PostGIS 2.0 ...

... and beyond

Vincent Picavet – Oslandia

PostGIS 2.0

PostGIS 2.0.0 : April 3, 2012

After 26 months !

Major version

Breaks compatibility

Loads of new features

Performance improvement

OSLANDIA

What's new ?

OSLANDIA

Features

Management fonctions

ISO SQL/MM compliancy

New functions for analysis

Topology (SQL/MM)

Real 3D storage

Raster / geometry functions

KNN indexed search

Install

Easier installation (PG >= 9.1)

CREATE EXTENSION postgis ;

CREATE EXTENSION postgis_topology ;

Manage

geometry_columns → view

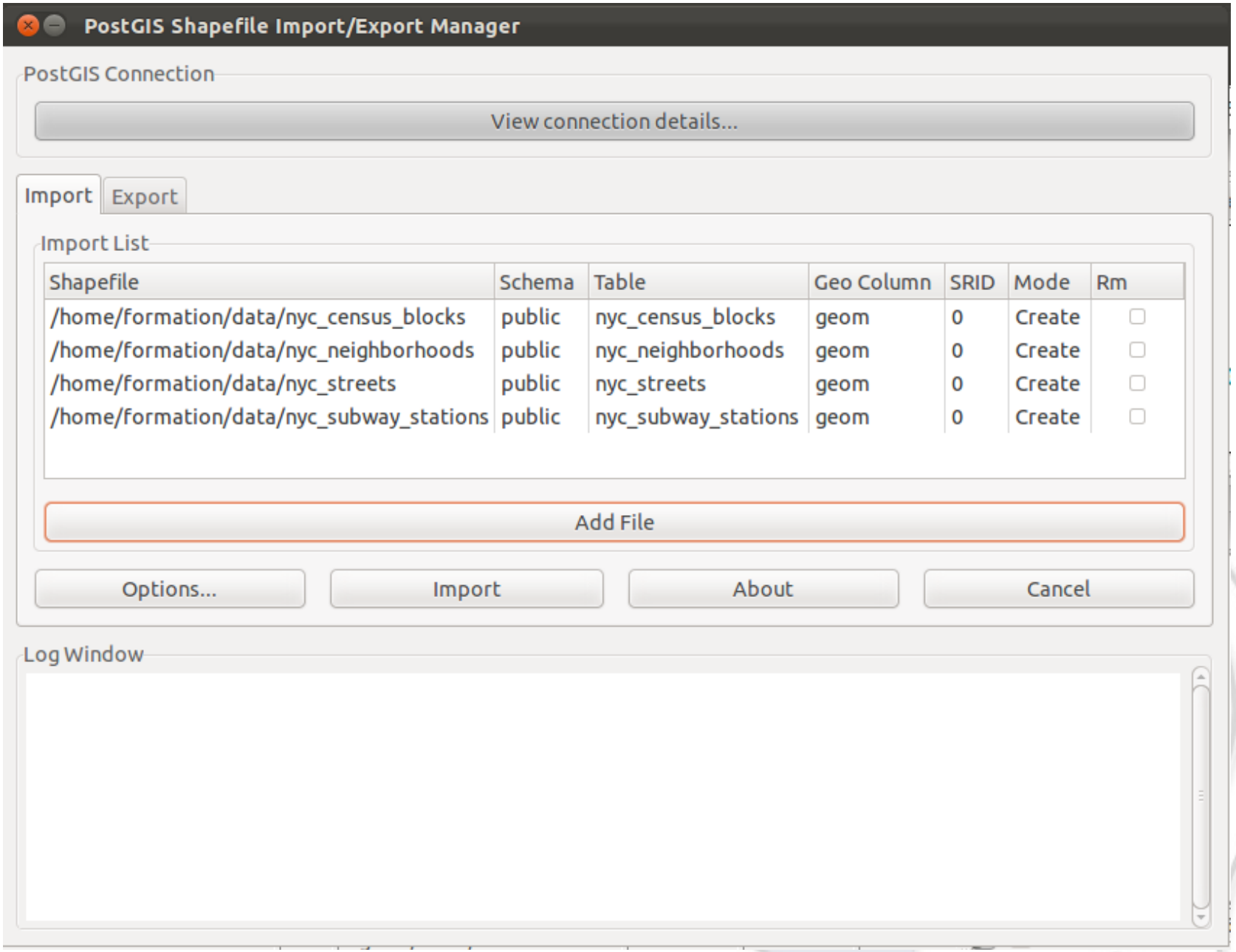
Typmod usage

Old way still available

```
CREATE TABLE buildings (
        gid SERIAL PRIMARY KEY
        , geom geometry(MultiPolygon, 26986)
);

alter table buildings
        alter column geom
                type geometry(MultiPolygon, 2154)
                using st_setsrid(geom, 2154);
```

Load


PostGIS Shapefile Import/Export Manager
PostGIS Connection
View connection details...
Import
Export
Import List
Shapefile
Schema
Table
Geo Column
SRID
Mode
Rm
/home/formation/data/nyc_census_blocks
public
nyc_census_blocks
geom
0
Create
/home/formation/data/nyc_neighborhoods
public
nyc_neighborhoods
geom
0
Create
/home/formation/data/nyc_streets
public
nyc_streets
geom
0
Create
/home/formation/data/nyc_subway_stations
public
nyc_subway_stations
geom
0
Create
Add File
Options...
Import
About
Cancel
Log Window

Functions

ST_ConcaveHull

ST_Snap

ST_Split

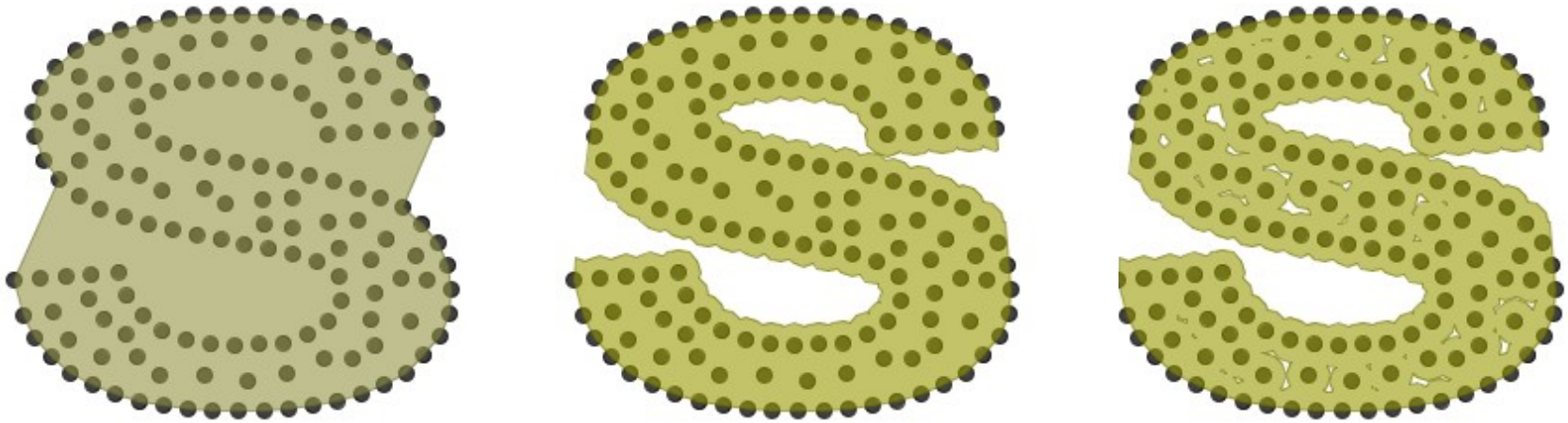
ST_MakeValid

ST_IsValidDetail

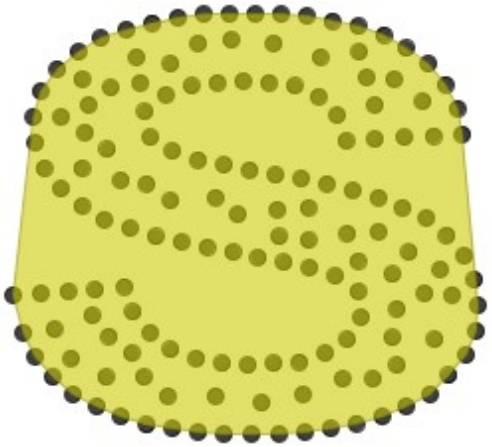
ST_OffsetCurve

...

hulls and curves



Concave hulls with various settings



Convex hull

Offset curves

OSLANDIA

cleaning data

Before : ST_Buffer(the_geom, 0)

After :

- ST_MakeValid()
- ST_RemoveRepeatedPoints()
- ST_isValidReason()
- ST_IsValidDetail()

```
SELECT ST_IsValid(geom),ST_IsValidReason(geom) FROM
 (SELECT ST_GeomFromText('POLYGON ((0 0, 0 10, 10 10, 10 0, 0 0),(20 20, 20 30, 30 30, 30 20, 20 20))') as geom) as foo;
 st_isvalid |                    st_isvalidreason
------------+---------------------------------------------------------
 f          | Hole lies outside shell at or near point (20.0, 20.0, NaN)
```

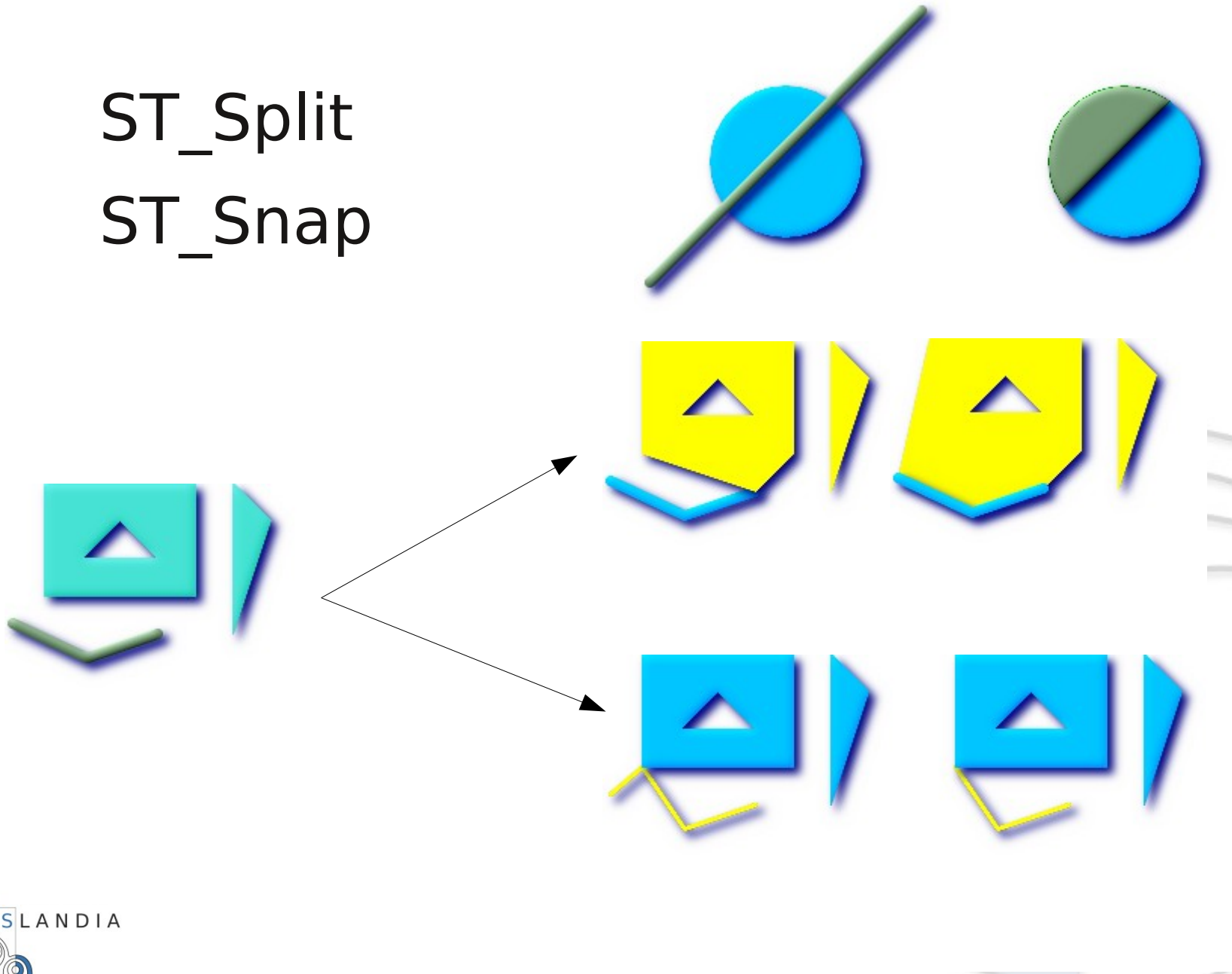
```
SELECT * FROM ST_IsValidDetail('LINESTRING(...)');
        gid  |      reason       |  location
       ------+-------------------+------------
        5330 | Self-intersection | POINT(32 5)
```

OSLANDIA

Splitting and snapping

ST_Split

ST_Snap



OSLANDIA

Topology

Beware of the spaghetti monster !

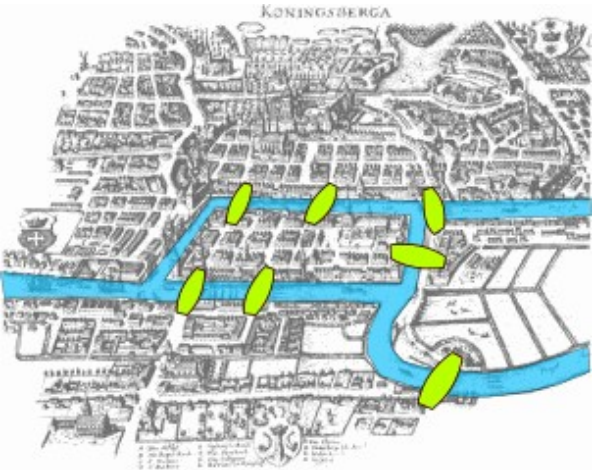
Topology - Graphs

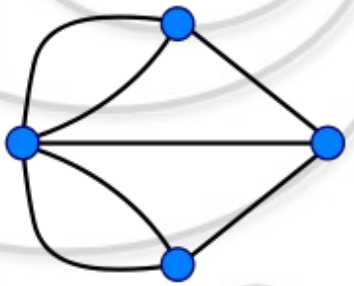
Explicit relations between objects

Graph representation

Various models

- OGC : Node / edge / face





Topology

Node/Edge/Face model

TopoGeometry Datatype

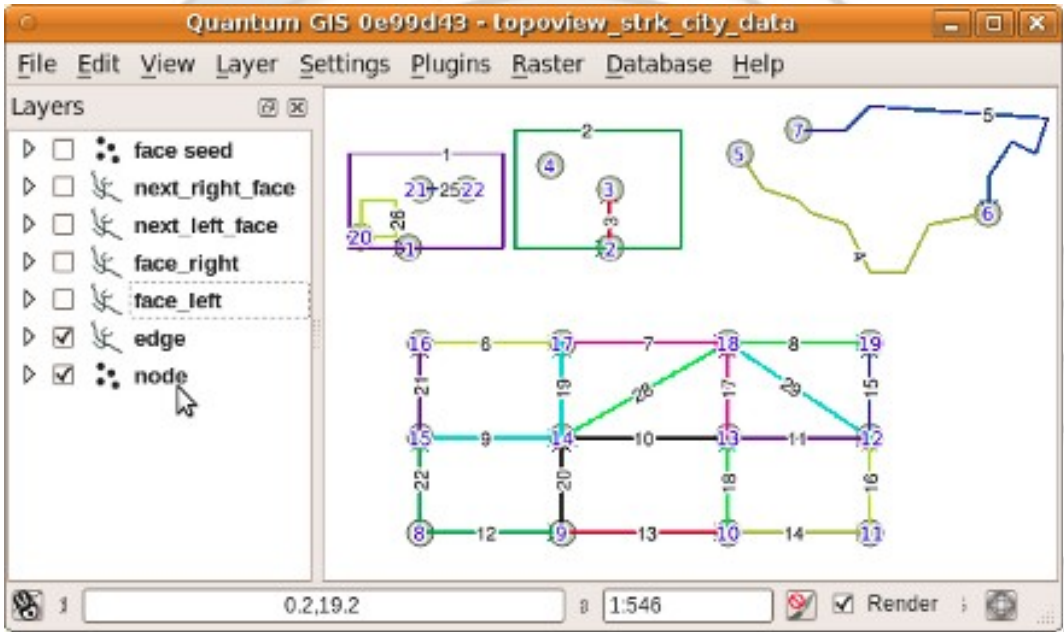
Use schemas

- «topology» for functions and others
- Each topology in its own schema

Full SQL/MM topology support

Integrated in 2.0

- Sandro Santilli
- Toscane Region



OSLANDIA

Topology

Reduced storage size

Explicit spatial relationships

Without topology !

Topology use case

Table name : **tr**

Fichier Éditer Vue Couche Préférences Extension Vecteur Base de donnée Raster Aide

Couches

- recursive_upstream_topo
- recursive_upstream
- shortest_path_topology
- shortest_path_pgrouting
- hydro network
- background

Contrôle de l'ordre de rendu des couches

Attribute table - hydro network :: 0 / 18936 feature(s) selected

	gid	source	target	hname	cost
0	17681	3042	3041	ruisseau de...	13.1468627...
1	50006	4363	4376	ruisseau de...	154.831357...
2	107308	4427	4443	ruisseau la ...	70.4784694...
3	110767	4810	4816	ruisseau le ...	426.452159...
4	8923	4892	4827	ruisseau de...	1648.21133...
5	109594	5158	5264	rivière la di...	946.014083...
6	45039	5407	5429	NULL	114.028638...
7	105937	5480	5594	ruisseau le ...	824.626701...
8	104620	5481	5518	ruisseau la ...	243.004034...

```
create table
        rec_res2 as
with recursive
        search_graph(edge_id, start_node, depth, path, length, cycle) as (
                select
                        g.edge_id, g.start_node, 1 as depth, ARRAY[g.edge_id] as path
                        , st_length(geom) as length, false as cycle
                from
                        hydro.edge as g
                where
                        edge_id = 173832
                union all
                select
                        g.edge_id
                        , g.start_node
                        , sg.depth + 1 as depth
                        , path || g.edge_id as path
                        , sg.length + st_length(g.geom) as length
                        , g.edge_id = ANY(path) as cycle
                from
                        hydro.edge as g
                join
                        search_graph as sg
                on
                        sg.start_node = g.end_node
                where
                        not cycle
                )
```

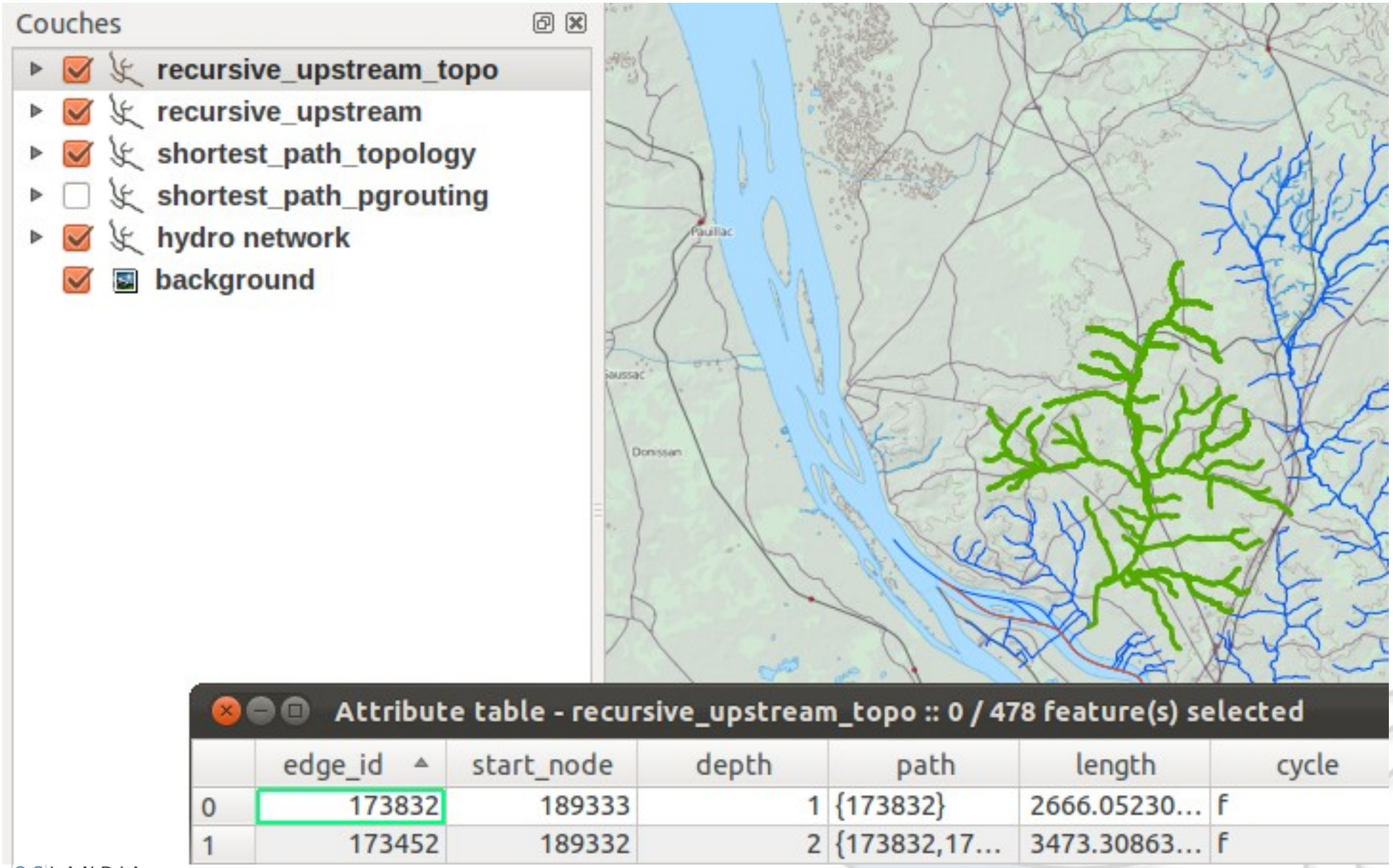
Recursive CTE

1

2

```
select
        sg.*
        , edge.geom as geom
from
        search_graph as sg
join
        hydro.edge as edge
on
        sg.edge_id = edge.edge_id
limit 1000;
```

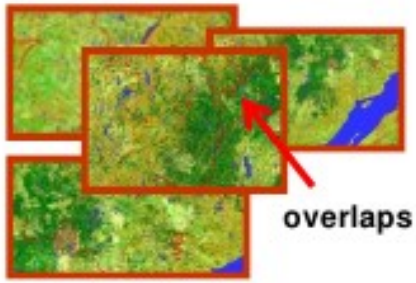
3


Couches
recursive_upstream_topo
recursive_upstream
shortest_path_topology
shortest_path_pgrouting
hydro network
background
Pauillac
Donnissan
Attribute table - recursive_upstream_topo :: 0 / 478 feature(s) selected
edge_id
start_node
depth
path
length
cycle
0
173832
189333
1
{173832}
2666.05230…
f
1
173452
189332
2
{173832,17…
3473.30863…
f

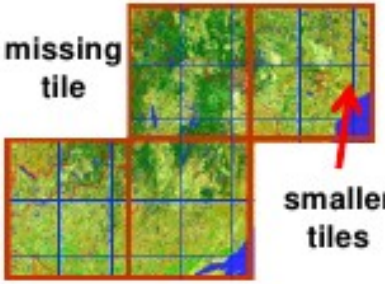
PostGIS Raster

- Raster / vector analysis
- New datatype
  - Looks like geometry
  - But for rasters
- Multiresolution, multiband, tile coverage
- Import/export (GDAL)
- Functions
  - Statistics, reprojection, edit, compute
  - Vector/raster functions

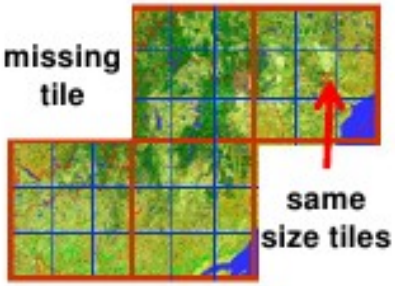
PostGIS Raster



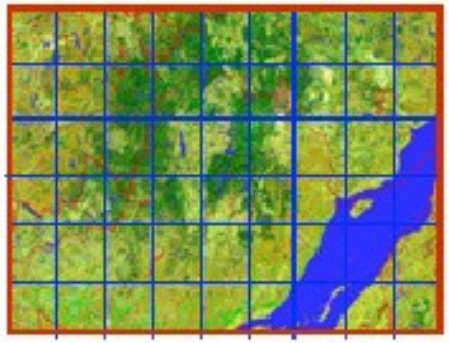
a) warehouse of untiled and unrelated images (4 images)



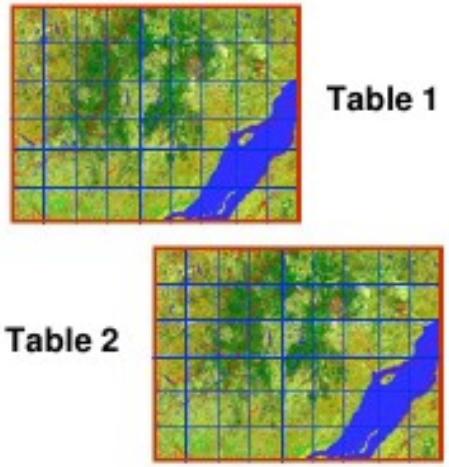
b) irregularly tiled raster coverage (36 tiles)



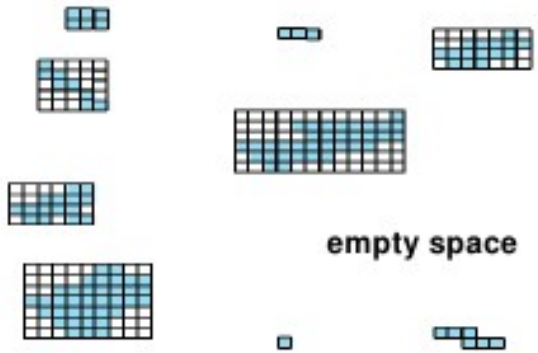
c) regularly tiled raster coverage (36 tiles)



d) rectangular regularly tiled raster coverage (54 tiles)



e) tiled images (2 tables of 54 tiles)

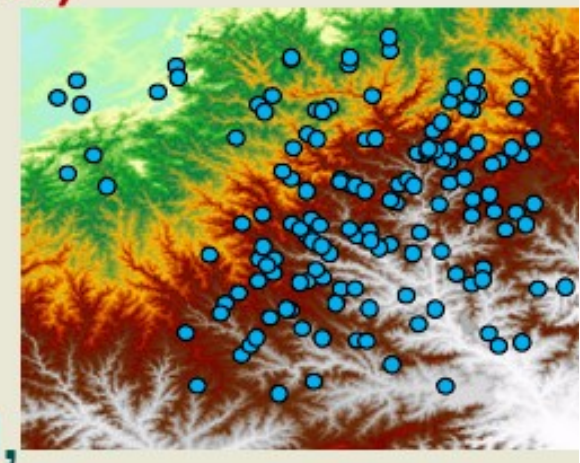


f) rasterized geometries coverage (9 lines in the table)

PostGIS 2.0 : PostGIS Raster

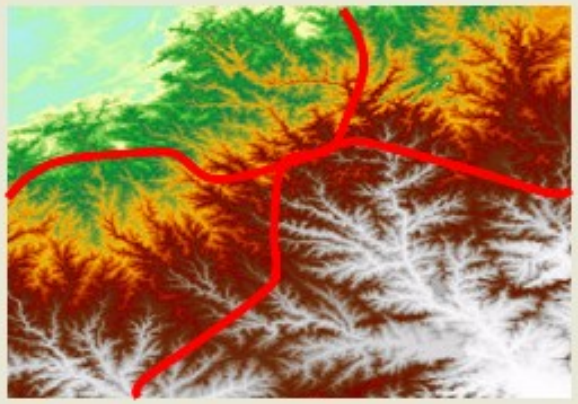
Extract ground elevation values for lidar points…

- SELECT pointID, ST_Value(rast, geom) elevation
  FROM lidar, srtm WHERE ST_Intersects(geom, rast)



Intersect a road network to extract elevation values for each road segment

- SELECT roadID,
  (ST_Intersection(geom, rast)).geom road,
  (ST_Intersection(geom, rast)).val elevation
  FROM roadNetwork, srtm WHERE ST_Intersects(geom, rast)



PostGIS 2.0 : nearest neighbours

KNN-GIST search in PostgreSQL 9.1

Use indexes !

Spatial nearest neighbors

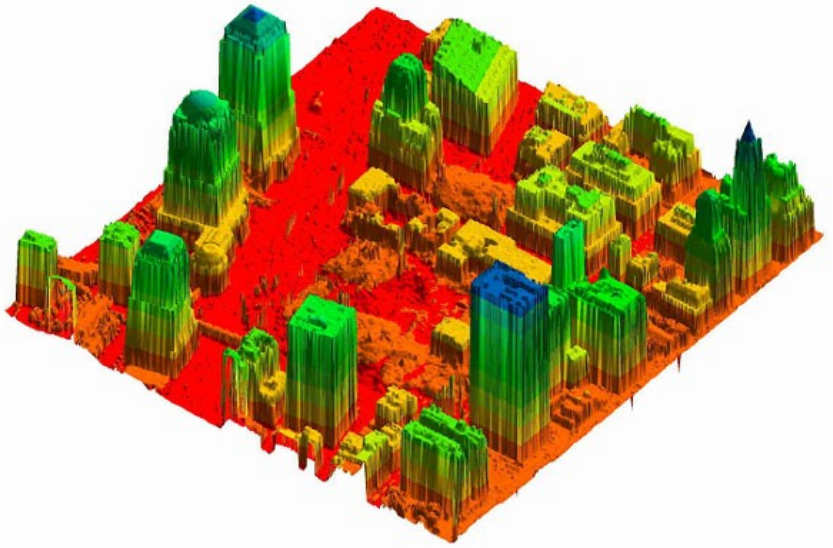
```
SELECT name, gid FROM geonames
ORDER BY
    geom <-> st_setsrid(st_makepoint(-90,40),4326)
LIMIT 10;
```

Distance operator

<-> or <#> : center or bbox (index)

Need to refine for non-point geometries

Point cloud (independant / 2.1)



- LIDAR data
- As a PG extension
  - And a PostGIS extension
- Points and patches of po
- Arbitrary number of dimensions
  - (Use of point cloud « schema »)
- Compression storage
- Link with PDAL

Point cloud

Spatial filtering

```
SELECT PC_AsText(PC_Explode(PC_Intersection(
      pa,
      'SRID=4326;POLYGON((-126.451 45.552, -126.42
47.55, -126.40 45.552, -126.451 45.552))'::geometry
)))
FROM patches WHERE id = 7;

                pc_astext
-----------------------------------------
 {"pcid":1,"pt":[-126.44,45.56,56,5]}
 {"pcid":1,"pt":[-126.43,45.57,57,5]}
 {"pcid":1,"pt":[-126.42,45.58,58,5]}
 {"pcid":1,"pt":[-126.41,45.59,59,5]}
```

PostGIS 2.1

PostGIS 2.1

Beta out, release soon

Arc-geometry distance

Distance with cached tree

R-Tree index improvement (pick-split)

SP-Gist Index :

- New in PG 9.1
- Up to 3x faster construction
- Faster to read

ST_MapAlgebra with N rasters

PostGIS 2.1

And more...

Topology improvement

Tiger geocoder as PG extension

PgRouting as PG extension

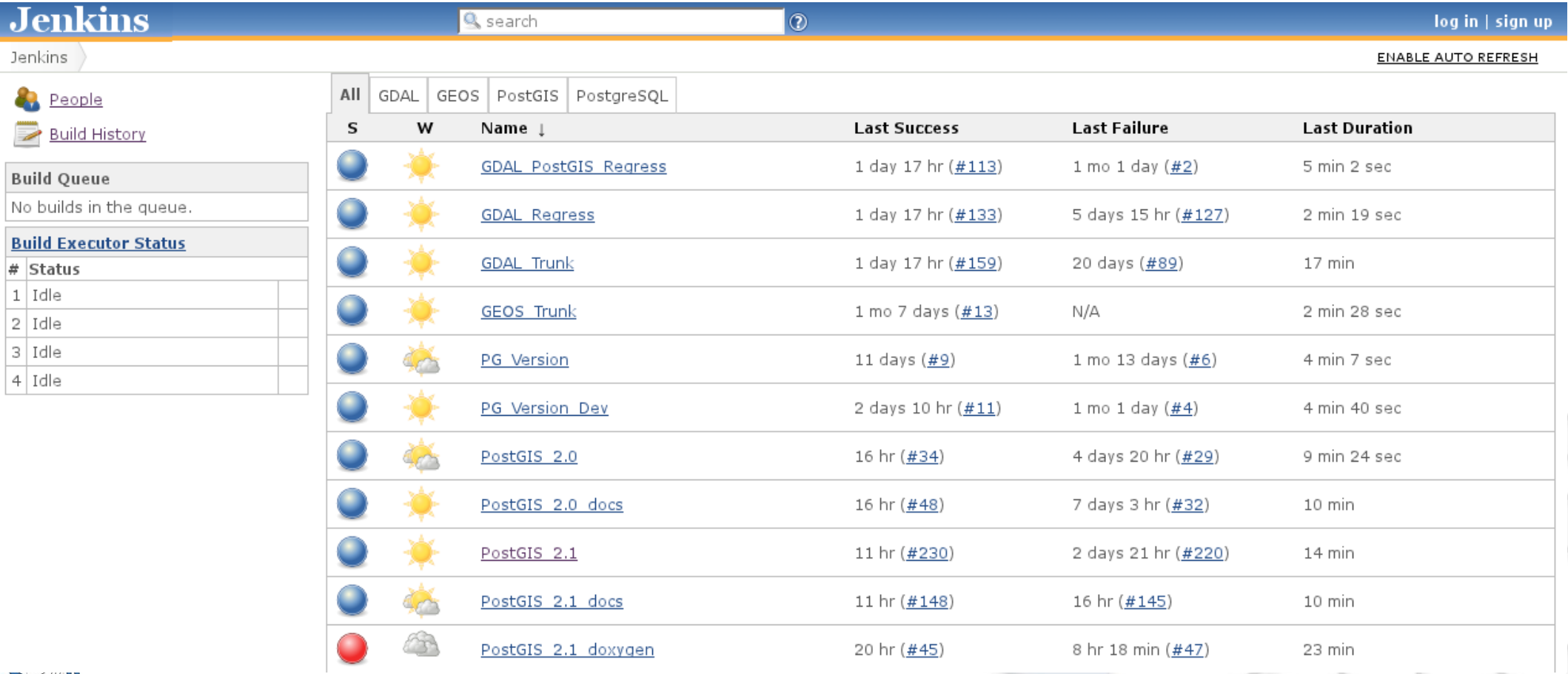
+ windows support

PostGIS 2.1

And even more...

Development platform improvement

Meet Debbie and Winnie !



PostGIS 2.1

[00:40] <debbie> Project PostGIS_2.0 build #34: SUCCESS in 9 min 21 sec: http://debbie.postgis.net:8080/job/Post
[00:40] <debbie> Paul Ramsey: (#2026) fix performance regression in geography distance calculation
[00:40] <sigq> Title: PostGIS_2.0 #34 [Jenkins] (at debbie.postgis.net:8080)
[00:44] <pramsey> done!
[00:47] <debbie> Project PostGIS_2.1 build #227: SUCCESS in 26 min: http://debbie.postgis.net:8080/job/PostGIS_
[00:47] <debbie> Paul Ramsey: (#2026) fix performance regression in geography distance calculation
[00:47] <sigq> Title: PostGIS_2.1 #227 [Jenkins] (at debbie.postgis.net:8080)
[00:51] --> tomkralidis a rejoint ce canal (~tomkralid@CPE0013ce450e14-CM001692413c80.cpe.net.cable.rogers.co
[00:51] <-- tomkralidis a quitté ce serveur (Changing host).
[00:51] --> tomkralidis a rejoint ce canal (~tomkralid@osgeo/member/tomkralidis).
[00:51] <debbie> Project PostGIS_2.1_docs build #145: FAILURE in 4 min 0 sec: http://debbie.postgis.net:8080/job/P
[00:51] <debbie> * Bborie Park: Added news and docs for ST_Tile(raster). Additional regression tests for
[00:51] <debbie> one additional variant of ST_Tile(raster)
[00:51] <sigq> Title: PostGIS_2.1_docs #145 [Jenkins] (at debbie.postgis.net:8080)
[00:51] <debbie> * Bborie Park: Added ST_Tile() and regression tests. The circle is complete.
[00:51] <debbie> * Bborie Park: Added rt_band_get_pixel_line() and regression tests
[00:51] <debbie> * Paul Ramsey: (#2063) fix the vertex-crossing logic in the circular tree code to use the new edge
[00:51] <debbie> * Paul Ramsey: (#2026) fix performance regression in geography distance calculation
[00:52] --> tbowden a rejoint ce canal (~tim@124-148-118-242.dyn.iinet.net.au).
[00:54] <-- epifanio a quitté ce serveur (Read error: Operation timed out).
[00:58] <winnie> Project PostGIS 2.1 mingW regress build #456: STILL FAILING in 2 min 30 sec: http://winnie.postg
[00:58] <sigq> Title: PostGIS 2.1 mingW regress #456 [Jenkins] (at winnie.postgis.net:1500)

+ Hallie :
documentation bot (PostgreSQL FTS & more)

PostGIS x.y : next dimension

3D

3D

New types :

- TRIANGLE, POLYHEDRALSURFACE, TIN

New functions :

- ST_3DDistance, ST_3DIntersects, ST_3DDWithin, ST_3DClosestPoint...
- Input/Output : ST_AsGML, ST_AsX3D...

New operators

- &&&

Spatial index : nd-indexes

Real 3D

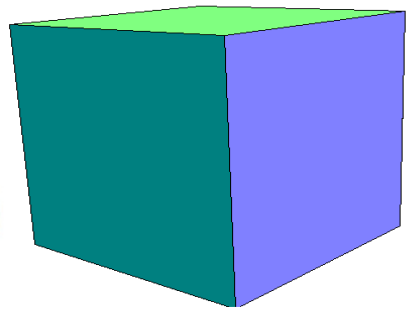
«real» 3D inside PostGIS

ISO and OGC standards

- ISO 19125, SQL/MM, SFS 1.2.0

First step of implementation

New data types & functions



PostGIS 3D

2.5D already in

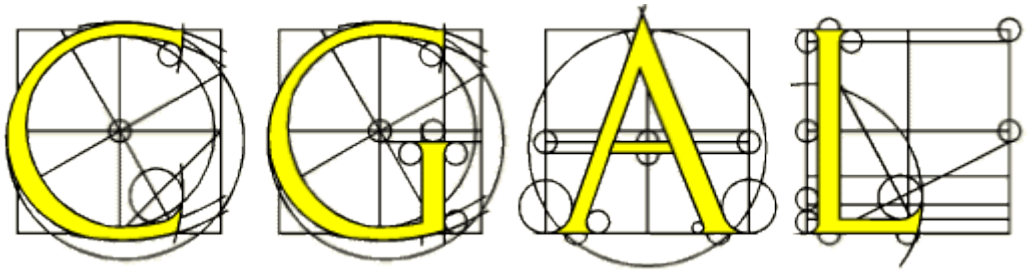
3D storage is in

We want analysis !

- ST_3Dintersects
- ST_3Dintersection
- ST_Extrude (2D → 3D)
- ST_3Dconvexhull
- ST_StraightSkeleton
- ST_Tesselate...

OSLANDIA

PostGIS 3D



2D & 3D geometric computation

C++

Exact computation

Efficient, generic, extensible...

Became GPL → PostGIS compatibility

PostGIS 3D

Some European funding

e-PLU

Cooperation with IGN & others

Creation of SFCGAL framework (OGC-SF)

Use it in PostGIS

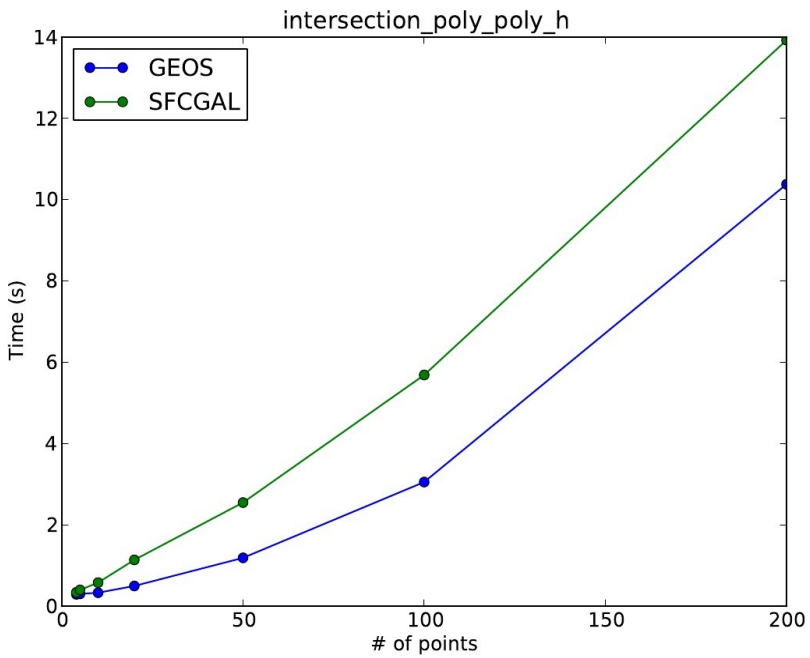
Compare with GEOS (for 2D)

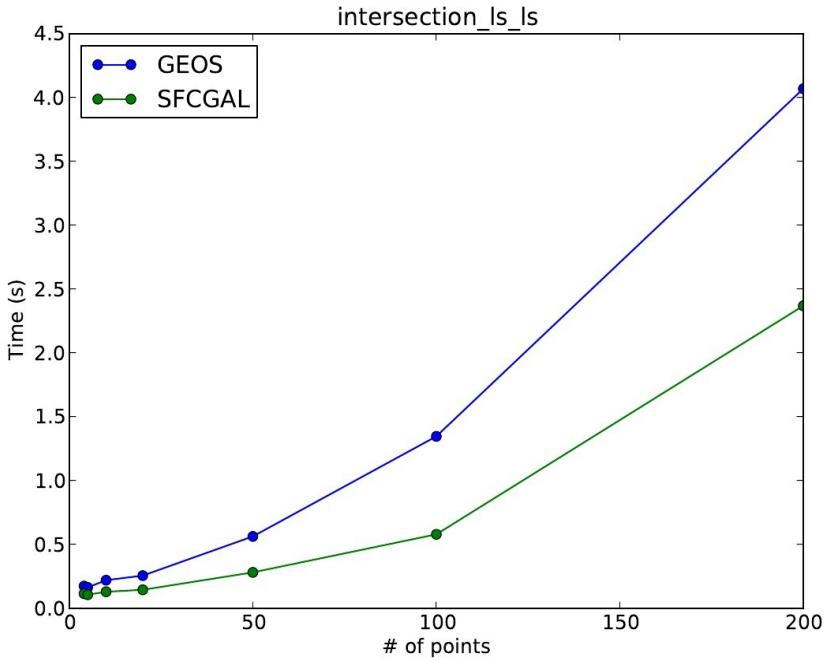
OSLANDIA

PostGIS 3D

Performance comparison with GEOS

→ Comparable performances





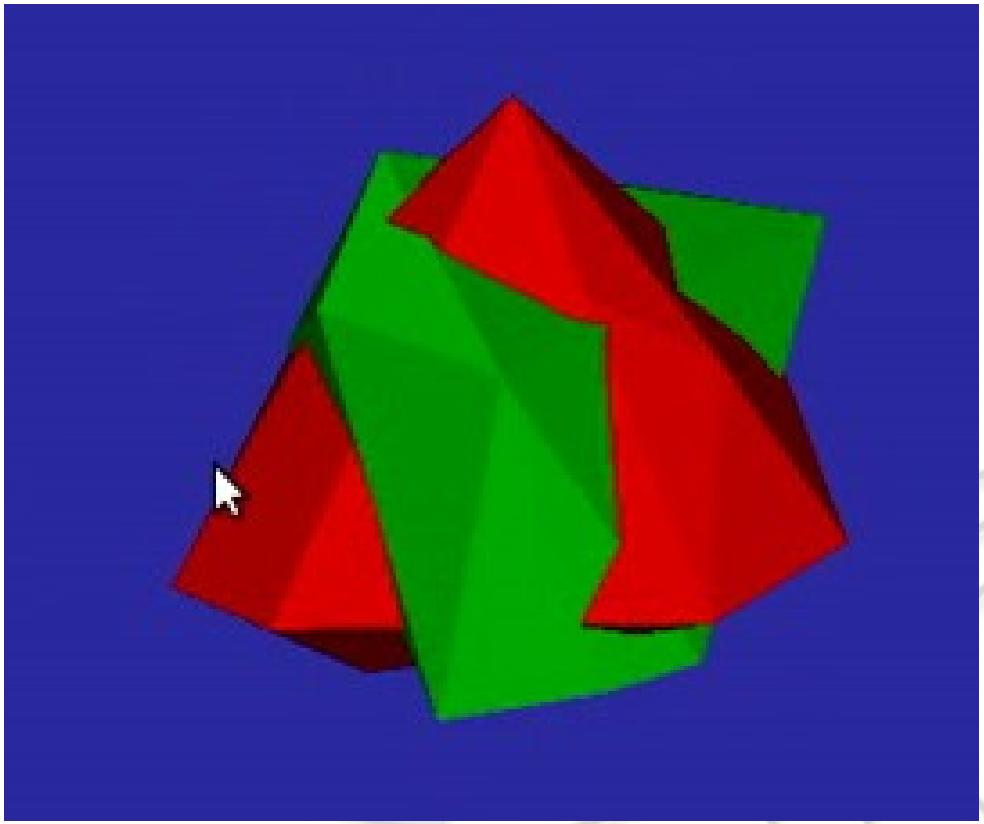
Recent progress

Quantum GIS client (Globe)

Some analysis functions

- ST_Extrude
- ST_3DConvexhull
- ST_3DIntersection
  - Surfaces
  - Solids

In PostGIS 2.1



OSLANDIA

Want to see ?

And beyond ?

OSLANDIA

3D Next steps

More features from CGAL

- Alpha shapes
- 3D Minkowski sum
- 3D snap rounding ?
- ...

Better QGIS support

CityGML & Collada loaders / exporters

Textures ?

Find €€€€€ to speed up development

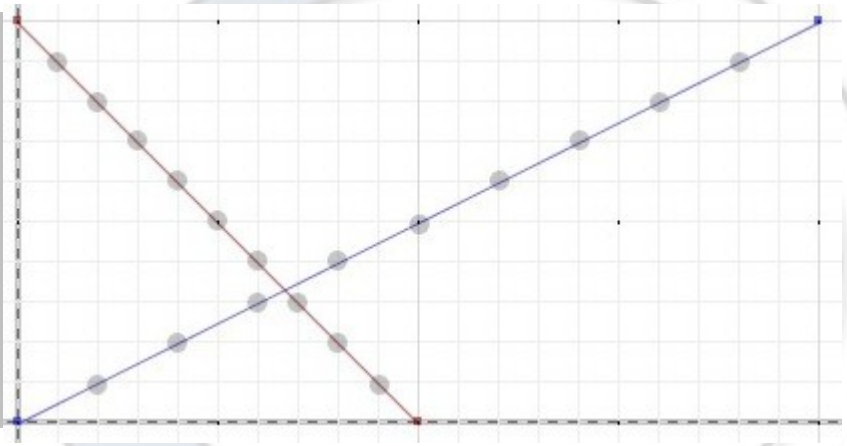
PostGIS 3D : next steps

CGAL : exact computations

New objects : exact geometries

- Try to avoid serialization
- (PostgreSQL patch)

```
SELECT
  ST_Intersects(
    ST_Intersection(
      'LINESTRING(0 0,2 1)'::geometry,
      'LINESTRING(1 0,0 1)'::geometry),
    'LINESTRING(0 0,2 1)'::geometry);
 st_intersects
---------------
 f
(1 row)
```



OSLANDIA

PostGIS 3.0 ?

- Paris codesprint and barcamp May 2012
- Boston codesprint March 2013
- Find directions for future
  - Git, build system (cmake, mainly Windows)
  - Geometry backend (GEOS vs BGL vs ?)
  - Raster improvement
  - 3D topology & processing (CGAL?)
  - Performance, performance, performance

PostgreSQL 9.3

Better replication

LATERAL JOIN

Awesome JSON support

→ PostgreSQL as document DB

Materialized views

+ recursive + updatable views

Writeable Foreign tables (FDW)

That's it...

Questions ?

vincent.picavet@oslandia.com

http://www.oslandia.com

http://ww.github.com/Oslandia