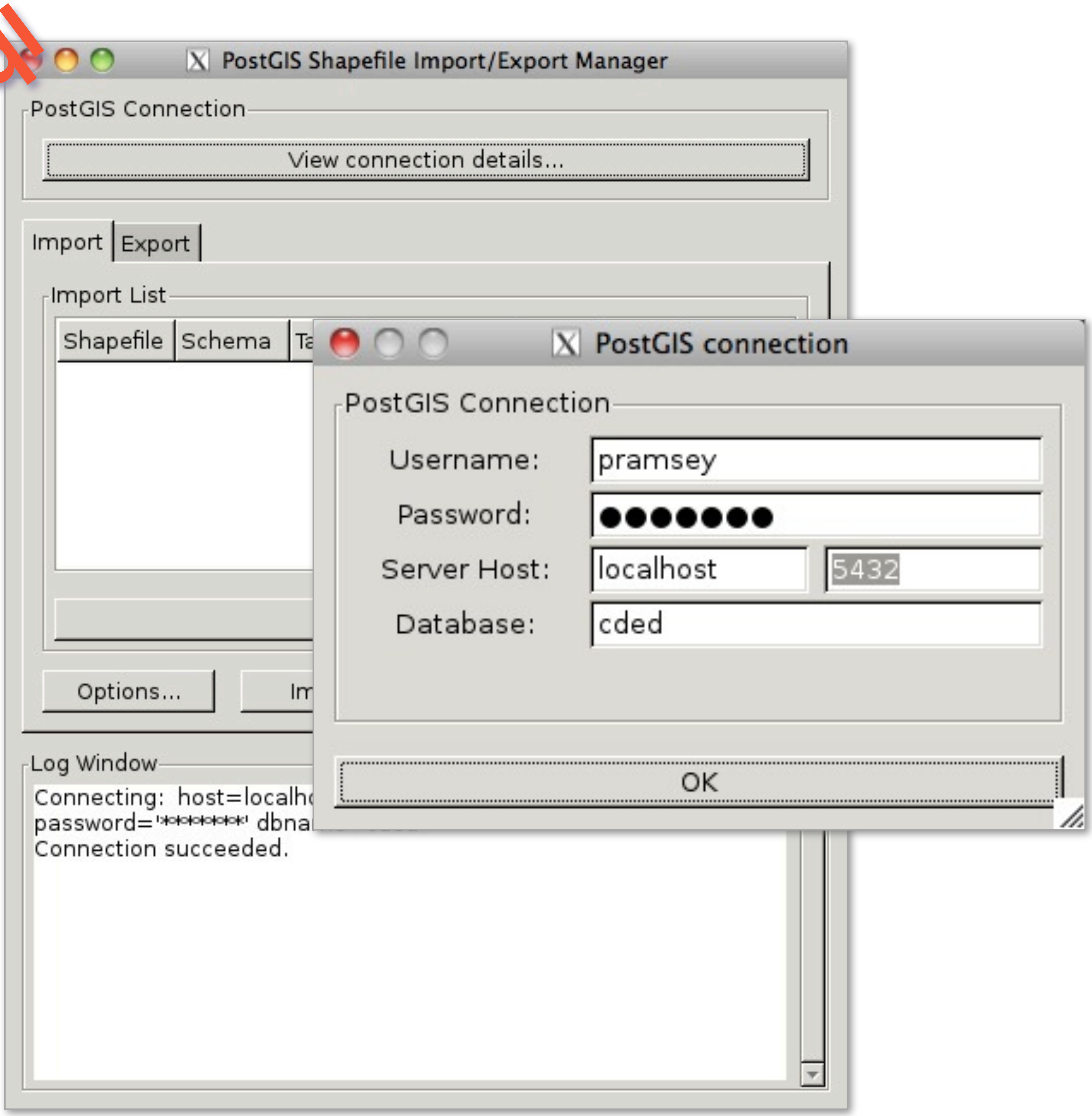
shp2pgsql



```
Connecting:  host=localhost port=5432 user=pramsey
password='********' dbname=cded
Connection succeeded.
```

PostGIS 1.5 added a GUI loader tool
The 2.0 version includes the ability to load multiple files in a batch!

shp2pgsql



```
Connecting:  host=localhc
password='********' dbna
Connection succeeded.
```

Username: pramsey

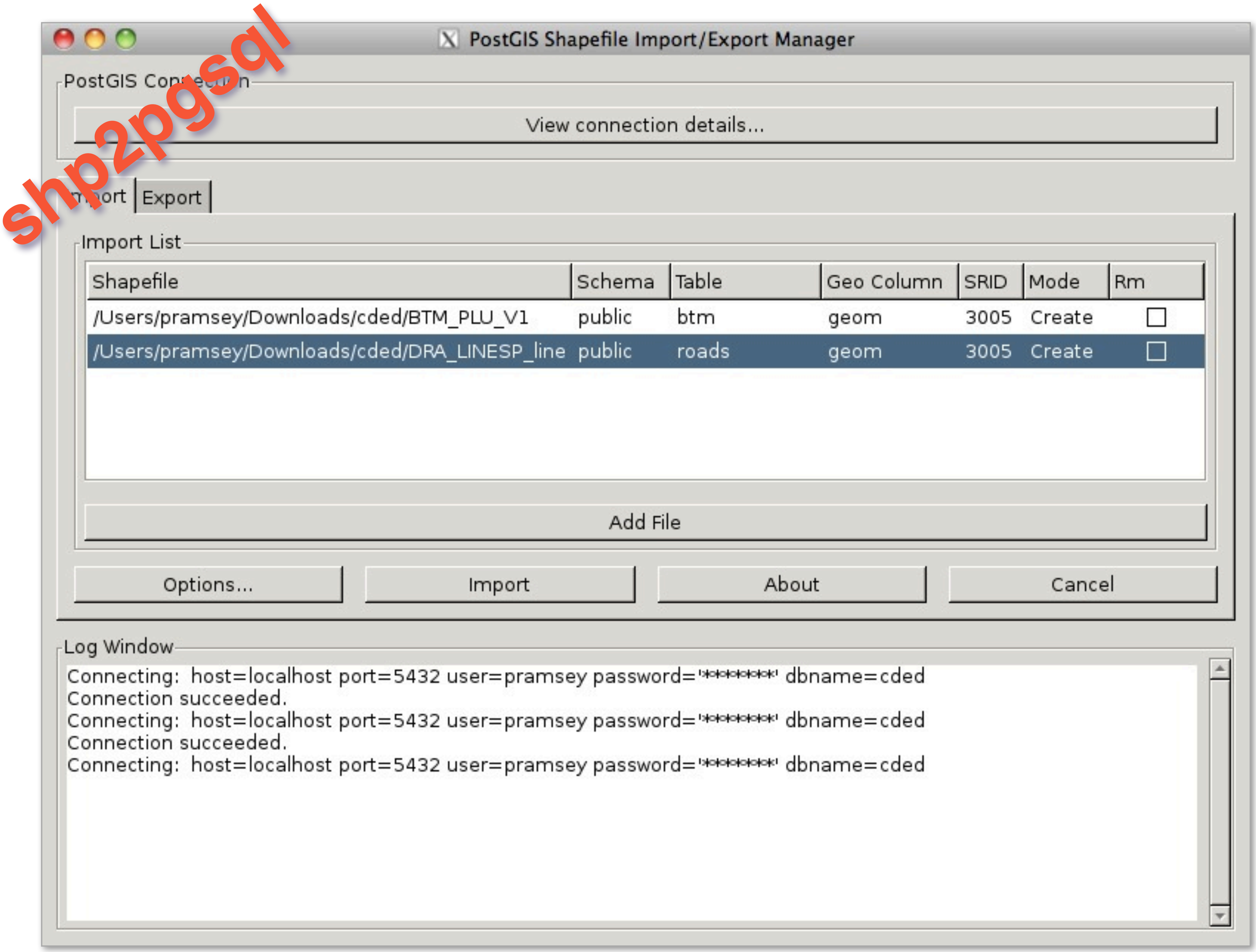
Password: ●●●●●●●●

Server Host: localhost 5432

Database: cded

Thursday, March 12, 15

PostGIS 1.5 added a GUI loader tool
The 2.0 version includes the ability to load multiple files in a batch!

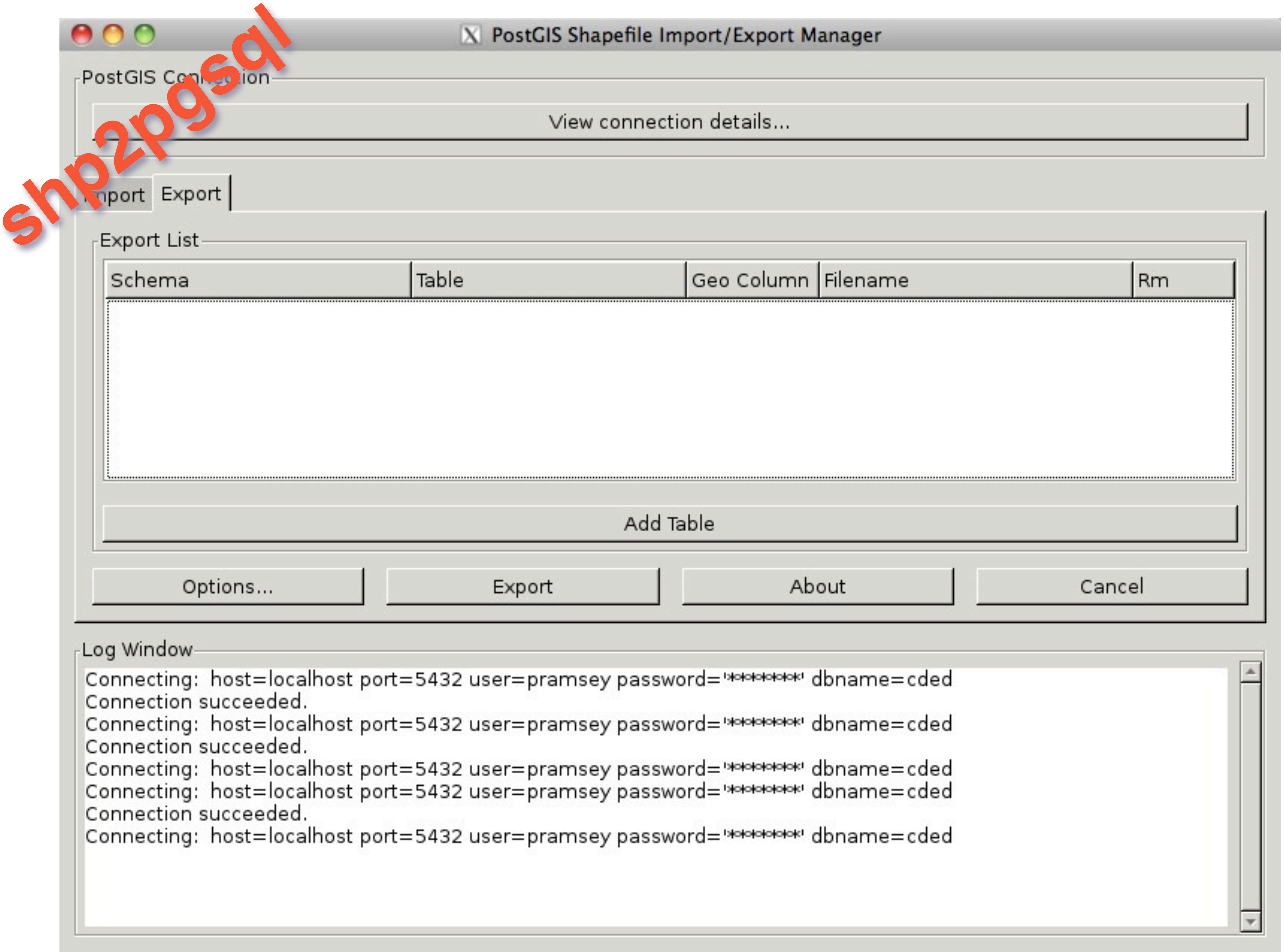


Shapefile	Schema	Table	Geo Column	SRID	Mode	Rm
/Users/pramsey/Downloads/cded/BTM_PLU_V1	public	btm	geom	3005	Create	□
/Users/pramsey/Downloads/cded/DRA_LINESP_line	public	roads	geom	3005	Create	□

```
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
```

Thursday, March 12, 15

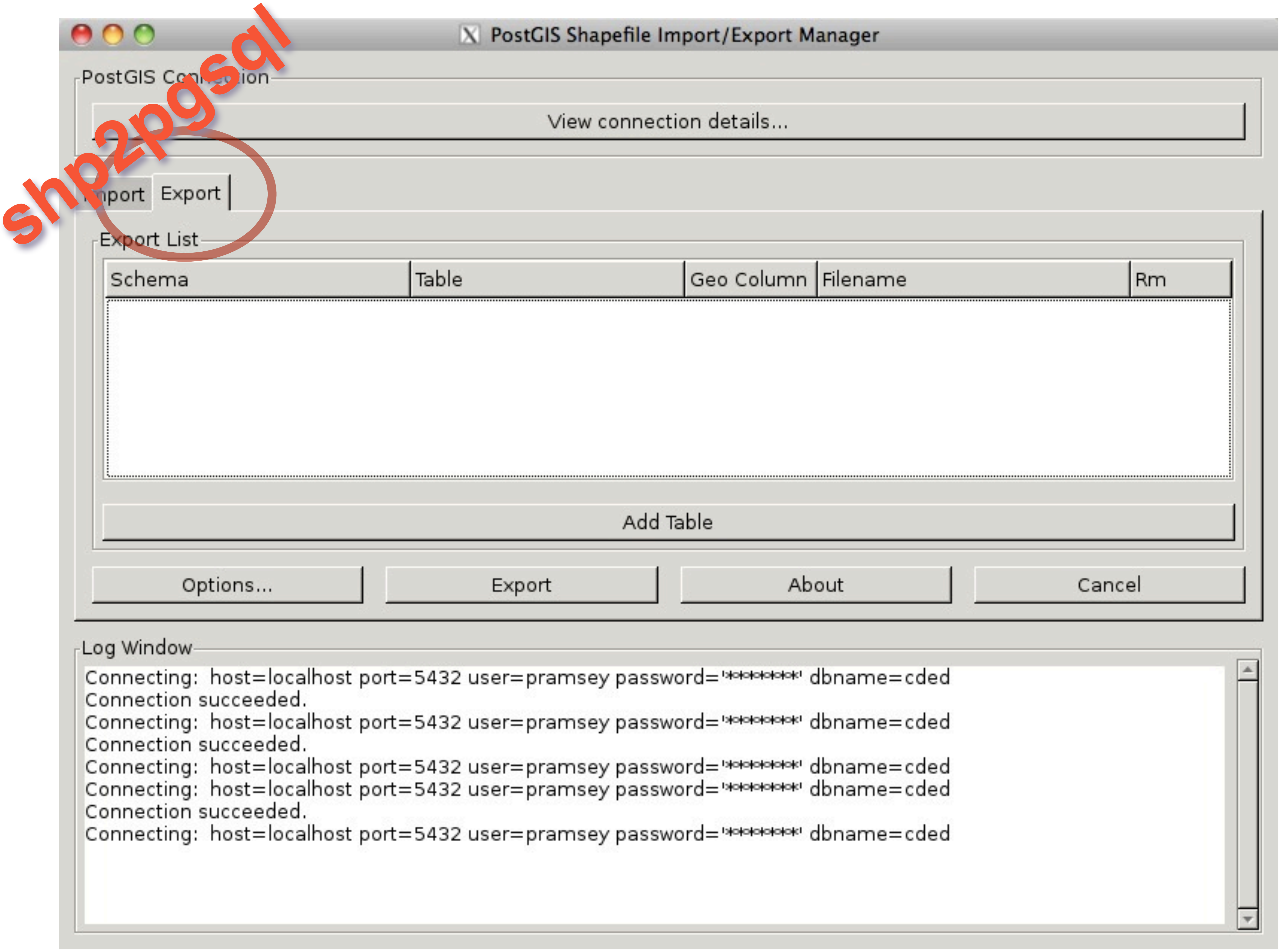
PostGIS 1.5 added a GUI loader tool
The 2.0 version includes the ability to load multiple files in a batch!



```
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
Connecting:  host=localhost port=5432 user=pramsey password='********' dbname=cded
Connection succeeded.
Connecting:  host=localhost port=5432 user=pramsey password='********' dbname=cded
```

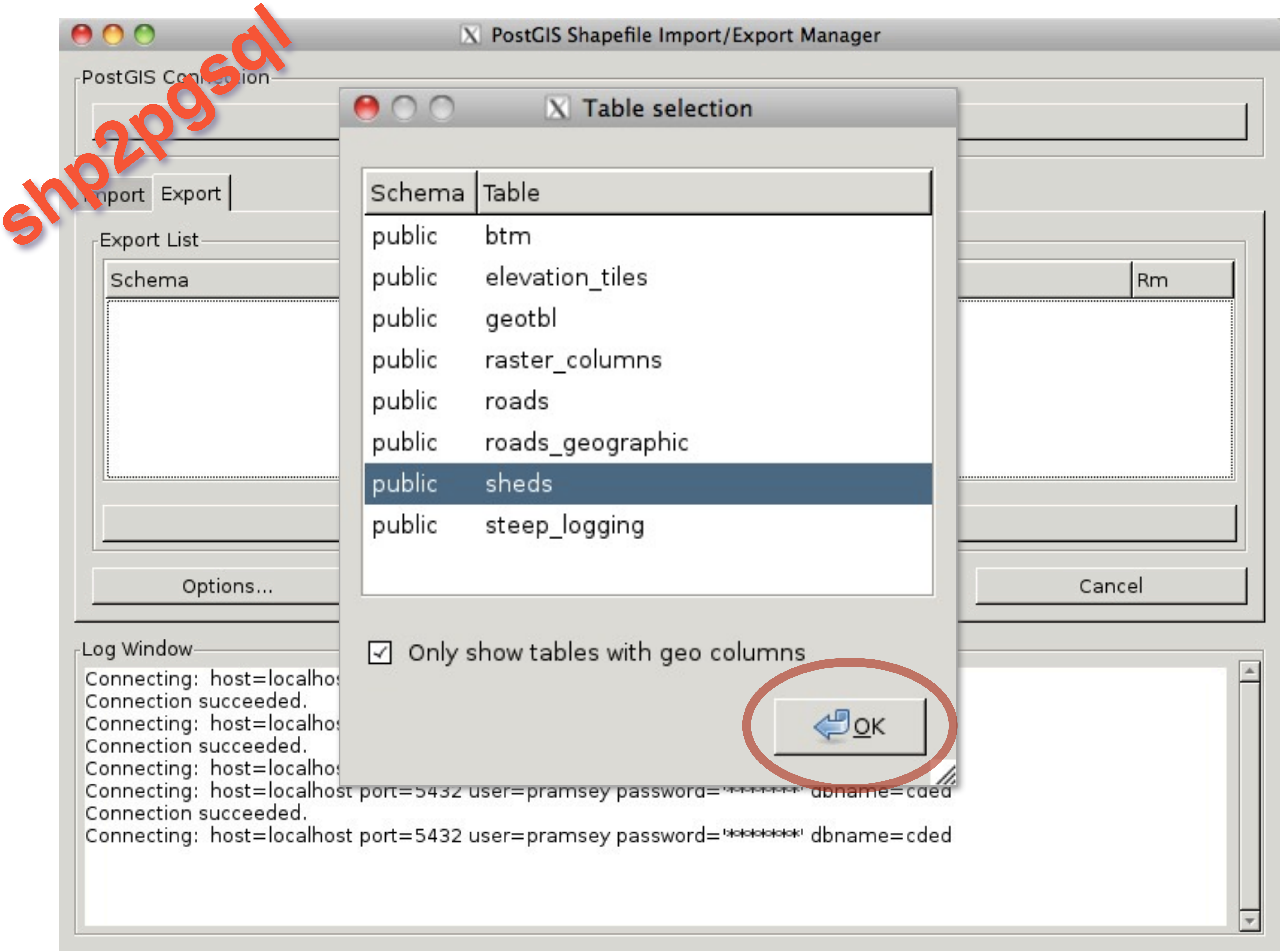
Thursday, March 12, 15

PostGIS 1.5 added a GUI loader tool
The 2.0 version includes the ability to load multiple files in a batch!



```
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
Connecting:  host=localhost port=5432 user=pramsey password='********' dbname=cded
Connection succeeded.
Connecting:  host=localhost port=5432 user=pramsey password='********' dbname=cded
```

PostGIS 1.5 added a GUI loader tool
The 2.0 version includes the ability to load multiple files in a batch!



Schema	Table
public	btm
public	elevation_tiles
public	geotbl
public	raster_columns
public	roads
public	roads_geographic
public	sheds
public	steep_logging

☑ Only show tables with geo columns

```
Connecting:  host=localhos
Connection succeeded.
Connecting:  host=localhos
Connection succeeded.
Connecting:  host=localhos
Connecting:  host=localhost port=5432 user=pramsey password='*******' dbname=cded
Connection succeeded.
Connecting:  host=localhost port=5432 user=pramsey password='*******' dbname=cded
```

Thursday, March 12, 15

PostGIS 1.5 added a GUI loader tool
The 2.0 version includes the ability to load multiple files in a batch!

Typmod

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

Thursday, March 12, 15

If you’re still on 1.5, it’s time to upgrade!
You’ll get
“TypMod” support !
lets you declare the type and srid and dimsionality
of a geometry right at table CREATE time.

Typmod

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

With type/srid in the system tables, changing the srid or type of a column becomes a single line of DDL!

Typmod

```
SELECT *
FROM geometry_columns
WHERE
  f_table_name =
  ‘my_spatial_table’
```

OMG it’s a view now!!!!

With all the type/srid info in the system tables,
GEOMETRY_COLUMNS IS A VIEW NOW!

Typmod

```
------------------+-----------------
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
```

In the 3rd Dimension

There’s lots and lots of new 3D support in recent releases!

3D Support!

3D Types!

- TRIANGLE
- TIN
- POLYHEDRALSURFACE

OSLANDIA

We also have 3D types to go with those new functions and indexes.

3D Support!

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

OSLANDIA

The collection of 3D enabled functions has grown a great deal. Distance, length, nearest points, even intersects and within.

3D Support!

3D Formats!

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

Thursday, March 12, 15

And new 3D formats to write those 3D objects out to the wire.

ND-Index!

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

Yes, in 2.0 you can create indexes and search in 3D and 4D!

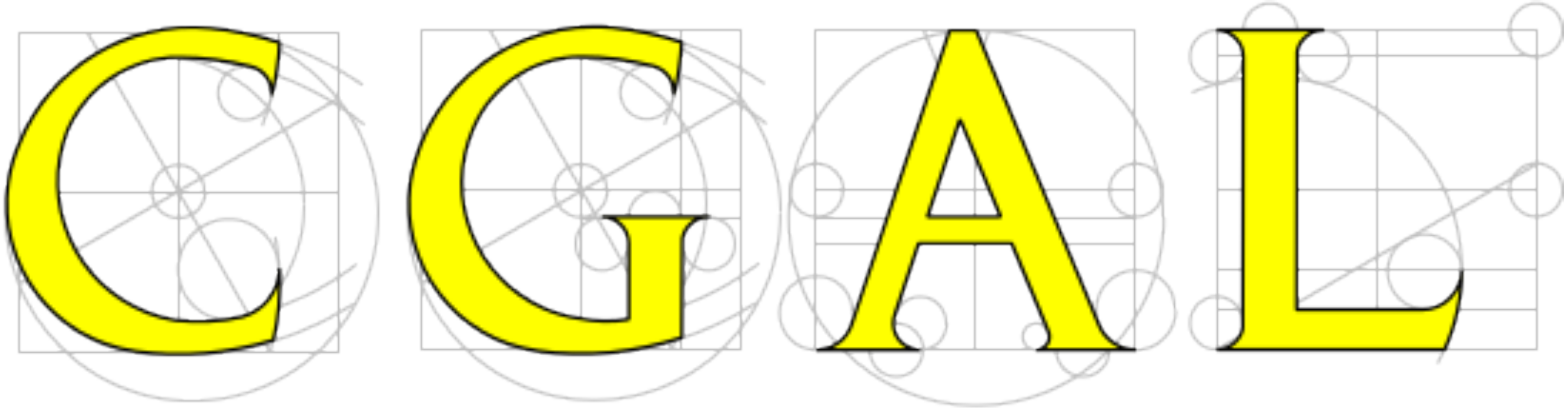
ND-Index!

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

Yes, in 2.0 you will be able to create indexes and search in 3D and 4D!

3D Support!

New to 2.1!



OSLANDIA

With 2.1, a binding to the CGAL computational geometry library has allowed us to add even more 3D functions than before. This binding will also give us access to more complex CGAL functionality in the future.

3D Support!

New to 2.1!

- **Leveraging the CGAL library**
- ST_3DIntersection
- ST_Tesselate
- ST_3DArea
- ST_Extrude
- ST_ForceLHR
- ST_Orientation
- ST_Minkowski
- ST_StraightSkeleton

OSLANDIA

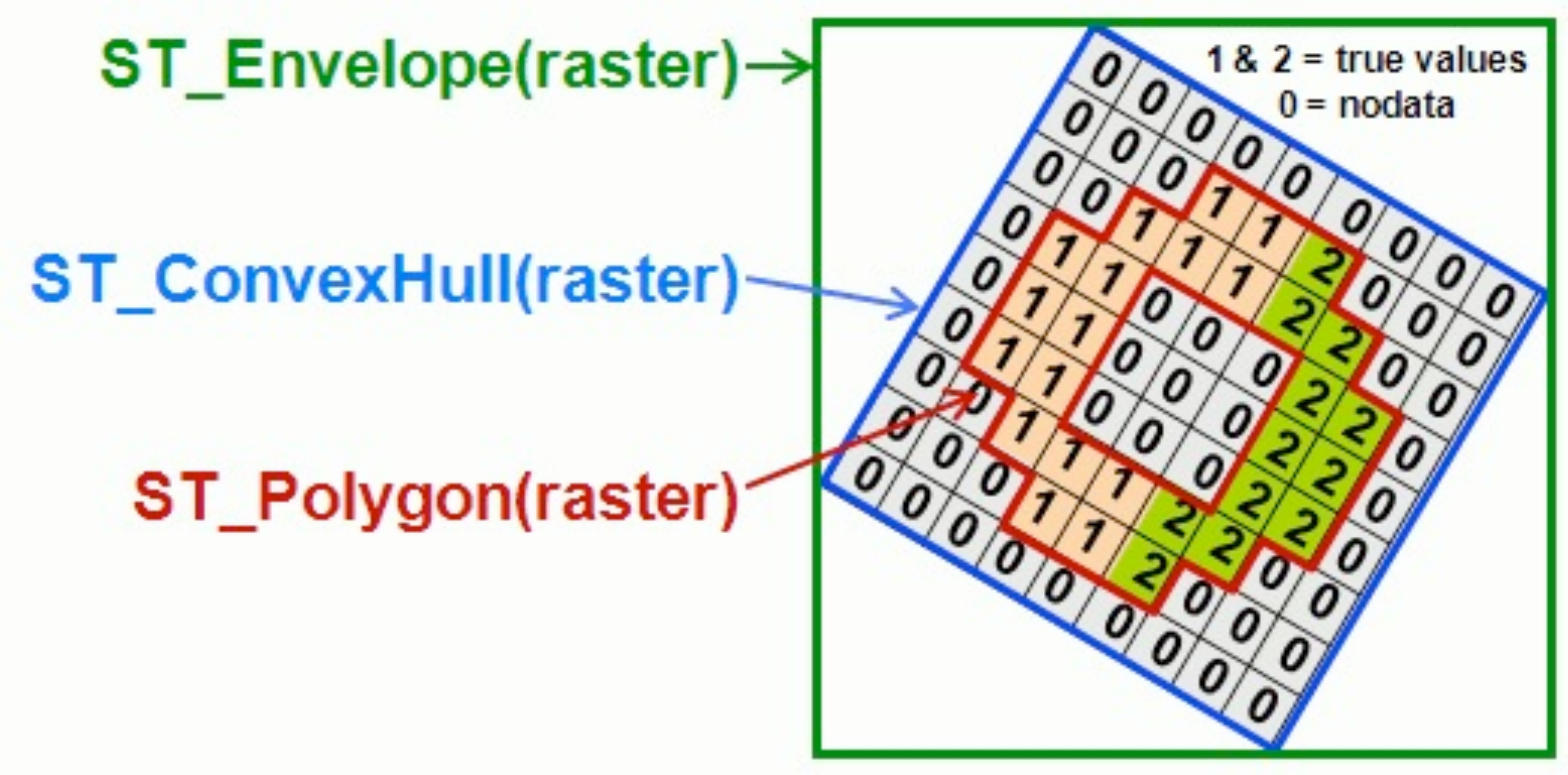
Swappable Backend!

New to 2.1!

- **Are you freaking kidding me?**
- set postgis.backend = ‘geos’;
  set postgis.backend = ‘cgal’;
- ST_Intersects()
  ST_3DIntersects()
  ST_Intersection()
  ST_Area()
  ST_Distance()
  ST_3DDistance()

OSLANDIA

Raster Types

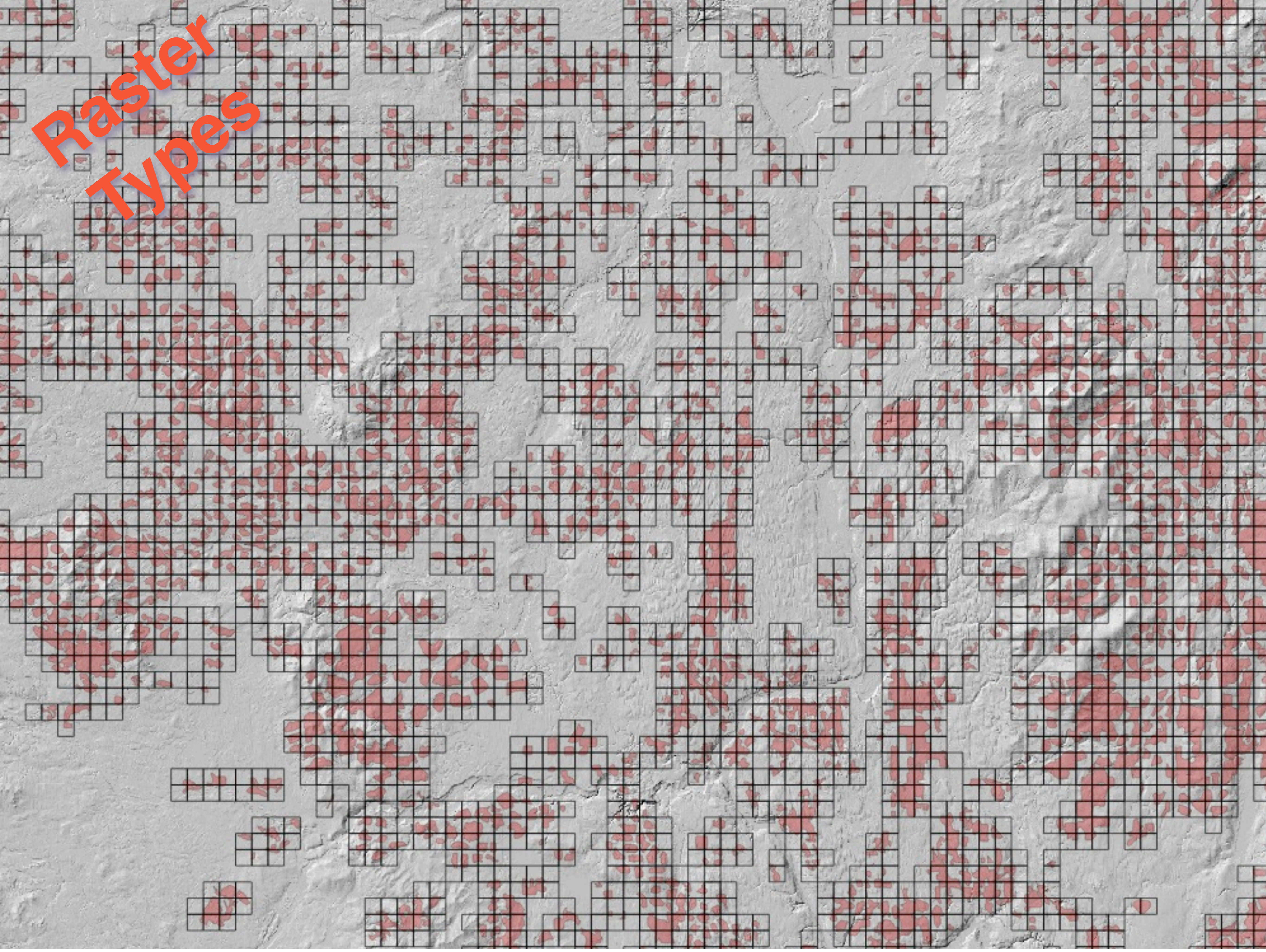


Raster is a stupid idea,
unless you’re doing ANALYSIS

Raster Types

Thursday, March 12, 15

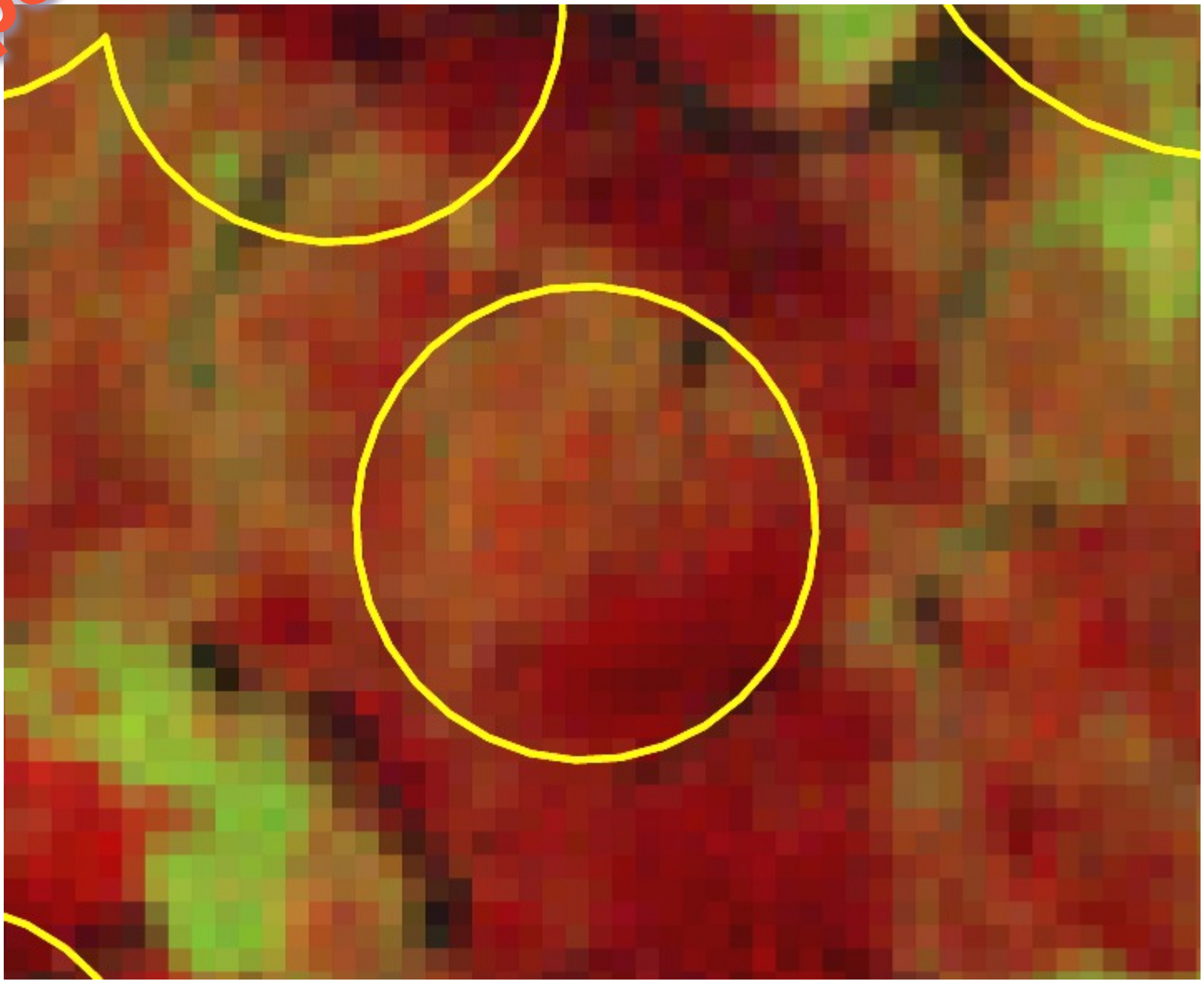
model a raster coverage as a big collection raster chips



Thursday, March 12, 15

So we can do a raster/vector spatial join and
ANALYSIS
We can join the two tables, finding the slope grid chips that intersect logging areas. And then summarize to find the actual steep slope logging.

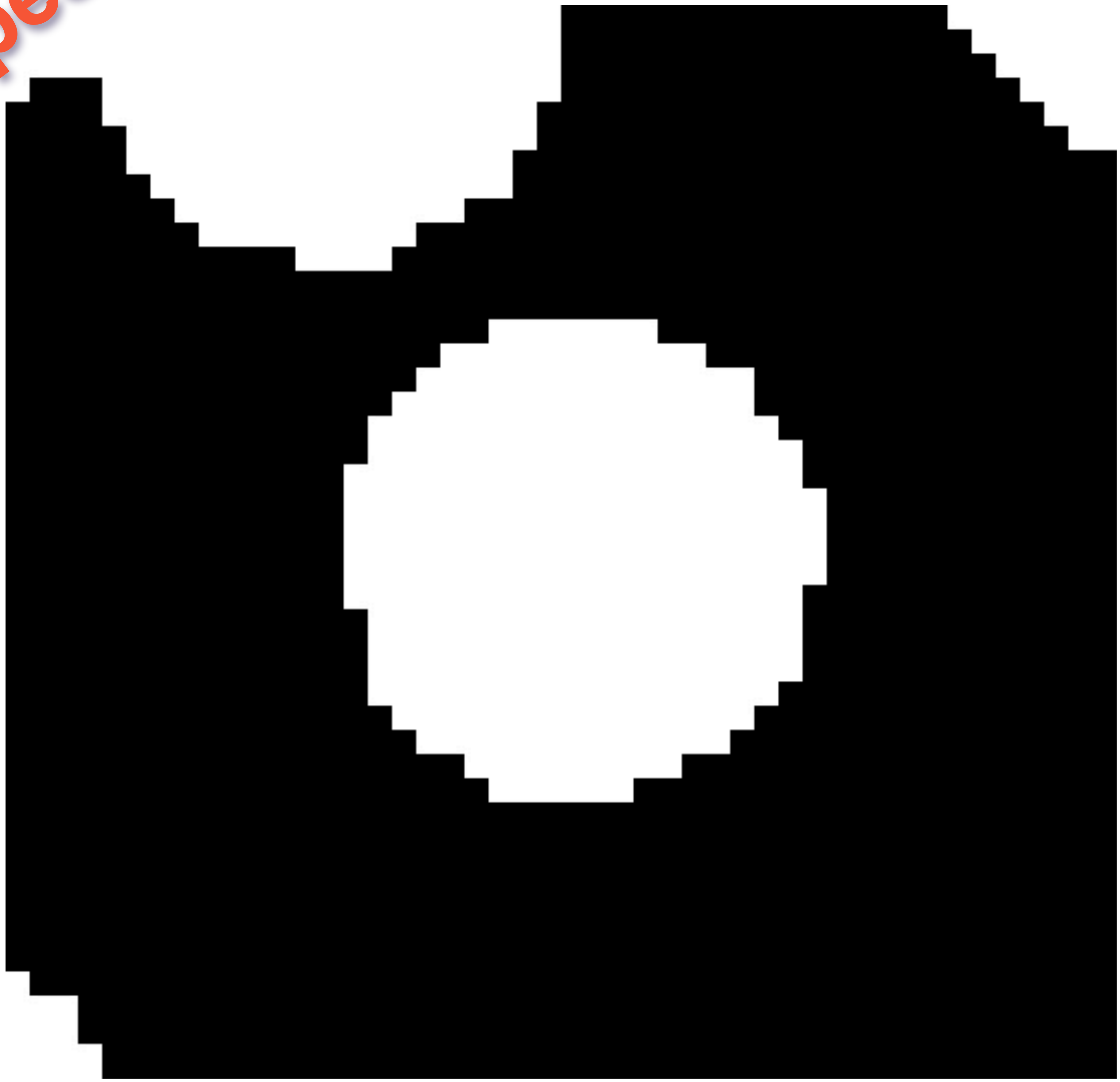
Raster Types



Thursday, March 12, 15

ANALYSIS,
what if you have areas of interest in vector
and data in raster

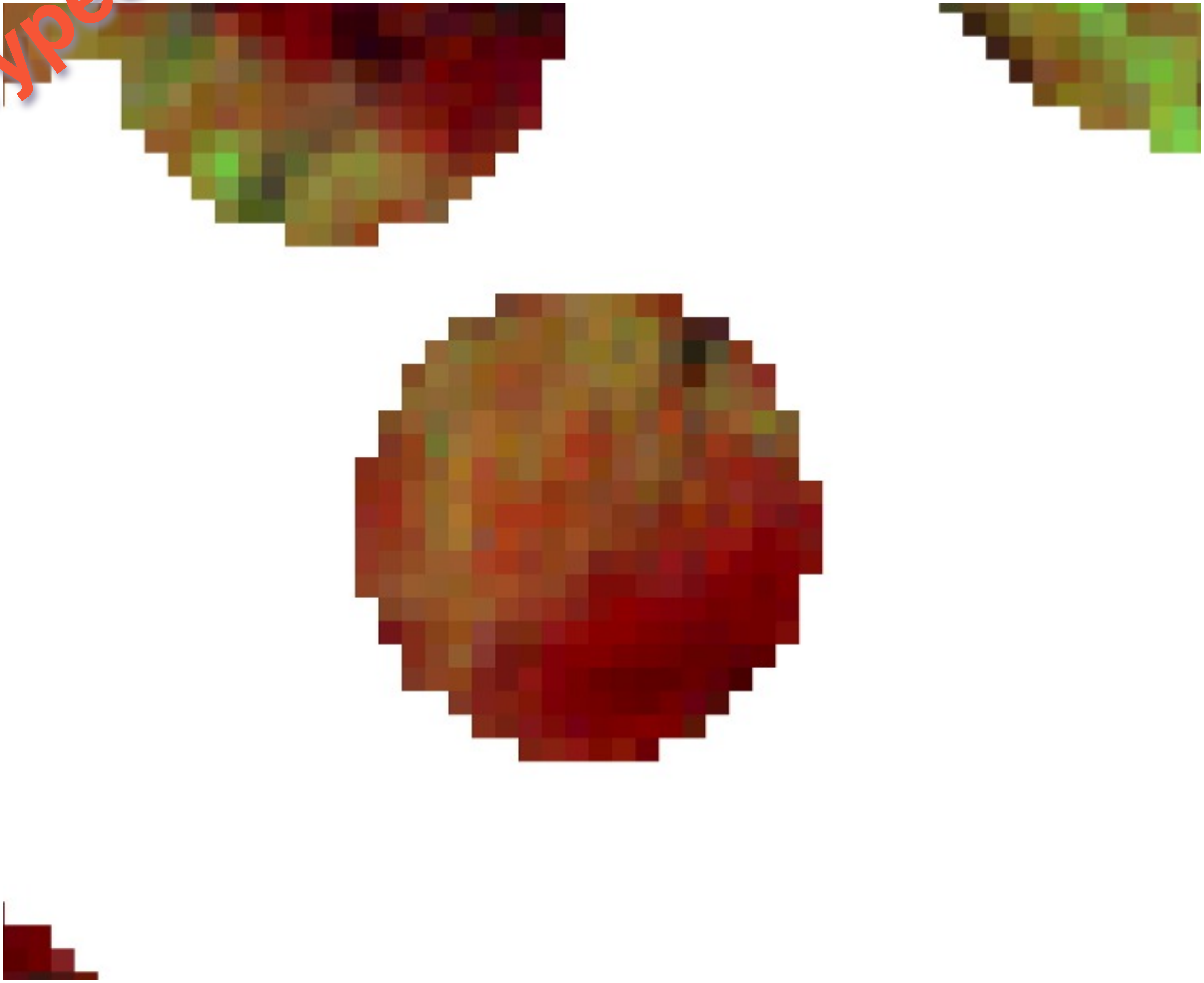
Raster Types



Thursday, March 12, 15

create a raster mask from the vectors

Raster Types

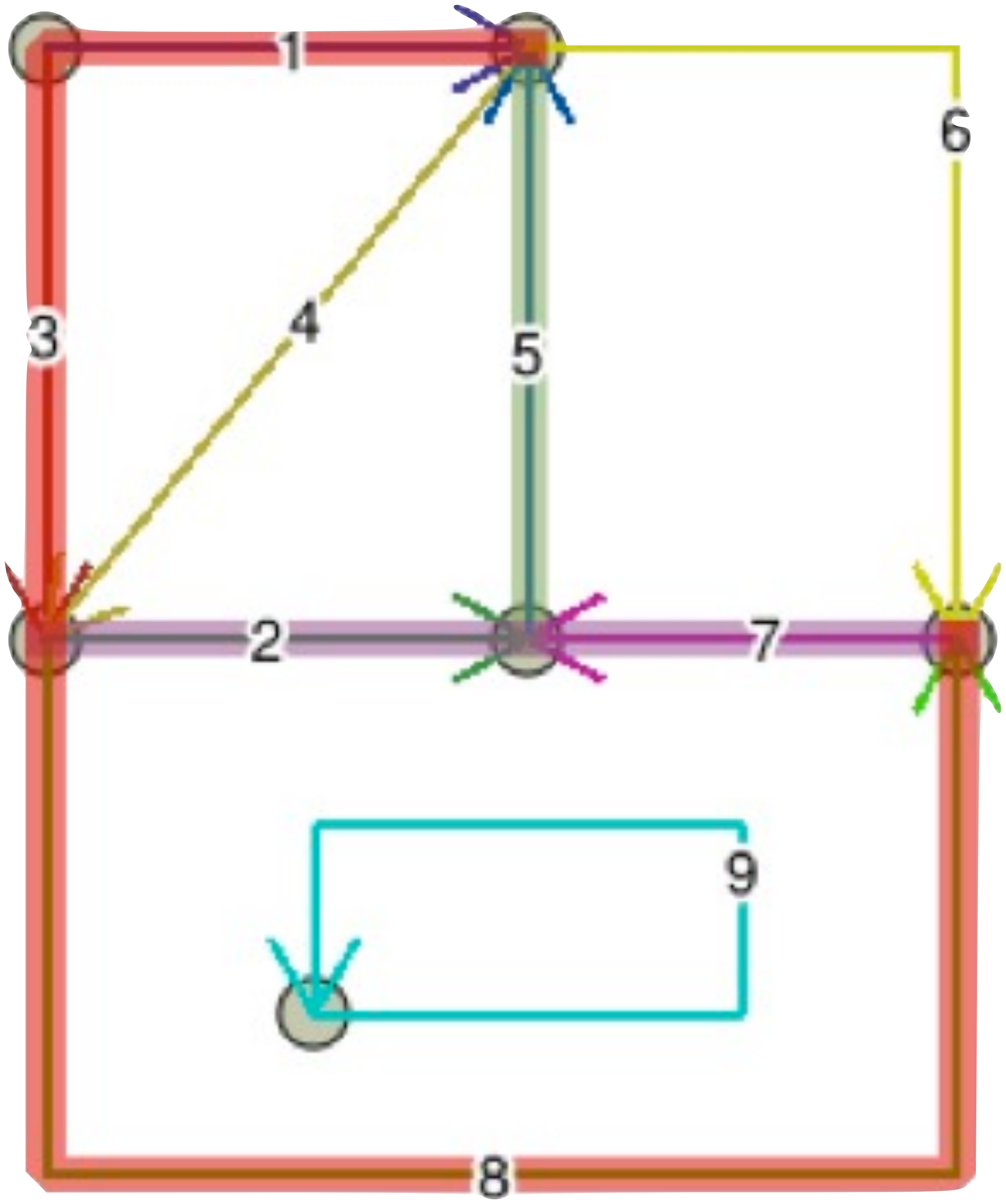


Thursday, March 12, 15

overlay and summarize!
ANALYSIS!

Topology!

Topology



Thursday, March 12, 15

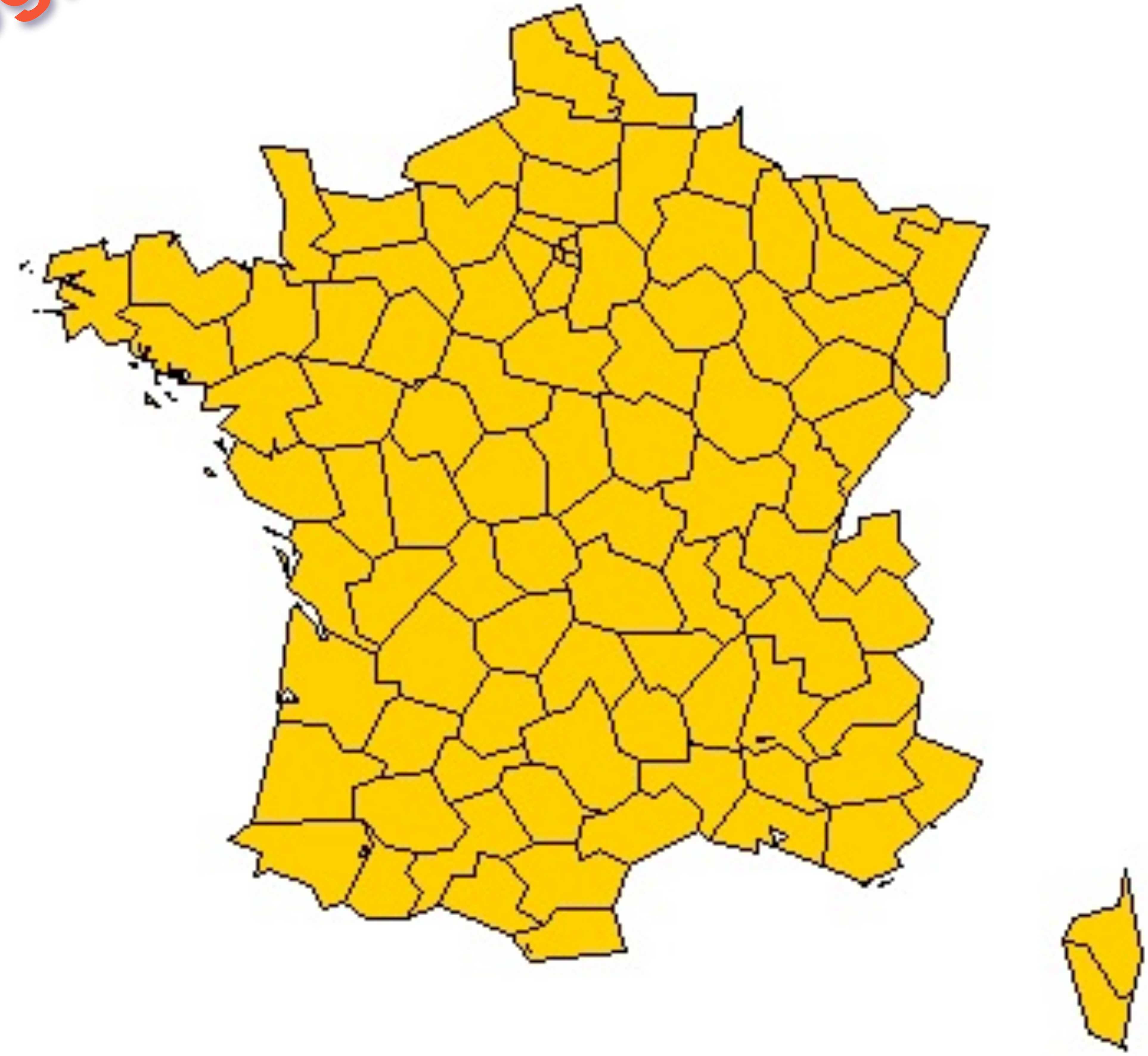
In 2.0 you can create topologies!
The classic use case if for data management of things like PARCELS and CADASTRE that have shared boundaries

Topology!

Thursday, March 12, 15

but you can also do fun processing tricks
like taking a polygon set with shared boundaries
building a topology and

Topology!



Thursday, March 12, 15

simplifying the edges without generating edge gaps

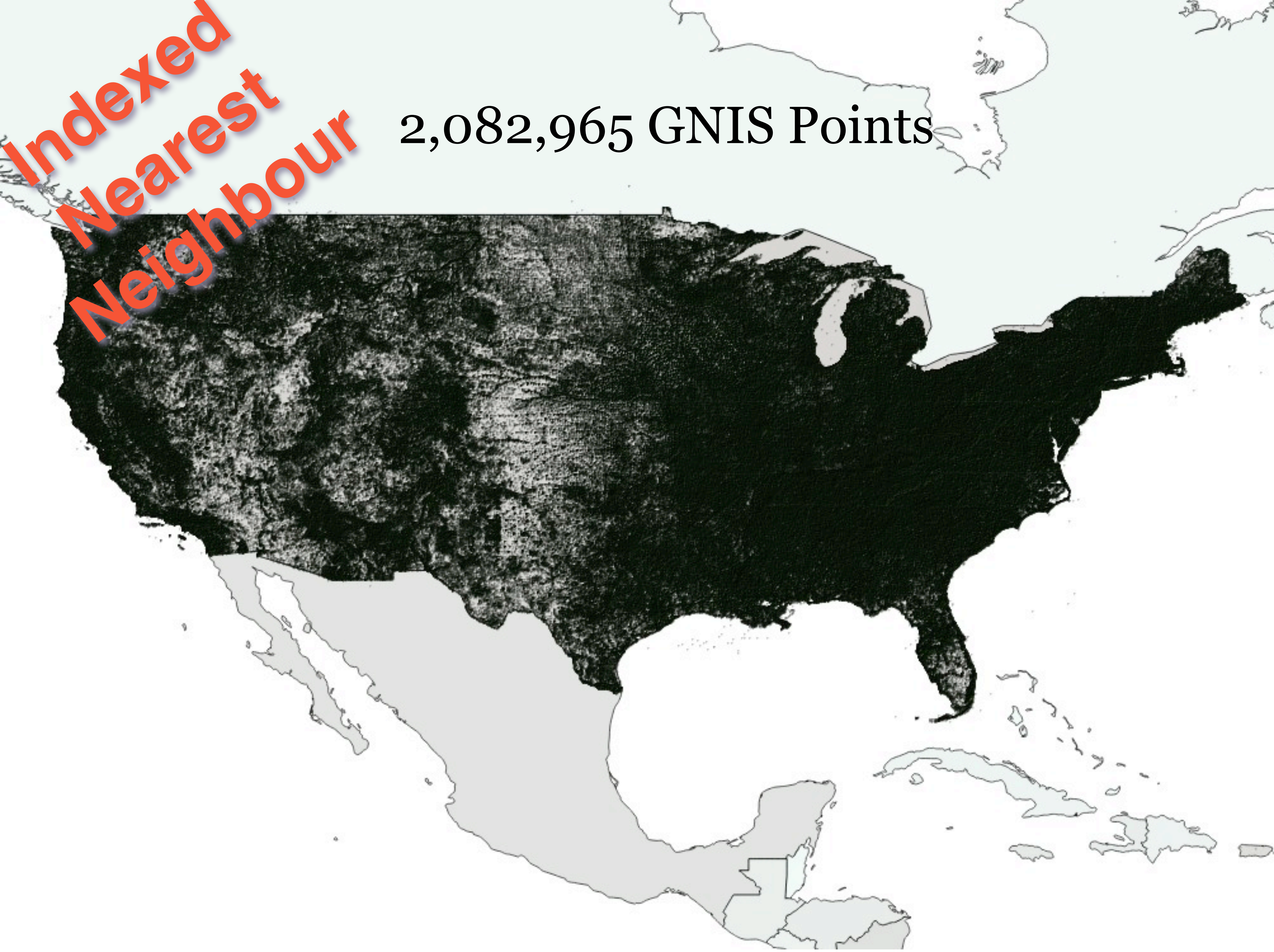
Indexed
Nearest
Neighbour

Indexed KNN

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

Thursday, March 12, 15

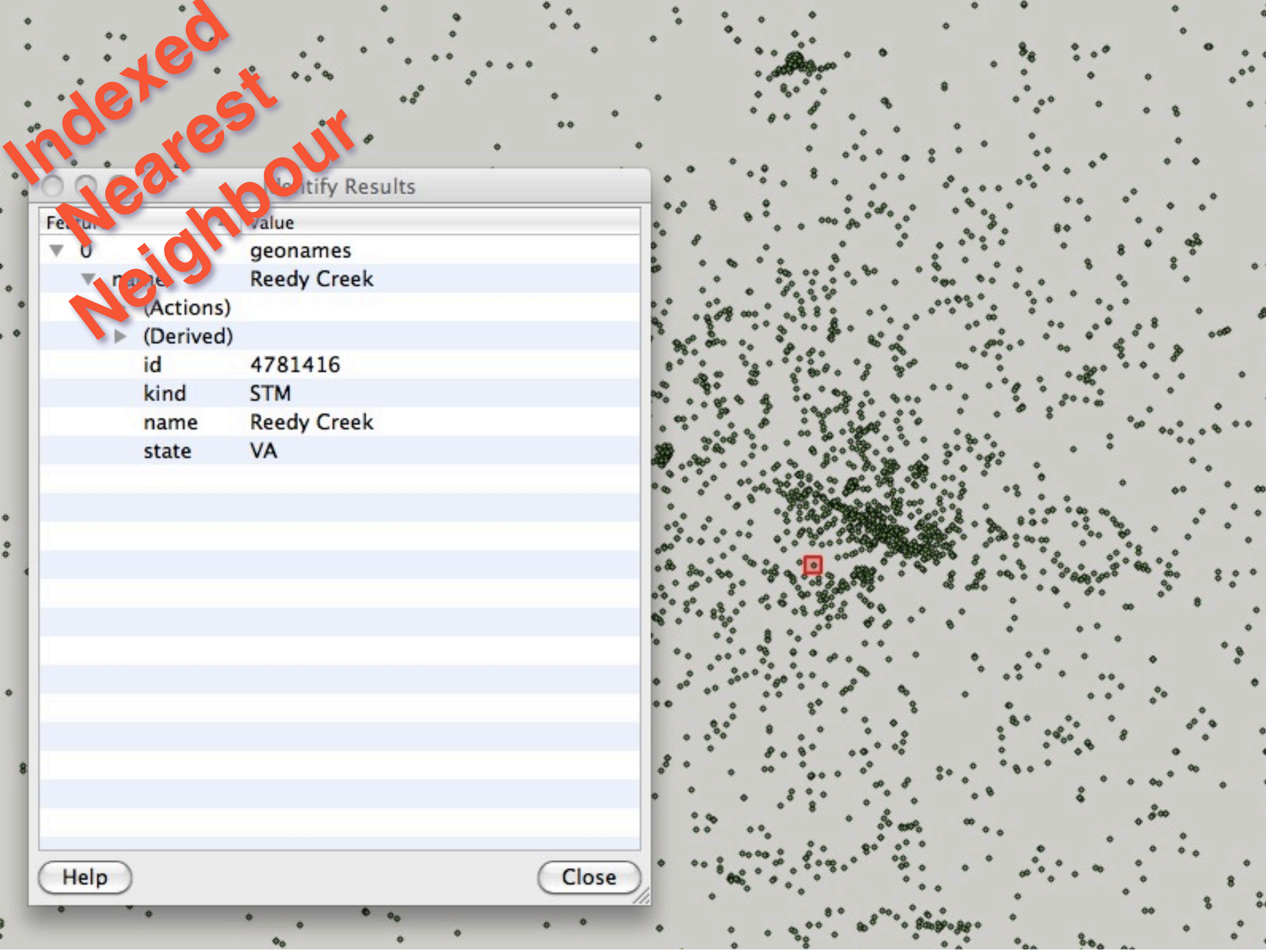
PostGIS 2.0 has support for nearest-neighbor indexed searching. For very large tables, with irregular densities, this can be a huge performance win.



Indexed Nearest Neighbour

2,082,965 GNIS Points

So, here’s an example I put together, loading all the USA named geographic points, 2M of them.



Indexed Nearest Neighbour

Identify Results

Feature	Value
▼ 0	geonames
▼ name	Reedy Creek
(Actions)	
▶ (Derived)	
id	4781416
kind	STM
name	Reedy Creek
state	VA

Find one point, in this case Reedy Creek.

Indexed Nearest Neighbour

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

Here’s how we find the 10 nearest names to Reedy Creek. Note the use of the funny arrow-like operator in the ORDER BY clause and the LIMIT. You have to use ORDER BY and you have to LIMIT.

Indexed Nearest Neighbour

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

Thursday, March 12, 15

Here’s how we find the 10 nearest names to Reedy Creek. Note the use of the funny arrow-like operator in the ORDER BY clause and the LIMIT. You have to use ORDER BY and you have to LIMIT.

Indexed Nearest Neighbour

```
   id    |               name                | state | kind
---------+-----------------------------------+-------+------
 4781416 | 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
```

But most importantly, note how fast we get back the 10 nearest entries from this 2M record table.

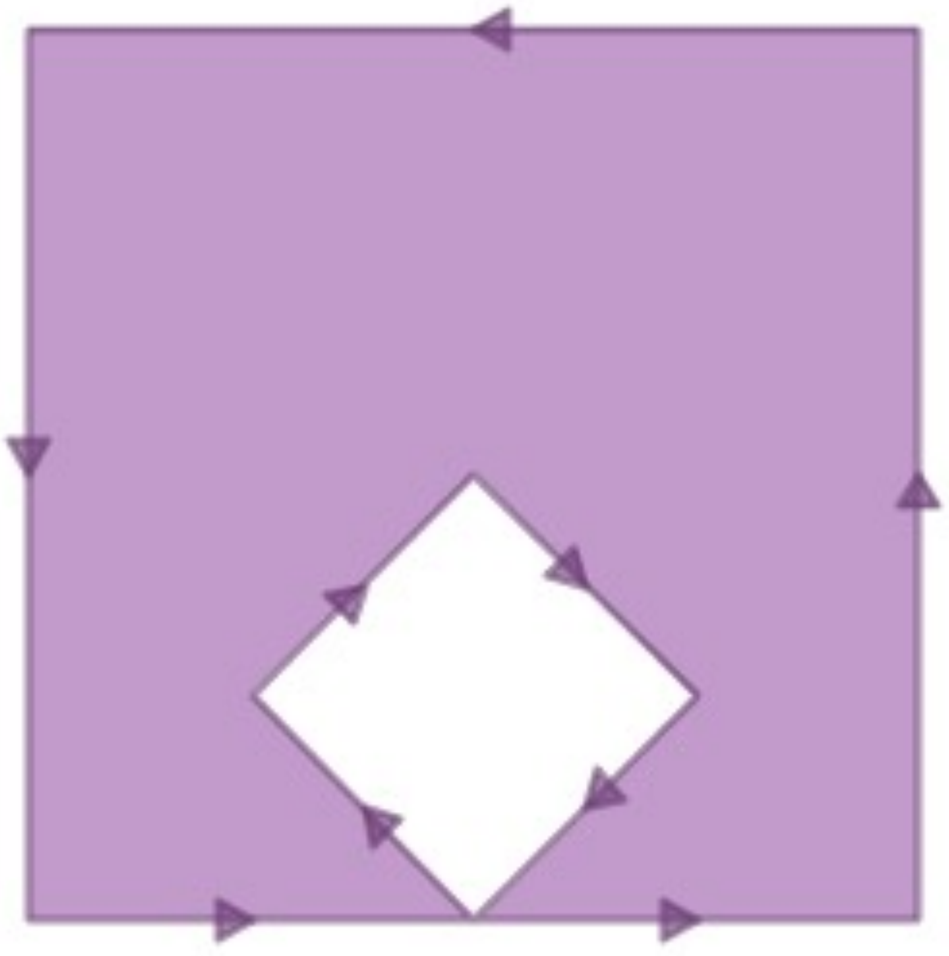
Indexed Nearest Neighbour

```
   id    |                 name                  | state | kind
---------+---------------------------------------+-------+------
 4781416 | 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
```

But most importantly, note how fast we get back the 10 nearest entries from this 2M record table.

Validity Reporting

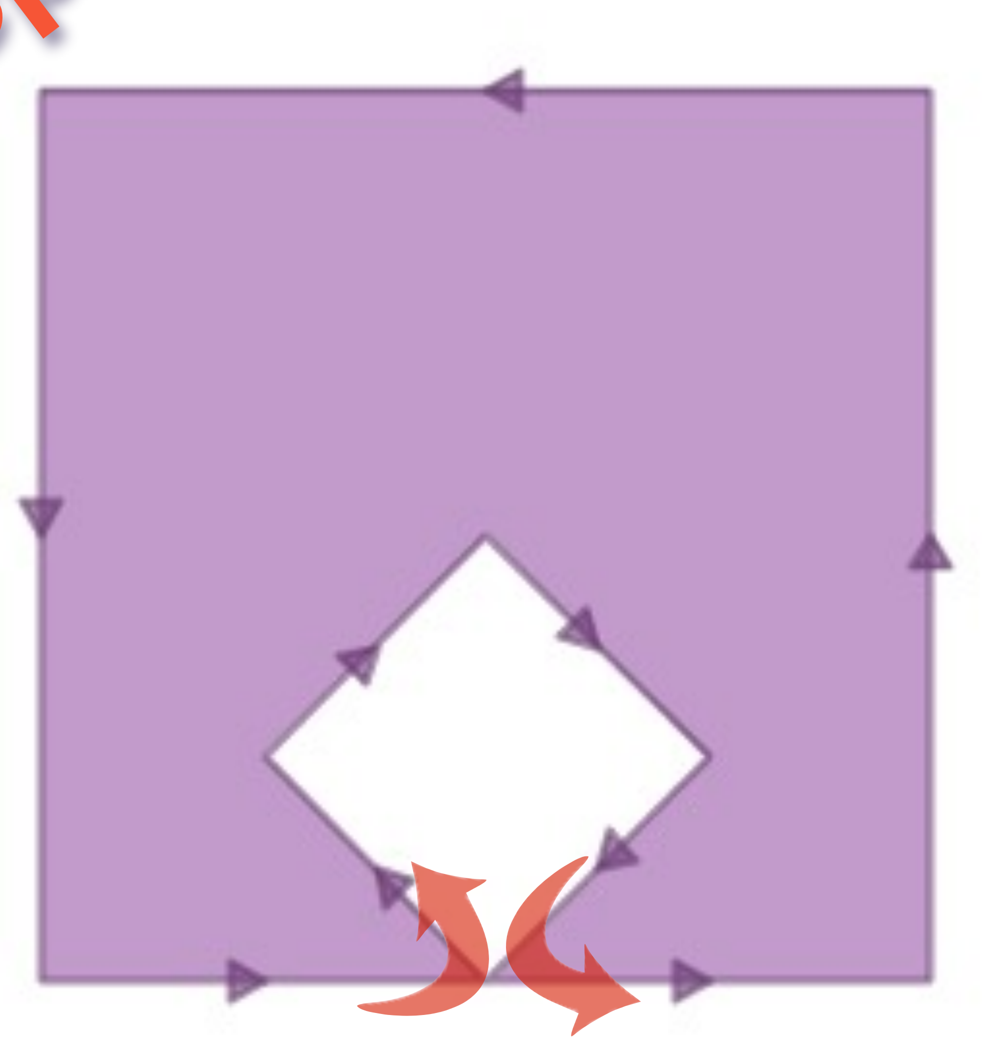


This polygon is invalid because it consists of just one ring that loops around and touches itself at the bottom. I call it a “banana polygon” because it is like a banana that has been bent until the ends touch.

The correct way to construct this shape is with an exterior and an interior ring that touch at one point.

There is no “right” way to do this. ESRI actually considers the first case valid and the second one invalid. They aren’t wrong, their internal standard is just different.

Validity Reporting



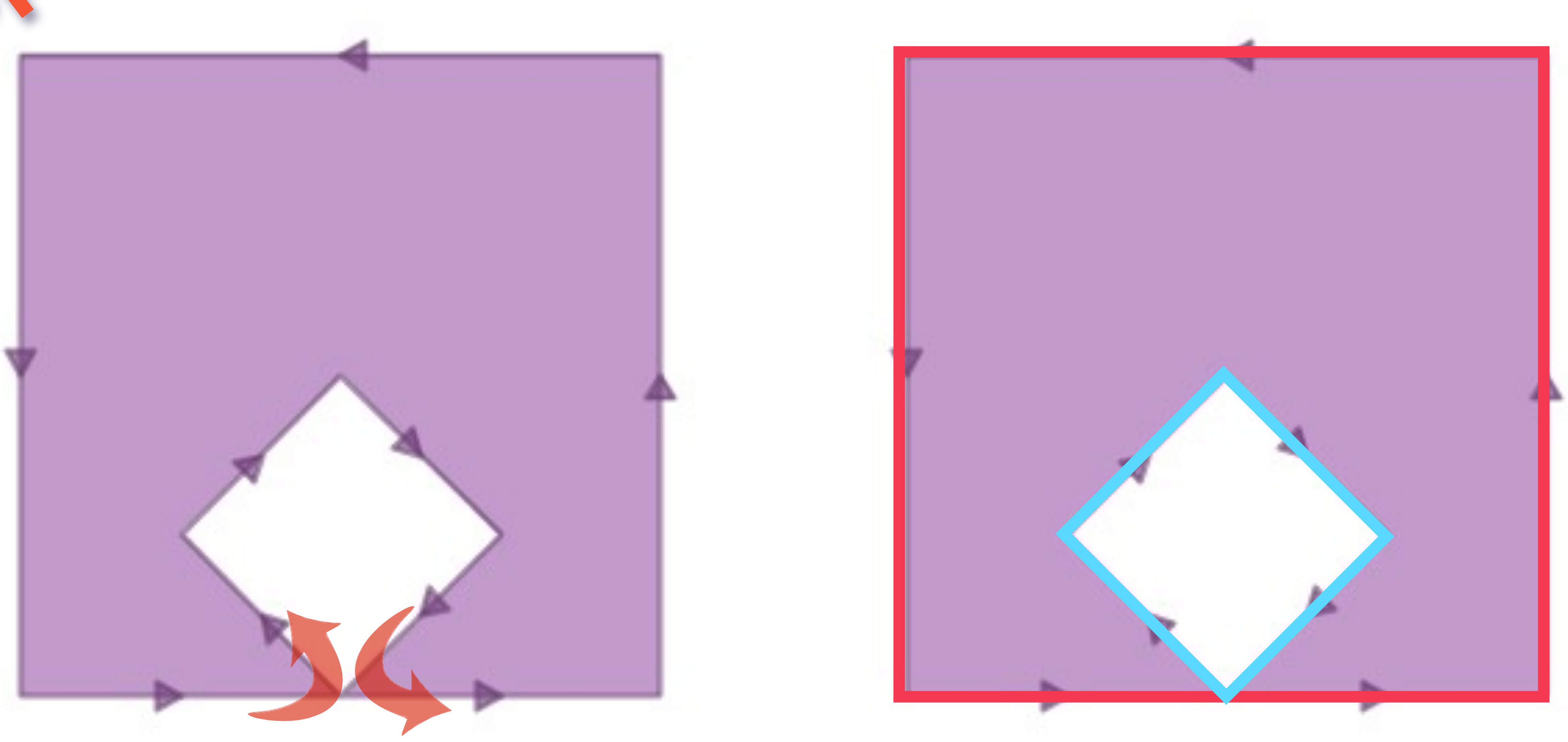
Thursday, March 12, 15

This polygon is invalid because it consists of just one ring that loops around and touches itself at the bottom. I call it a “banana polygon” because it is like a banana that has been bent until the ends touch.

The correct way to construct this shape is with an exterior and an interior ring that touch at one point.

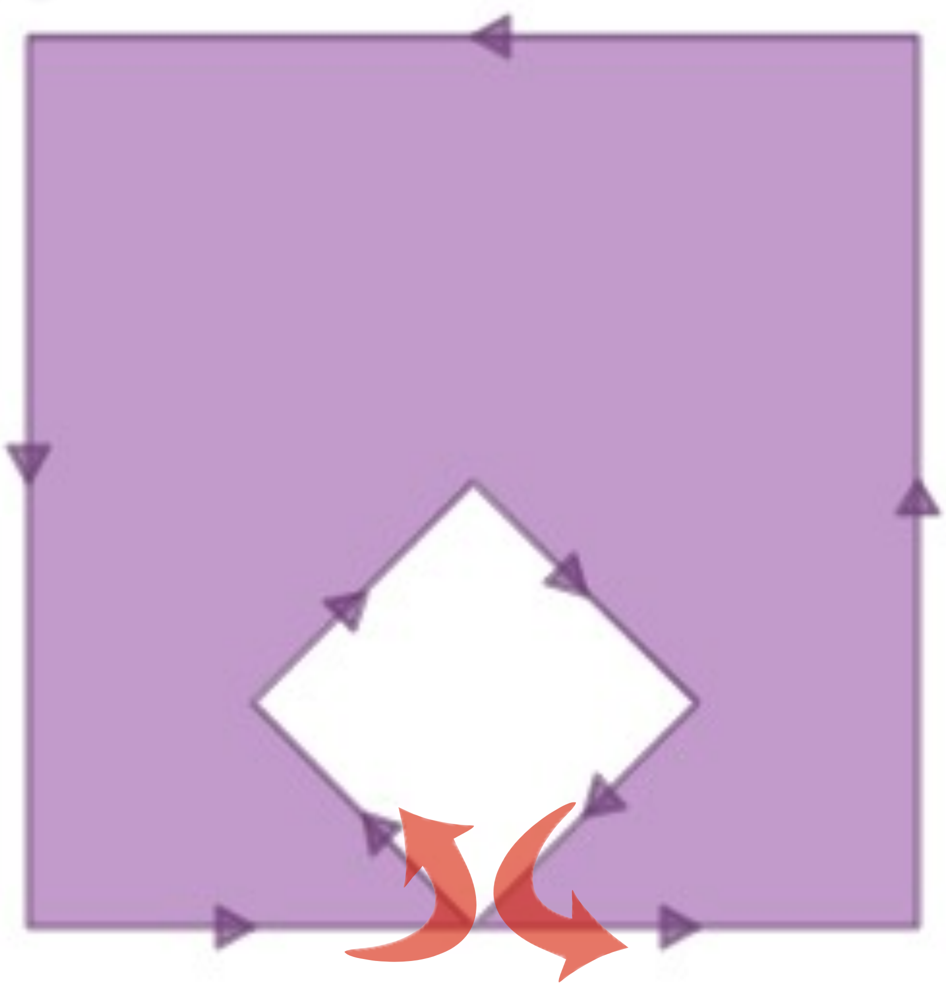
There is no “right” way to do this. ESRI actually considers the first case valid and the second one invalid. They aren’t wrong, their internal standard is just different.

Validity Reporting

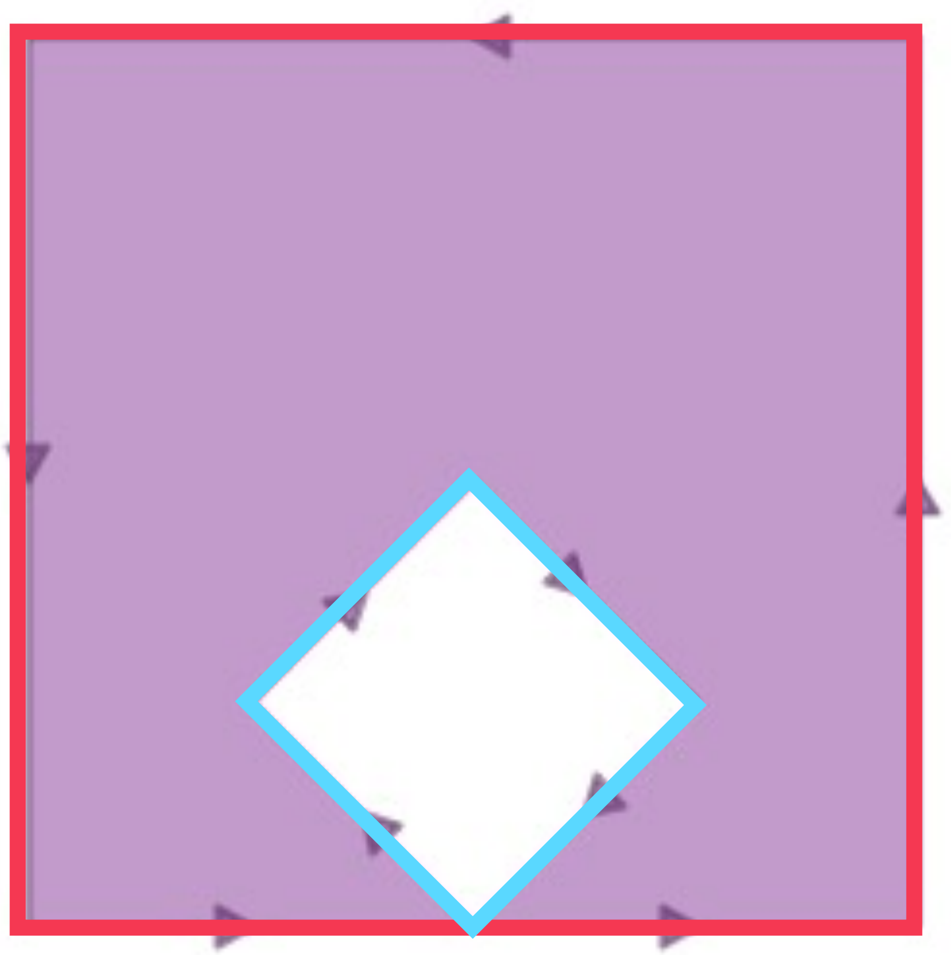


This polygon is invalid because it consists of just one ring that loops around and touches itself at the bottom. I call it a “banana polygon” because it is like a banana that has been bent until the ends touch.
The correct way to construct this shape is with an exterior and an interior ring that touch at one point.
There is no “right” way to do this. ESRI actually considers the first case valid and the second one invalid. They aren’t wrong, their internal standard is just different.

Validity Reporting



ESRI



OGC

Thursday, March 12, 15

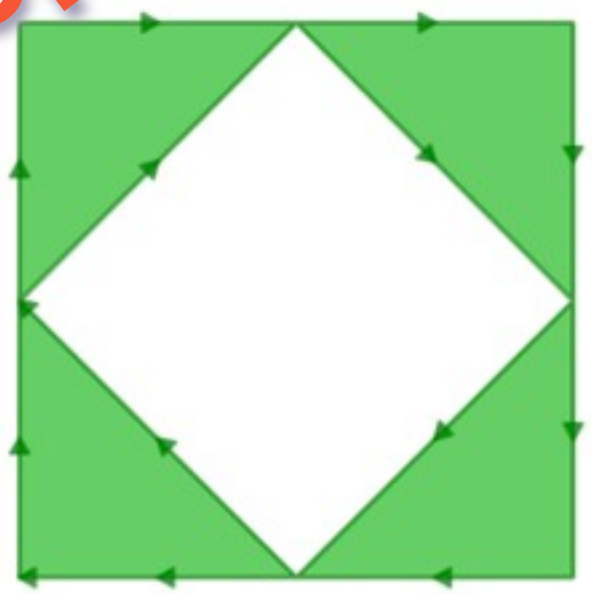
This polygon is invalid because it consists of just one ring that loops around and touches itself at the bottom. I call it a “banana polygon” because it is like a banana that has been bent until the ends touch.

The correct way to construct this shape is with an exterior and an interior ring that touch at one point.

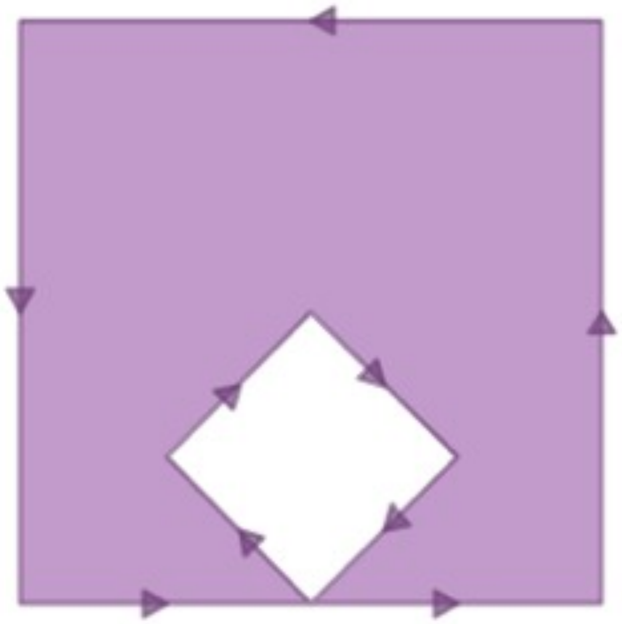
There is no “right” way to do this. ESRI actually considers the first case valid and the second one invalid. They aren’t wrong, their internal standard is just different.

Validity Reporting

ST_IsValidReason()
ST_IsValid()



Interior is disconnected[-2 0]



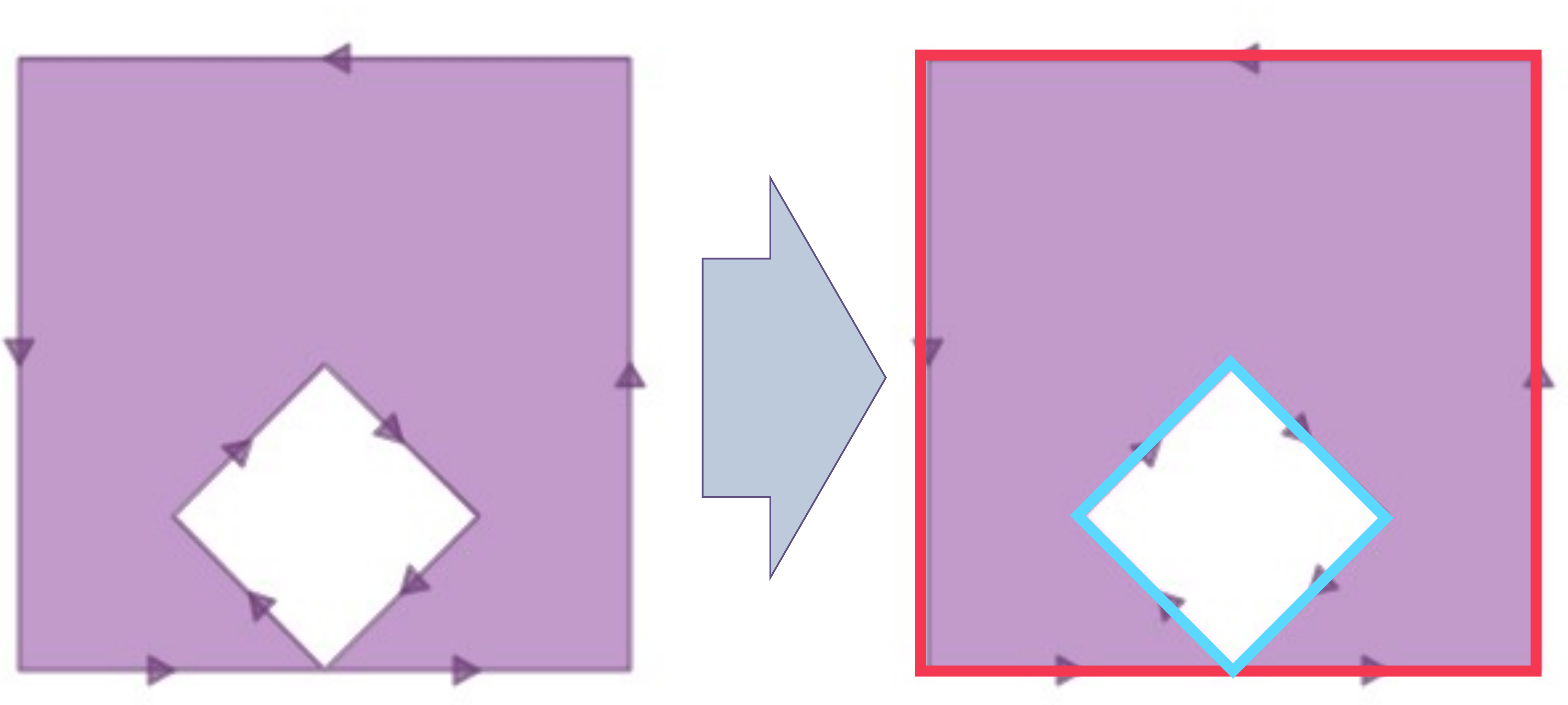
Ring Self-intersection[2 0]

Thursday, March 12, 15

We can use the ST_IsValid() function to get a boolean true/false answer for validity, and once we find invalid features, the ST_IsValidReason() function returns a text description of the invalidity. The numbers at the end are the coordinates of the invalidity. For cases that have multiple invalidity points (like the first example) only the first coordinate of invalidity is returned.

Validity Repair

ST_MakeValid()



Thursday, March 12, 15

We can fix lots of classic cases of invalidity with the ST_MakeValid() function, but unfortunately it is not available in the current releases.

Curve Distance!

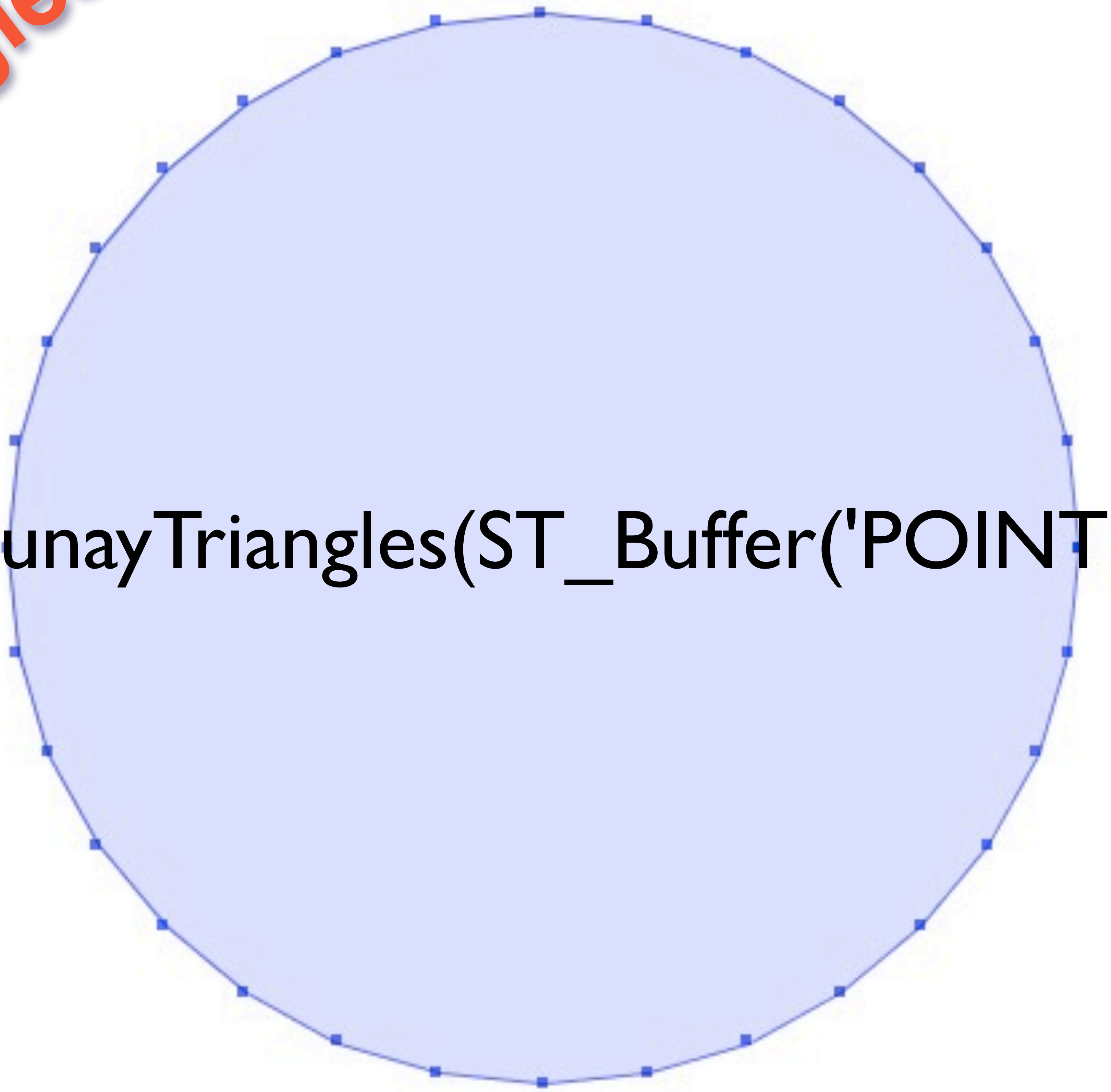
New in 2.1!

```
SELECT ST_Distance(
   ‘CIRCULARSTRING(...)’,
   ‘CURVEPOLYGON(...)’
);
```

Since 1.2, we’ve had curve types
which are part of ISO SQL/MM standard
And our curve support has been getting more complete with each release.
You can convert curves to linestrings, and even convert linestrings to curves!
Curve types are useful for storing CAD data, which uses curves

Delaunay Triangles

New in 2.1!

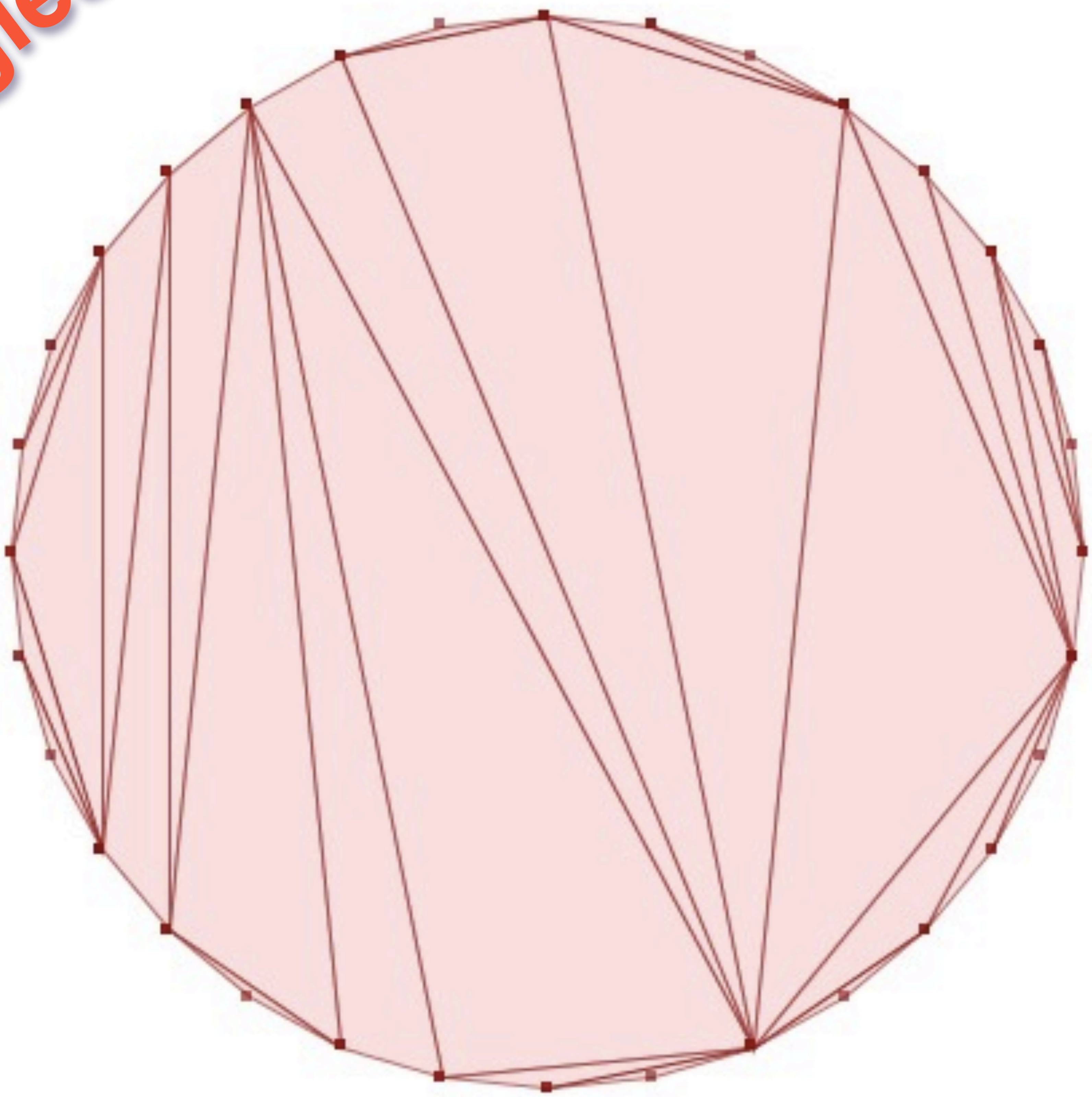


```
ST_DelaunayTriangles(ST_Buffer('POINT(0 0)', 10))
```

At version 3.4, GEOS provides a Delaunay triangulization routine, that you can access, to for example convert this circle,

Delaunay Triangles

New in 2.1!



Thursday, March 12, 15

into its equivalent triangle set. The vertices of any geometry can be used as input to the triangulation routine.

Hidden Feature

New in 2.1!

Interruptibility!!

More Performance!!

New in 2.1!

More Performance!!

New in 2.1!

- Raster ST_Union(), native implementation, **10x faster**

More Performance!!

New in 2.1!

- Raster ST_Union(), native implementation, **10x faster**
- Geography ST_Distance() and ST_DWithin(), internally indexed, **20-30x faster**

More Performance!!

New 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**

More Performance!!

New 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**

More Performance!!

New 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**

Coming soon!!

New in 2.2!

Coming soon!!

New in 2.2!

- ST_Retile(), ST_CreateOverview(), in-database overview generation

Coming soon!!

New in 2.2!

- ST_Retile(), ST_CreateOverview(), in-database overview generation
- Raster re-projection in the database

Coming soon!!

New in 2.2!

- ST_Retile(), ST_CreateOverview(), in-database overview generation
- Raster re-projection in the database
- More CGAL functions, better support (extrude, skeleton)

Coming soon!!

New in 2.2!

- ST_Retile(), ST_CreateOverview(), in-database overview generation
- Raster re-projection in the database
- More CGAL functions, better support (extrude, skeleton)
- Google encoded polyline format input/output

Coming soon!!

New in 2.2!

- ST_Retile(), ST_CreateOverview(), in-database overview generation
- Raster re-projection in the database
- More CGAL functions, better support (extrude, skeleton)
- Google encoded polyline format input/output
- GML support for curvilinear types

Coming soon!!

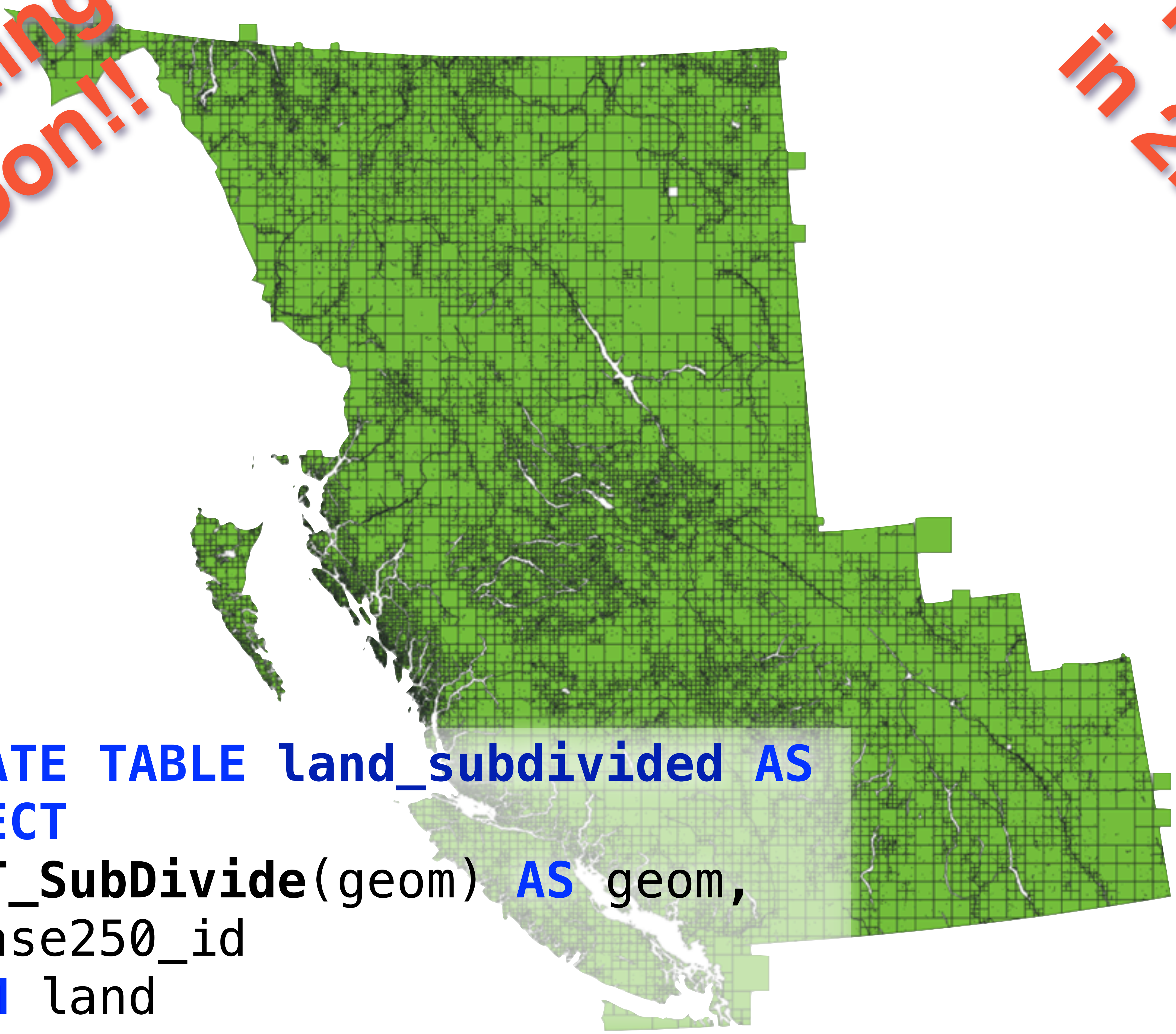
New in 2.2!

Thursday, March 12, 15

Polygon has 900 THOUSAND points in it (14MB).

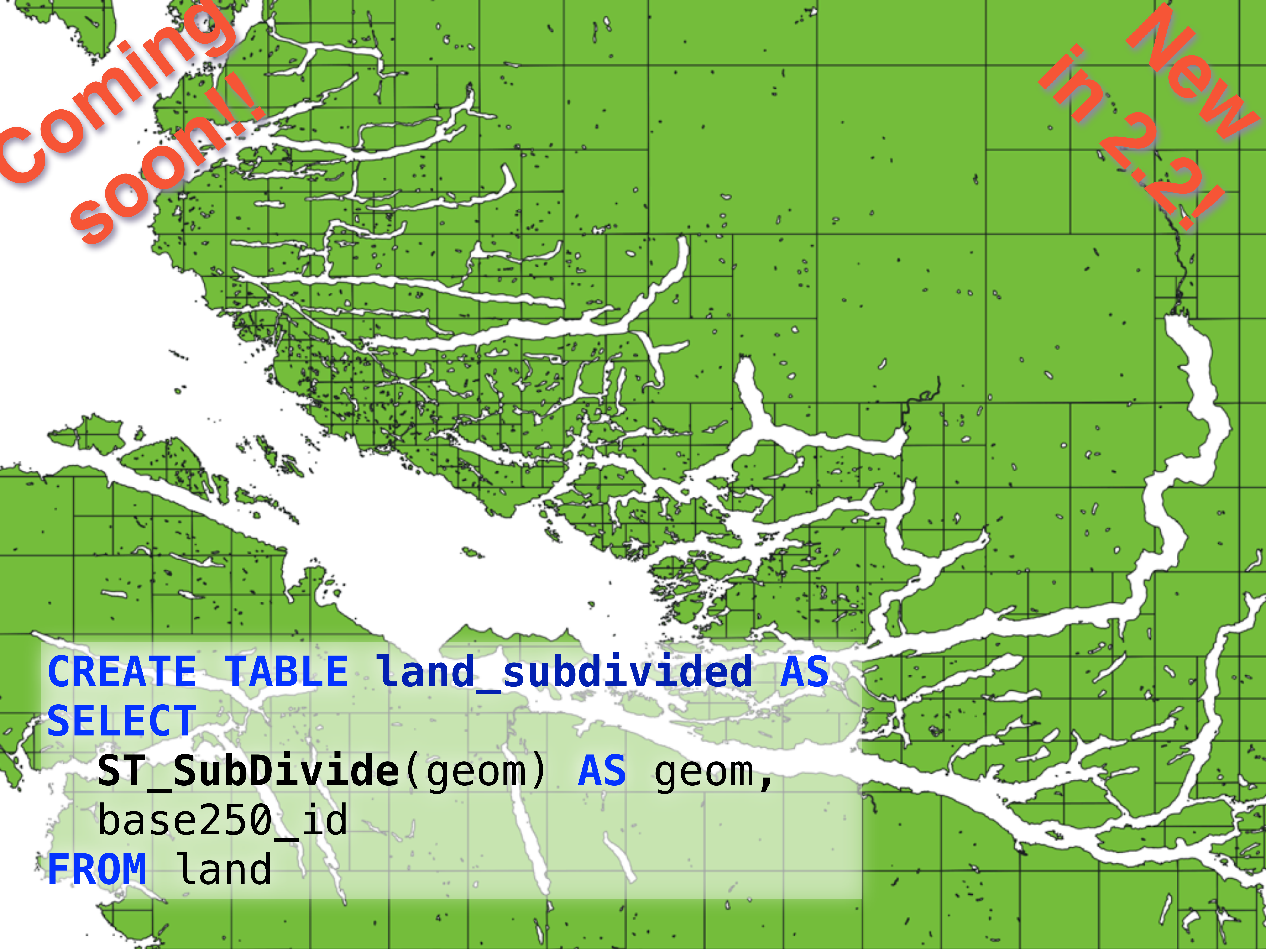
Coming soon!!

New in 2.2!



```
CREATE TABLE land_subdivided AS
SELECT
  ST_SubDivide(geom) AS geom,
  base250_id
FROM land
```

ST_Subdivide, SET RETURNING function, breaks big things into small things (by default no more than 256 vertices)



Coming soon!!

New in 2.2!

```
CREATE TABLE land_subdivided AS
SELECT
  ST_SubDivide(geom) AS geom,
  base250_id
FROM land
```

Recursive quad division until each piece is less than the maximum allowed size.
Breaking up my 14MB polygon took about 5 minutes. Could be faster, but... if you only do it once?

Coming soon??

New in 2.2??

MAYBE?

when is 2.2 going to be released

Coming soon??

New in 2.2??

- SP-GIST index binding?

MAYBE?

when is 2.2 going to be released

Coming soon??

New in 2.2??

- SP-GIST index binding?
- KNN-GIST recheck support? (PgSQL 9.5+)

MAYBE?

when is 2.2 going to be released

Coming soon??

New in 2.2??

- SP-GIST index binding?
- KNN-GIST recheck support? (PgSQL 9.5+)
- X/Y/Z temporal data helper functions

MAYBE?

Thursday, March 12, 15

when is 2.2 going to be released

Coming soon??

New in 2.2??

- SP-GIST index binding?
- KNN-GIST recheck support? (PgSQL 9.5+)
- X/Y/Z temporal data helper functions
- GML support for curvilinear types

MAYBE?

Thursday, March 12, 15

when is 2.2 going to be released

Coming soon??

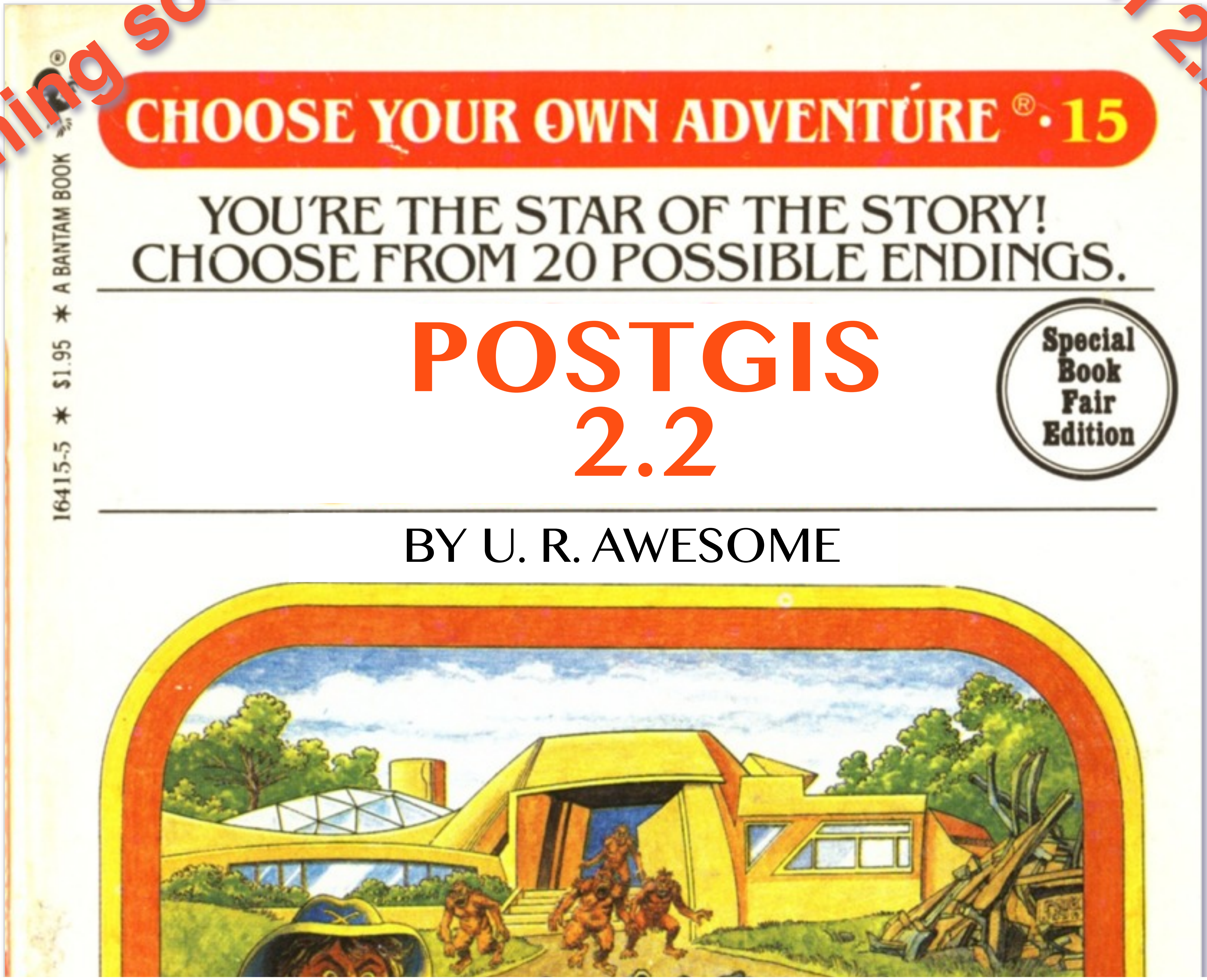
New in 2.2??

- SP-GIST index binding?
- KNN-GIST recheck support? (PgSQL 9.5+)
- X/Y/Z temporal data helper functions
- GML support for curvilinear types

when is 2.2 going to be released

Coming soon!!

New in 2.2!



CHOOSE YOUR OWN ADVENTURE ® · 15

YOU'RE THE STAR OF THE STORY!
CHOOSE FROM 20 POSSIBLE ENDINGS.

POSTGIS 2.2

BY U. R. AWESOME

Special Book Fair Edition

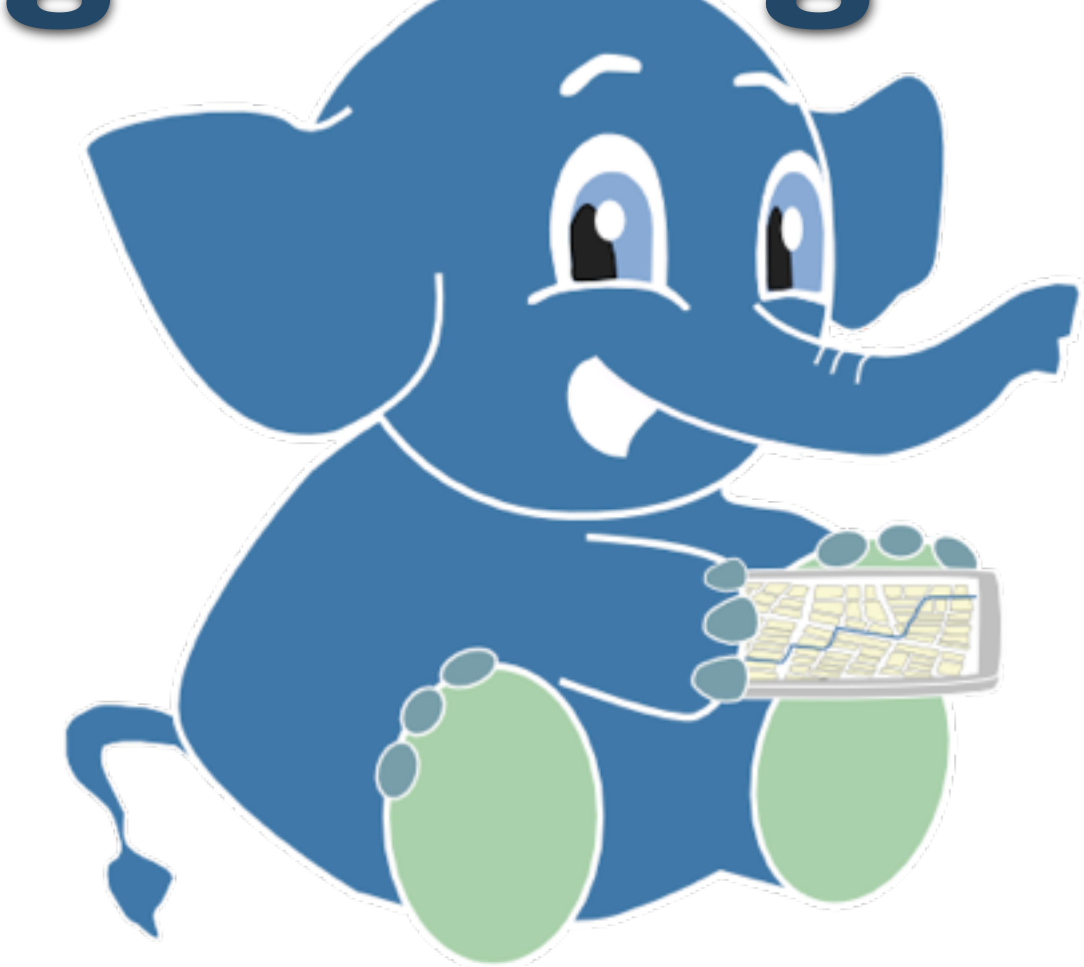
16415-5 ★ $1.95 ★ A BANTAM BOOK

what is in 2.2 is up to you

github.com/pgpointcloud

not strictly postgis, but postgis-related...!

pgrouting.org



pgRouting: A Practical Guide

http://locatePRESS.com/pgrouting

Open Source "Geo" Books & Training

All glory to the Hypnotoad!

Thursday, March 12, 15

or up to you and your trusted development partner

PostGIS

is a

Spatial Database

no matter what, PostGIS has had a great first 14 years, and is looking strong for the next 14

PostGIS

is ~~a~~ **the**

Spatial Database

Thursday, March 12, 15

no matter what, PostGIS has had a great first 14 years, and is looking strong for the next 14

Do you have questions about...?

PostGIS Feature Frenzy!!!

PRAMSEY

CARTODB.COM