PostGIS 2.0 ...

... et au delà

Hugo Mercier – Oslandia

OSLANDIA

Oslandia

- Experts SIG Open Source
  - PostGIS
  - QGIS
  - TinyOWS
  - ...
- Conseil
- Formation
- Développement

PostGIS 2.0

PostGIS 2.0.0 : 3 avril, 2012

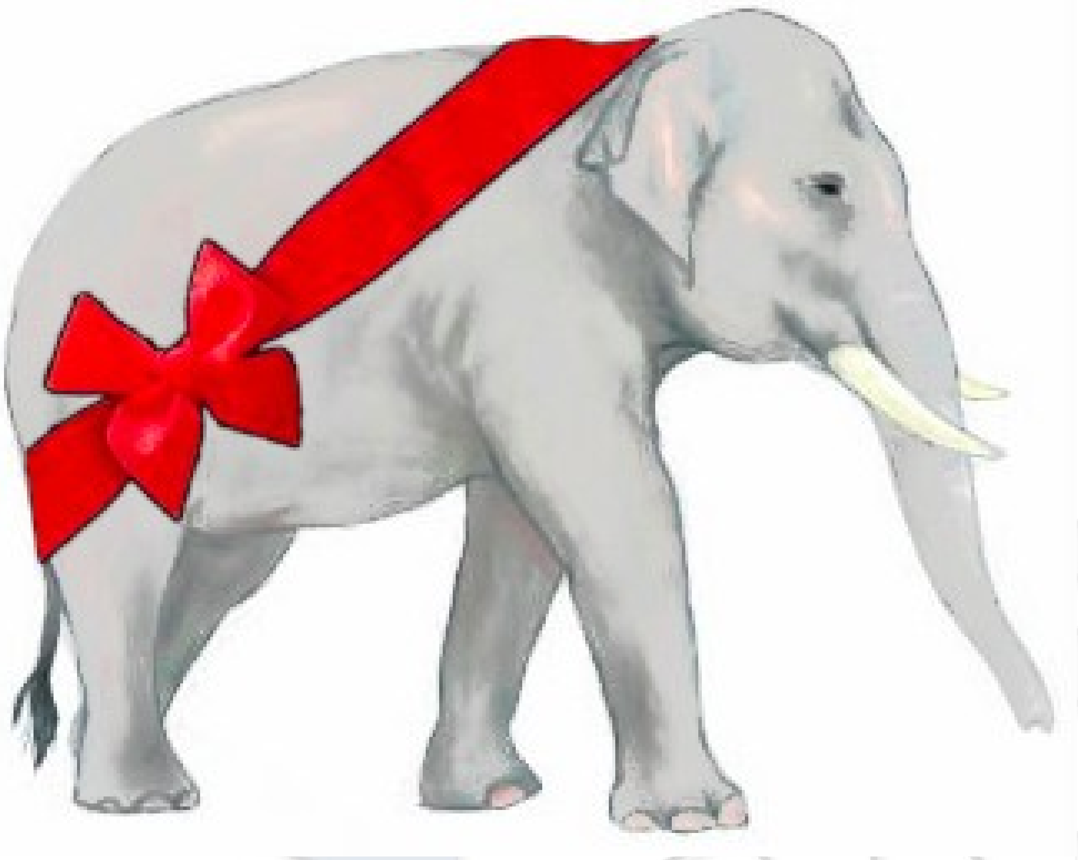
Après 26 mois !

Version majeure

Casse la compatibilité

Nombreux ajouts

Améliorations des performances



Quoi de neuf ?

Features

Administration

Compatibilité ISO SQL/MM

Fonctions d'analyse

Topologie (SQL/MM)

Stockage 3D (et analyse)

Fonctions raster / géométrie

KPPV indexé

Installation

Installation facilitée (PG >= 9.1)

CREATE EXTENSION postgis ;

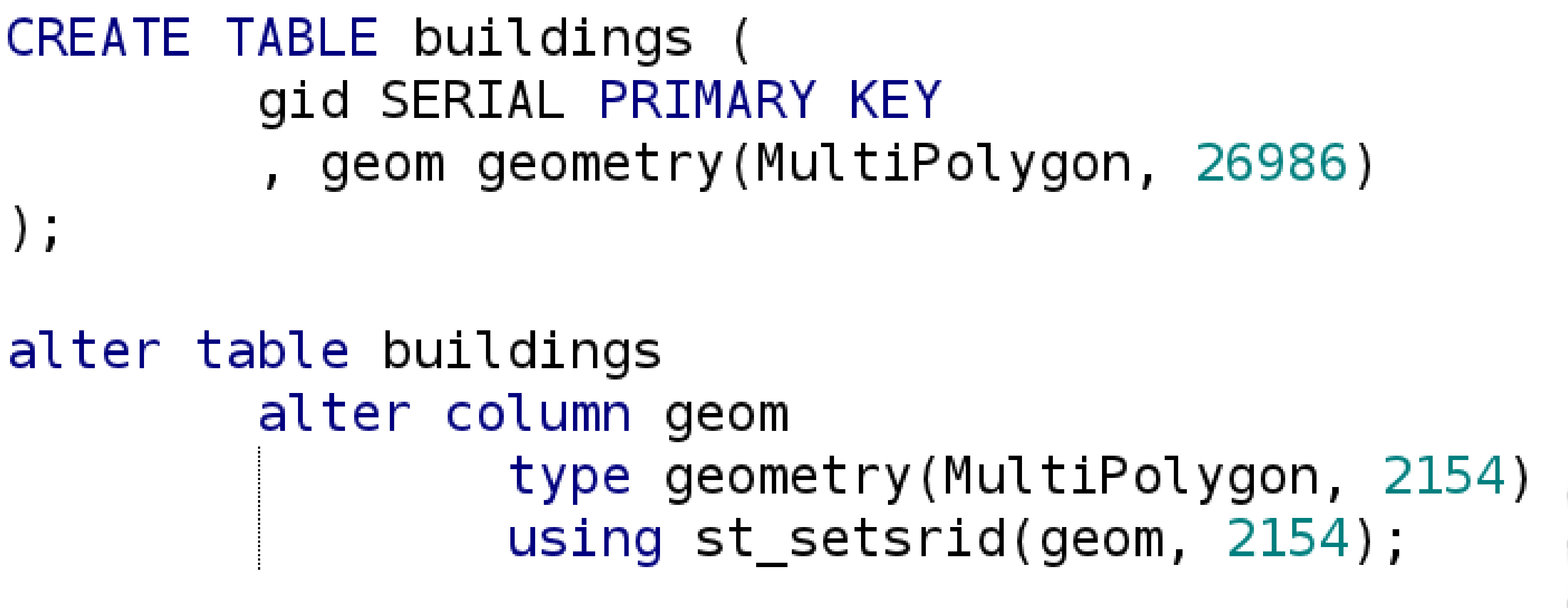
CREATE EXTENSION postgis_topology ;

Mises à jour facilitées

Administration

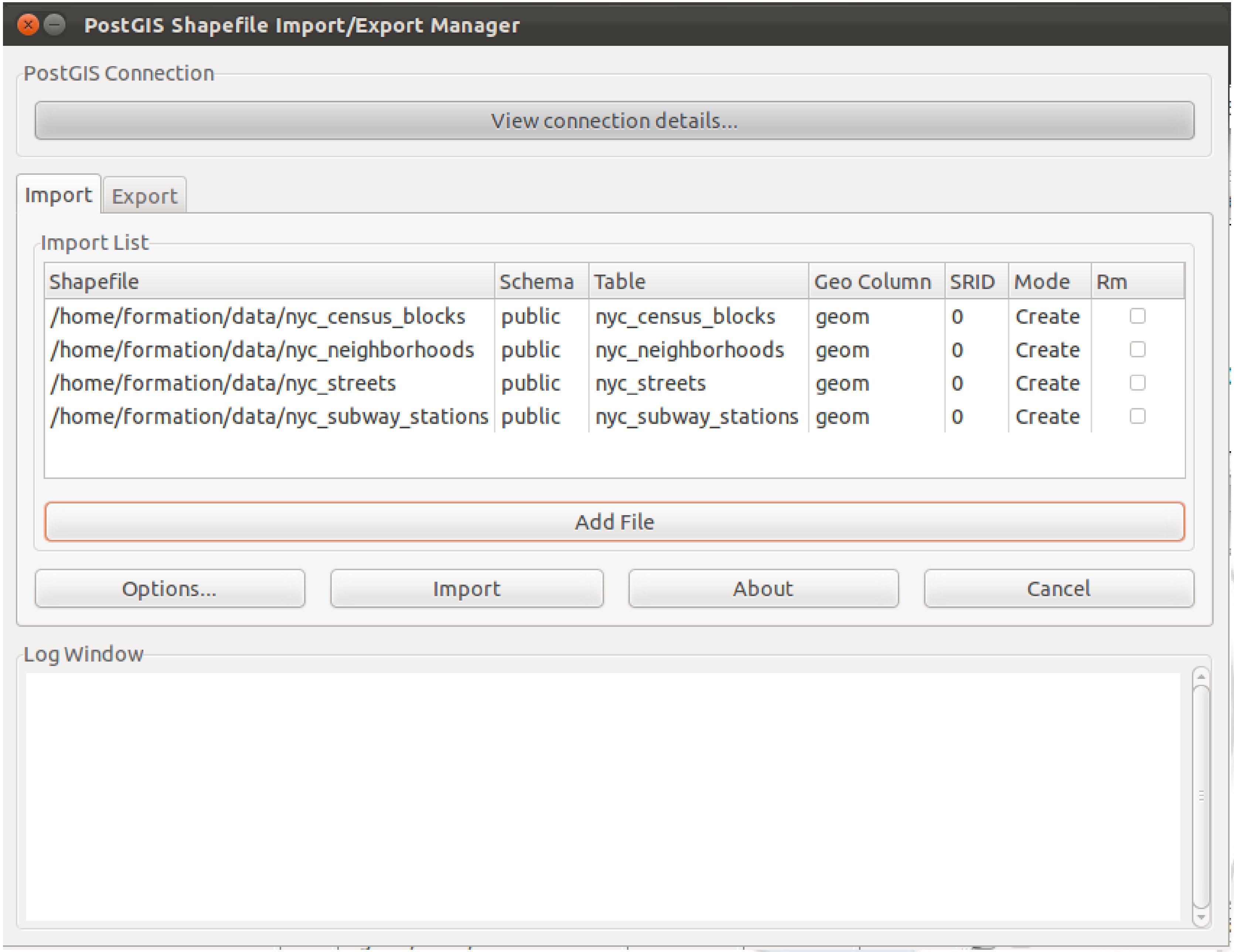
geometry_columns -> vue

Utilisation de Typmod



```
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);
```

Import shapefiles



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	□

Fonctions

ST_ConcaveHull

ST_Snap

ST_Split

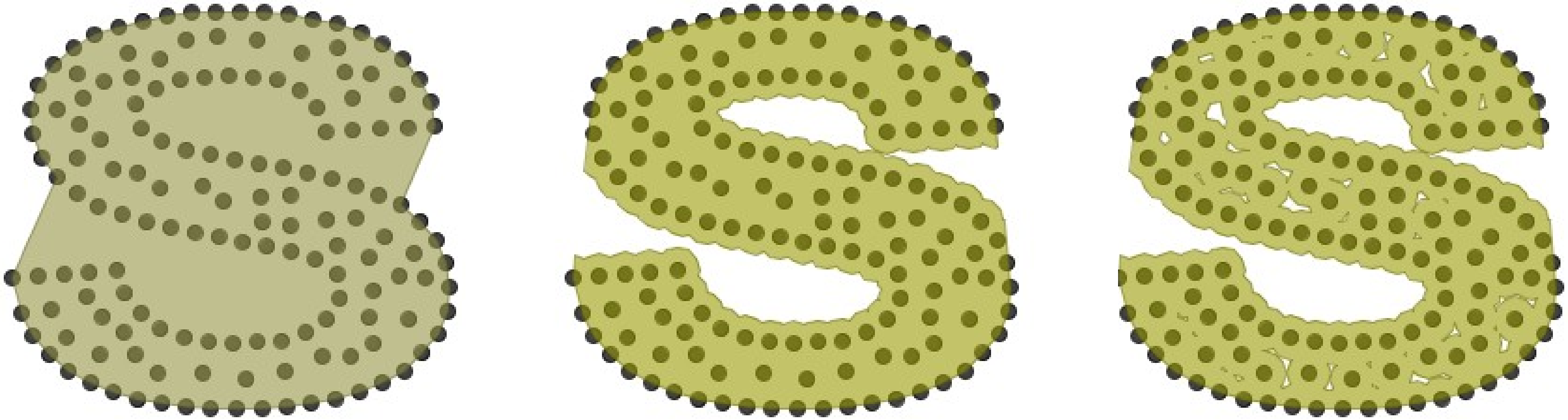
ST_MakeValid

ST_IsValidDetail

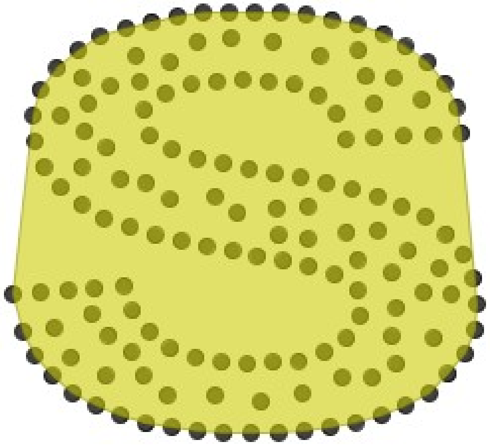
ST_OffsetCurve

...

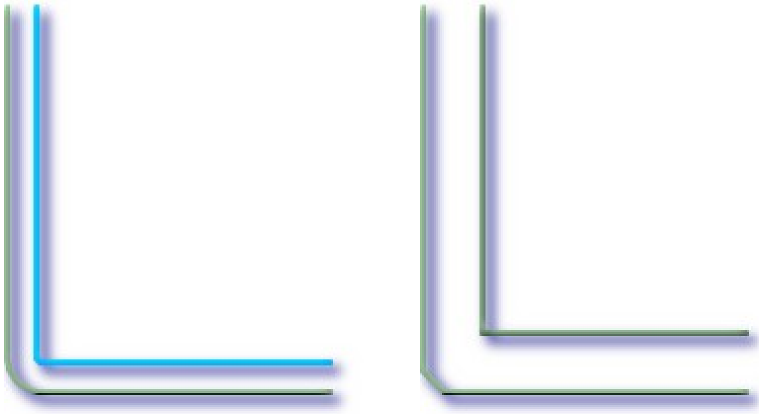
hulls and curves



Concave hulls with various settings



Convex hull



Offset curves

Nettoyage de données

Avant : ST_Buffer(the_geom, 0)

Après :

- 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)
```

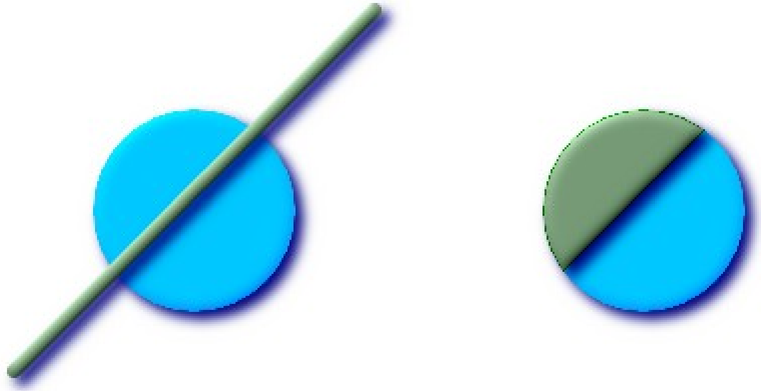
```
SELECT * FROM ST_IsValidDetail('LINESTRING(...)');
        gid  |      reason       |  location
       ------+-------------------+-------------
        5330 | Self-intersection | POINT(32 5)
```

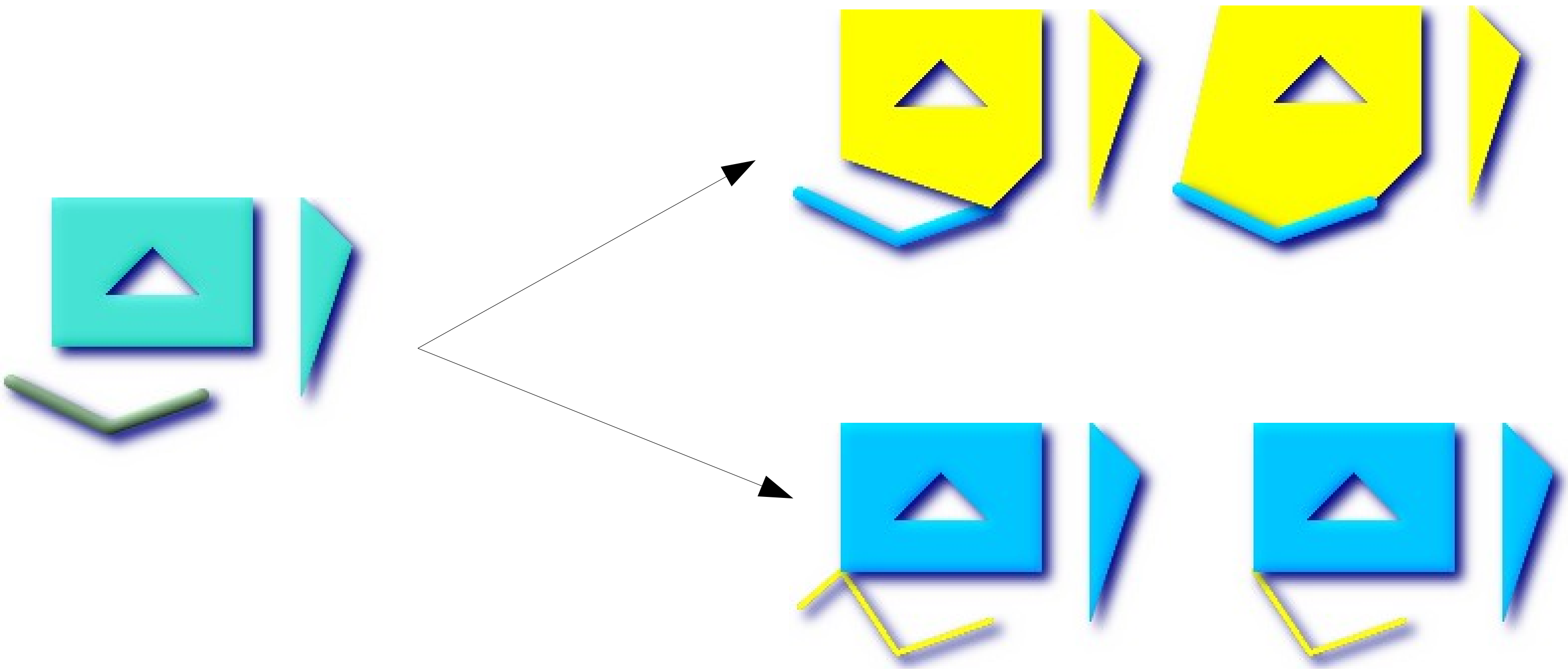
OSLANDIA

Splitting and snapping

ST_Split

ST_Snap





Topologie

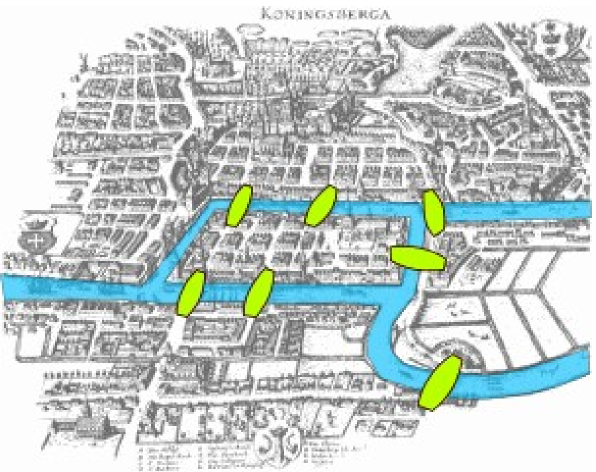
Beware of the spaghetti monster !

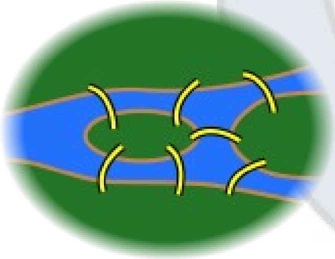
Topologie - Graphes

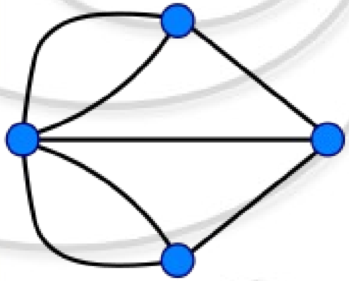
Modélisation explicite des relations

Représentation sous forme de graphe

Modèle OGC : Sommet, Arc, Face

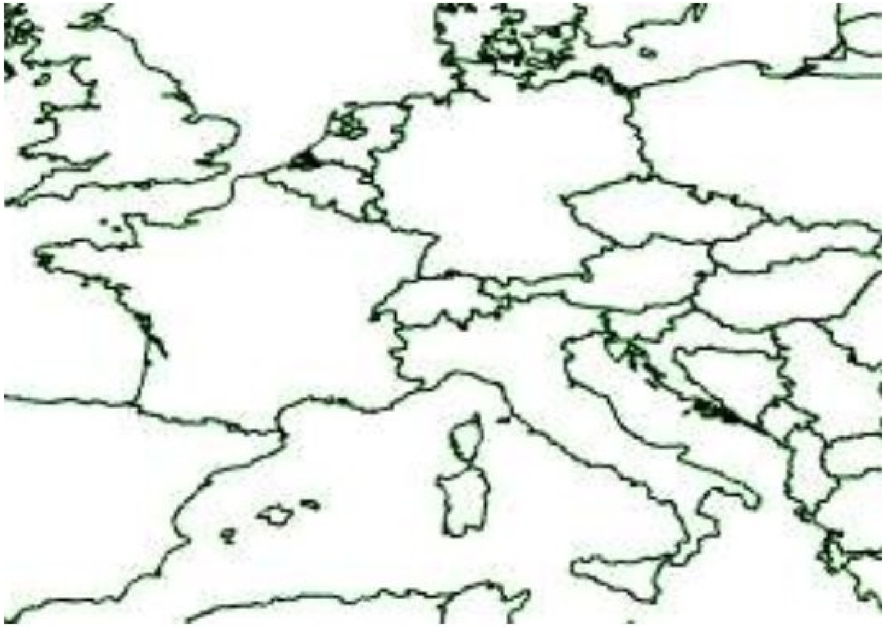


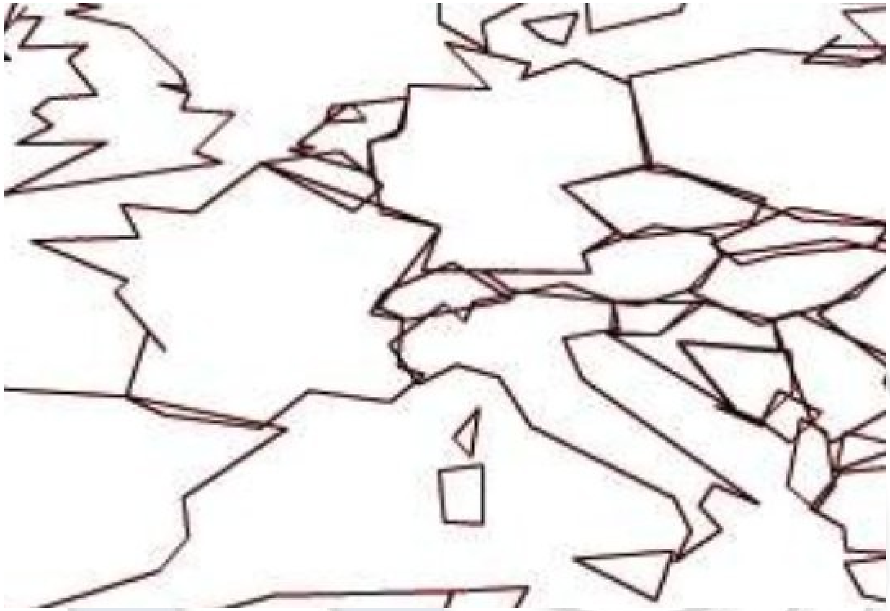




Topologie

Réduction du stockage

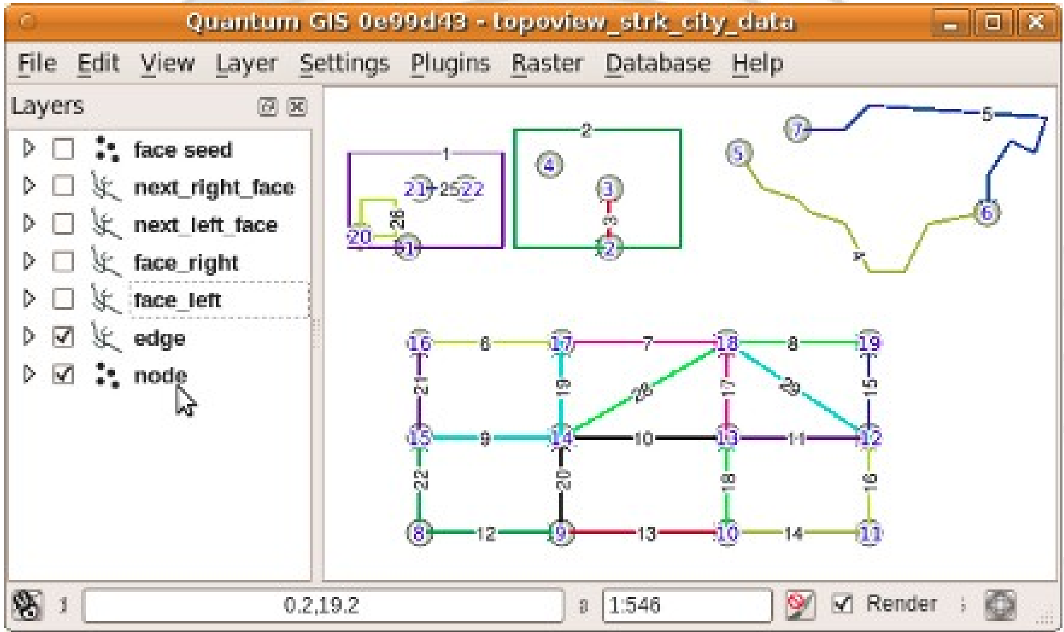




Sans topologie !

Topology

- Nouveau type : TopoGeometry
- Utilisation des schémas
  - «topology» pour les fonctions
  - Chaque topologie a son propre schéma
- Support complet SQL/MM
- Intégré en 2.0
  - Sandro Santilli
  - Région Toscane



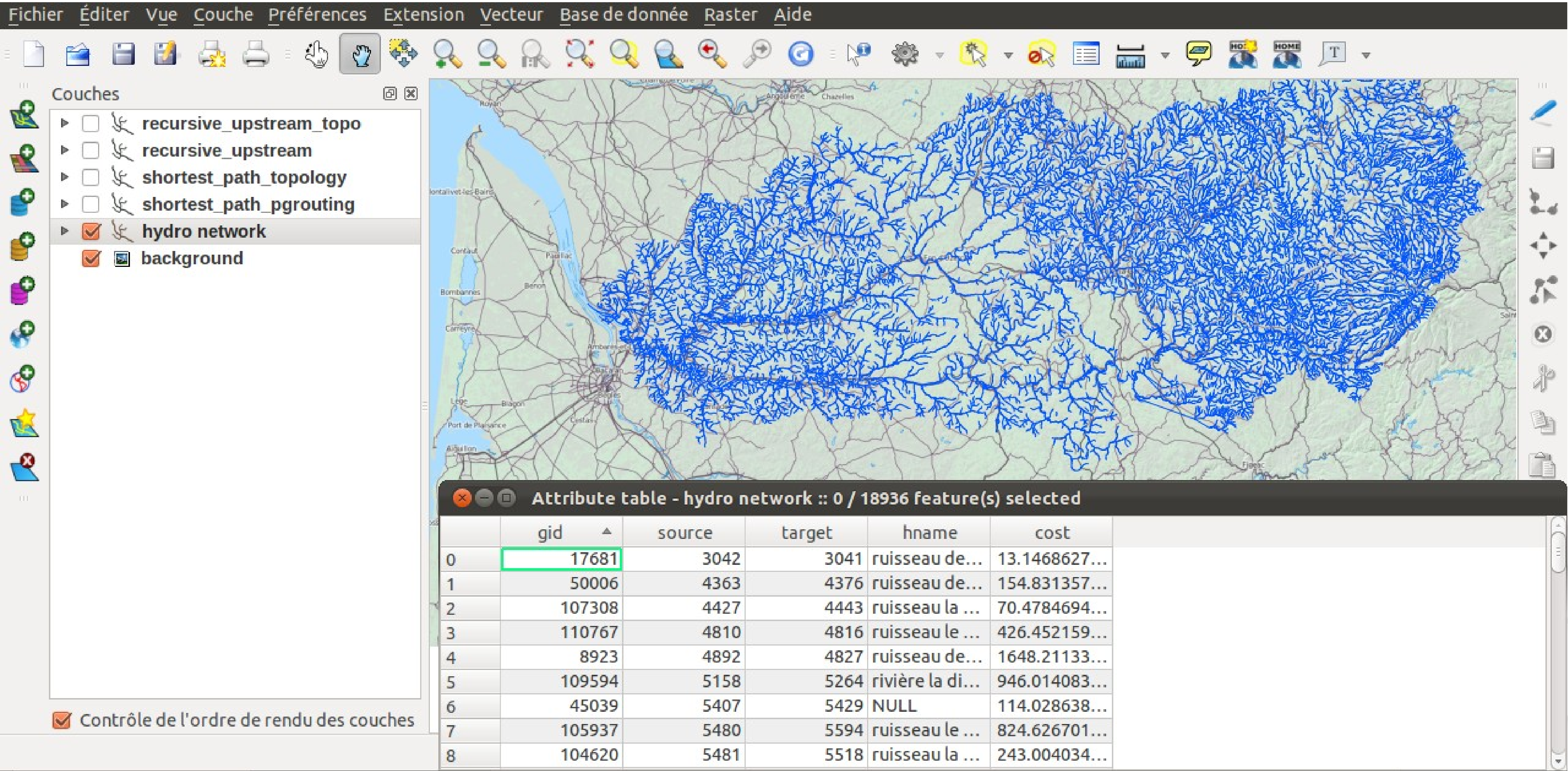
Quantum GIS 0e99d43 - topoview_strk_city_data

- ▷ □ face seed
- ▷ □ next_right_face
- ▷ □ next_left_face
- ▷ □ face_right
- ▷ □ face_left
- ▷ ☑ edge
- ▷ ☑ node

Topologie : exemple

Parcours de cours d'eau

Longueur, débit, etc.



Couches

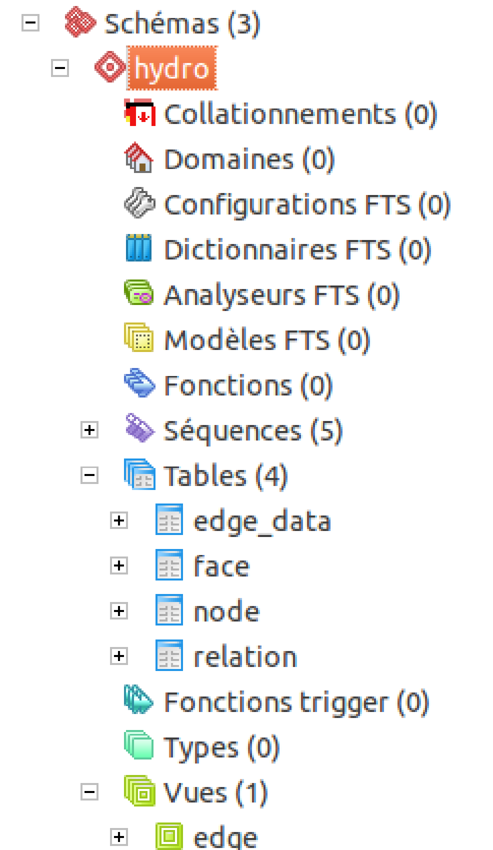
- ☐ recursive_upstream_topo
- ☐ recursive_upstream
- ☐ shortest_path_topology
- ☐ shortest_path_pgrouting
- ☑ hydro network
- ☑ background

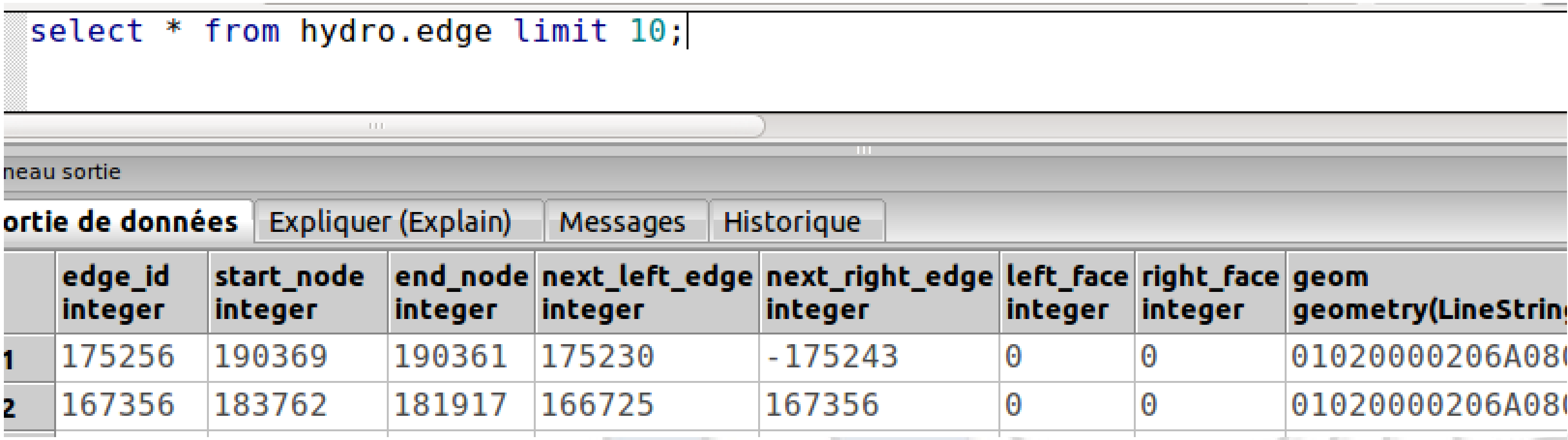
☑ Contrôle de l'ordre de rendu des couches

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...

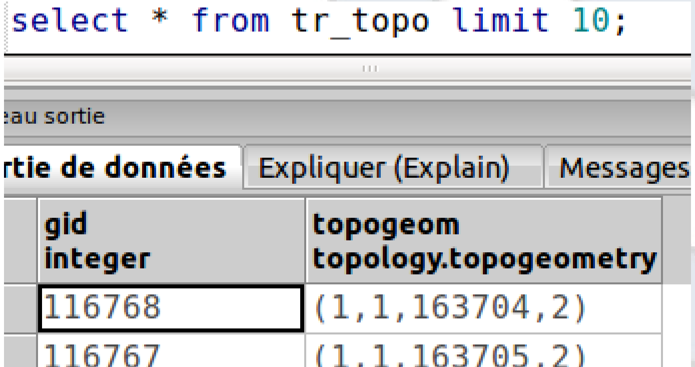
```
-- Create a topology
SELECT topology.CreateTopology('hydro', 2154);
-- 1

-- we put the postgis topology features for hydro network in another table
CREATE TABLE tr_topo (gid integer);

-- Add a layer
SELECT topology.AddTopoGeometryColumn('hydro', 'public',
        'tr_topo', 'topogeom', 'MULTILINESTRING');
-- 1

-- Populate the layer and the topology from tr geometry features
INSERT into tr_topo (gid, topogeom)
        SELECT gid, topology.toTopoGeom(geom, 'hydro', 1) FROM tr;
```





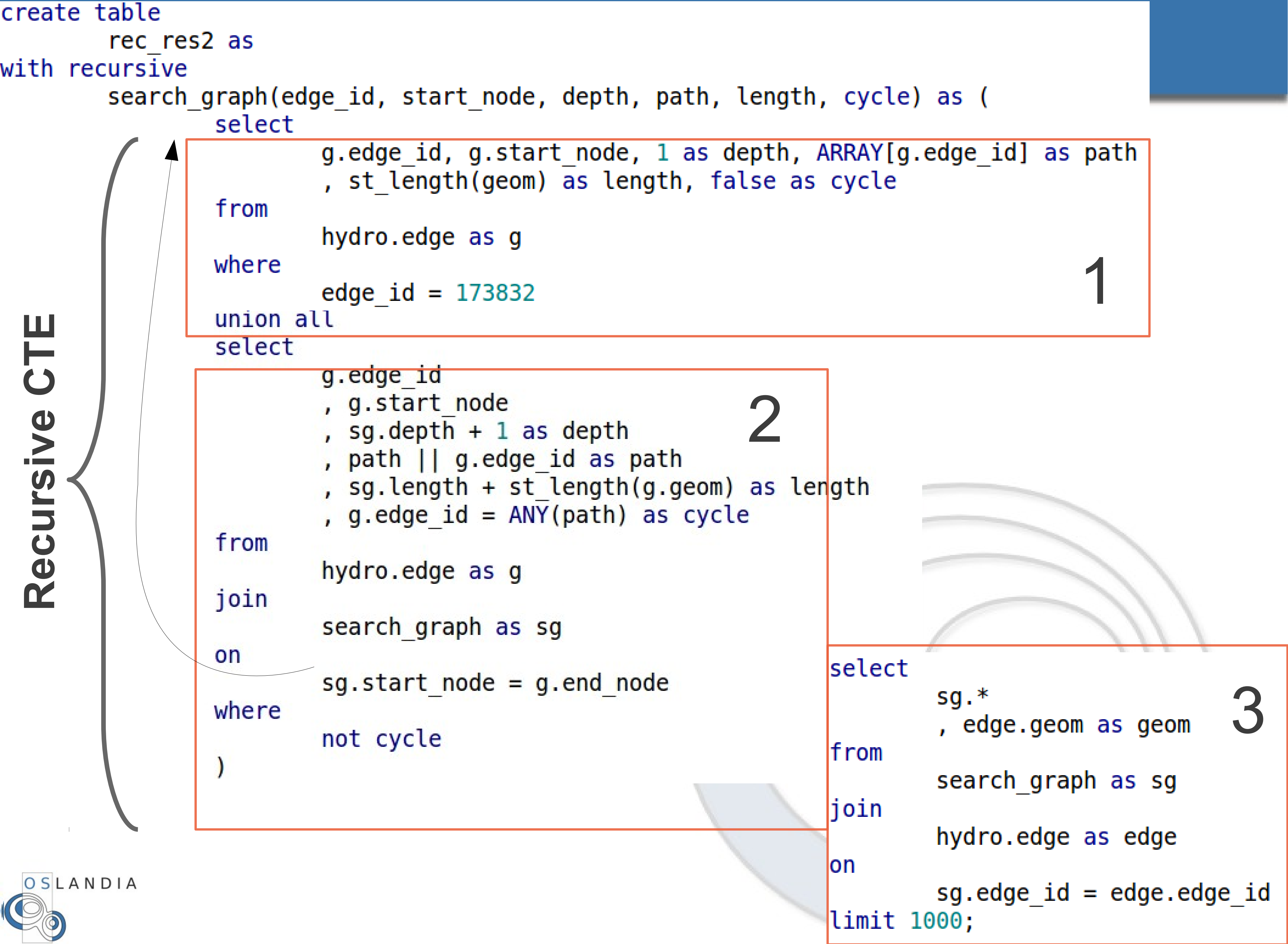
```
select * from hydro.edge limit 10;
```

	edge_id integer	start_node integer	end_node integer	next_left_edge integer	next_right_edge integer	left_face integer	right_face integer	geom geometry(LineStrin
1	175256	190369	190361	175230	-175243	0	0	01020000206A08
2	167356	183762	181917	166725	167356	0	0	01020000206A08



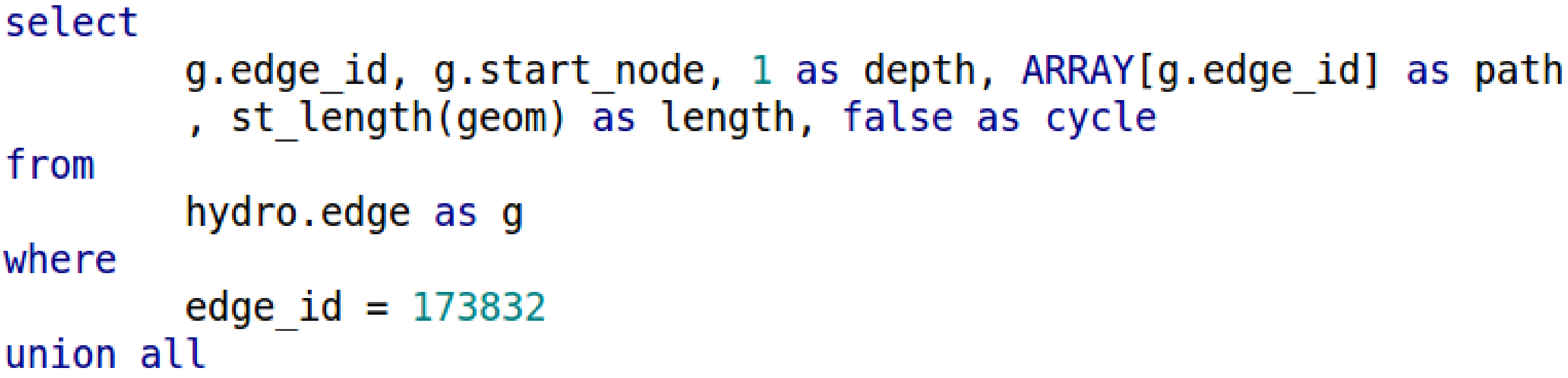
```
select * from tr_topo limit 10;
```

gid integer	topogeom topology.topogeometry
116768	(1,1,163704,2)
116767	(1,1,163705,2)



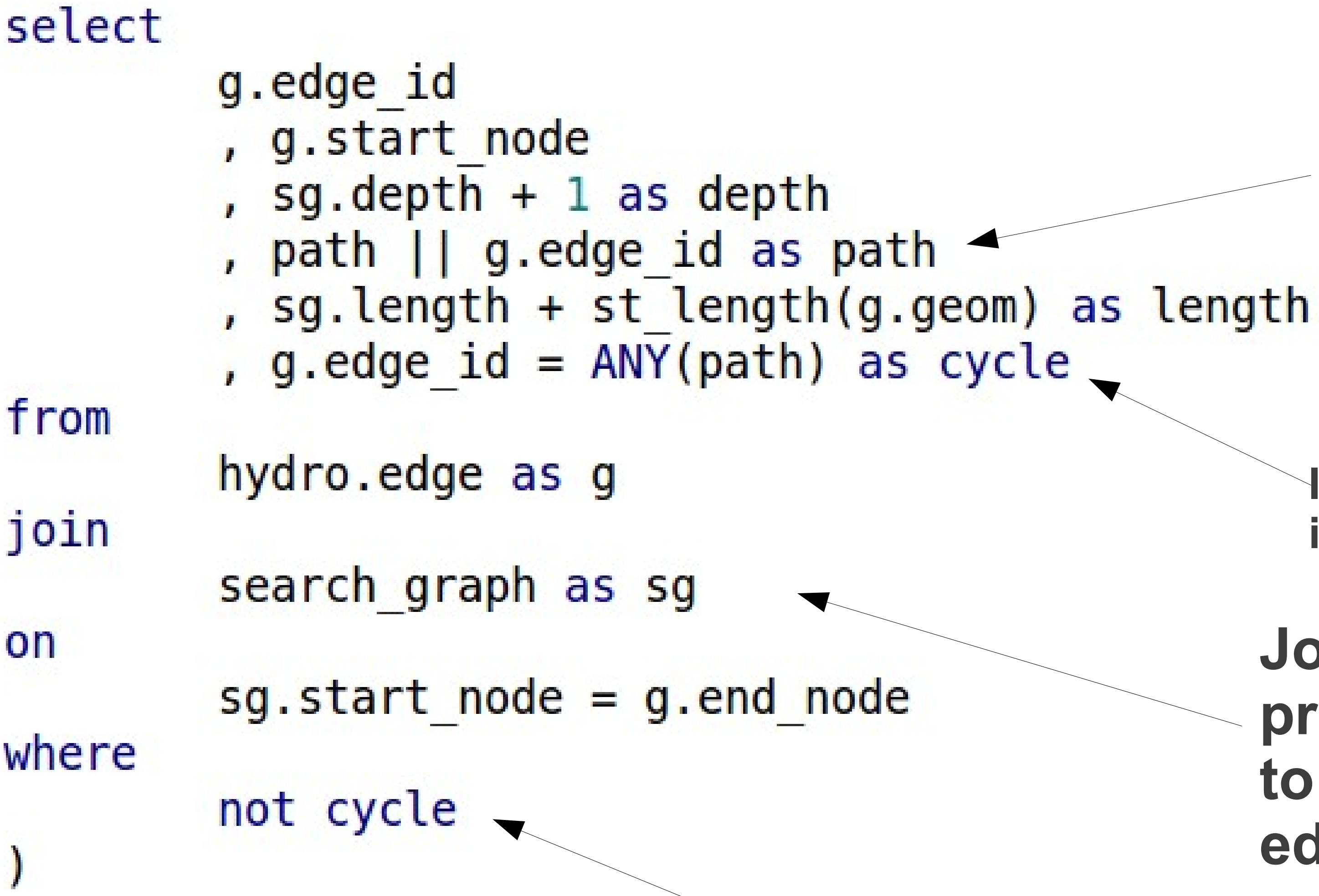
```
create table
        rec_res2 as
with recursive
        search_graph(edge_id, start_node, depth, path, length, cycle) as (
                select
                        g.edge_id, g.start_node, 1 as depth, ARRAY[g.edge_id] as path
                        , st_length(geom) as length, false as cycle
                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
                )
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;
```

1 : init



```
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
```

2 : recursive part



```
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
)
```

Stack the gid to the path
for this record

Sum up the cost
(it's the length here)

If the record gid is already
in the path, we have a cycle

Join result set from previous iteration to connected upstream edges

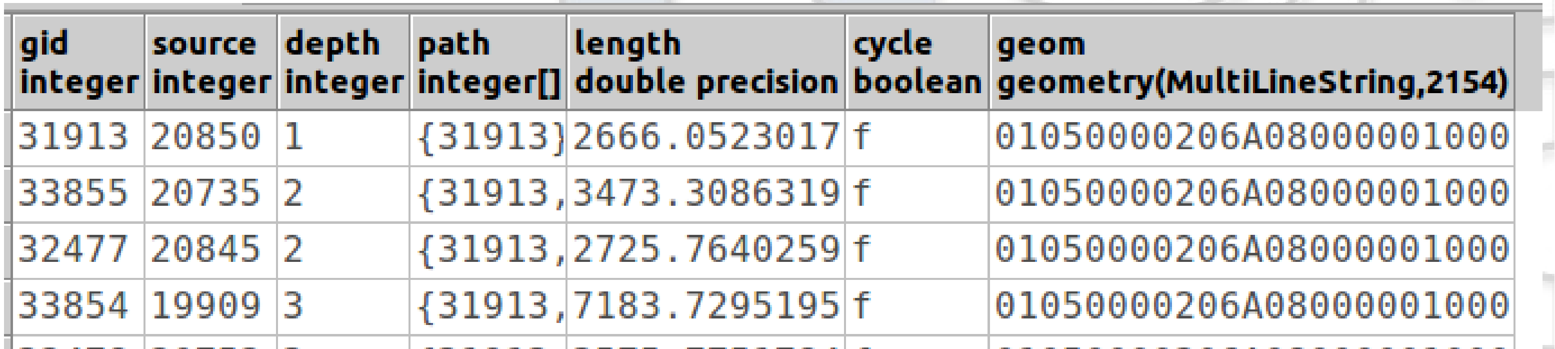
Do not take elements
which make a cycle

3 : Get results

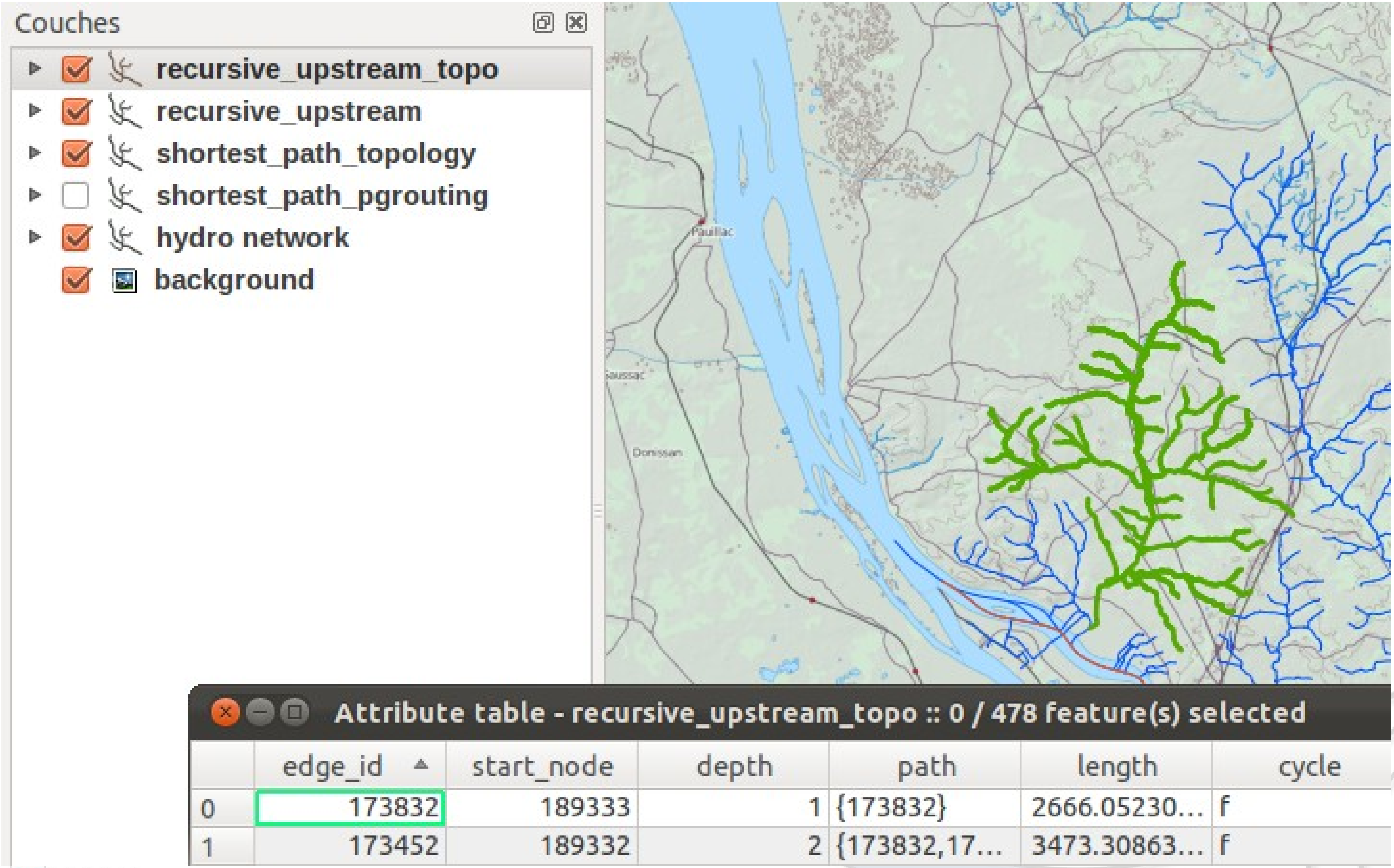
```
select
        sg.*
        , edge.geom as geom
from
        search_graph as sg
join
        hydro.edge as edge
on
        sg.edge_id = edge.edge_id
limit 1000;
```

Join CTE results to original table to get geometries

Better limit recursive queries to avoid unfinite loops



gid integer	source integer	depth integer	path integer[]	length double precision	cycle boolean	geom geometry(MultiLineString,2154)
31913	20850	1	{31913}	2666.0523017	f	01050000206A08000001000
33855	20735	2	{31913,	3473.3086319	f	01050000206A08000001000
32477	20845	2	{31913,	2725.7640259	f	01050000206A08000001000
33854	19909	3	{31913,	7183.7295195	f	01050000206A08000001000



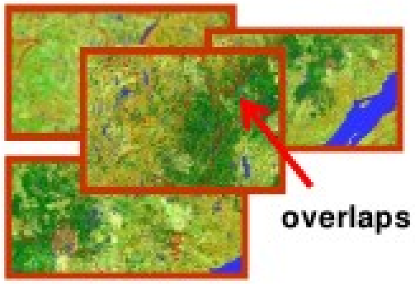
Attribute table - recursive_upstream_topo :: 0 / 478 feature(s) selected

	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

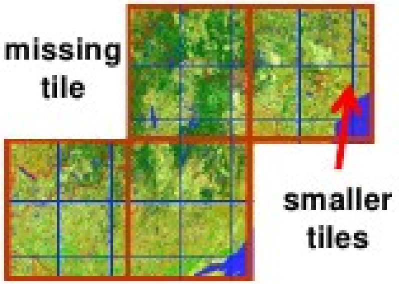
PostGIS Raster

- Analyse Raster / Vector
- Nouveau type
  - Comme des géométries
- Multiresolution, multiband, tuilage
- Import/export (GDAL)
- Fonctions
  - Statistiques, reprojection, édition, calcul
  - Liens avec vecteurs
  - Chaîne de traitements (ST_MapAlgebra)

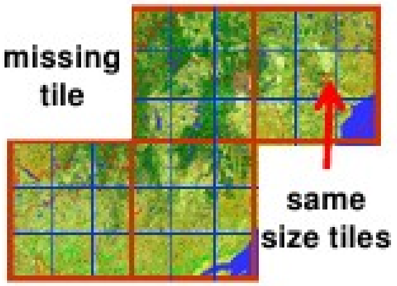
PostGIS Raster



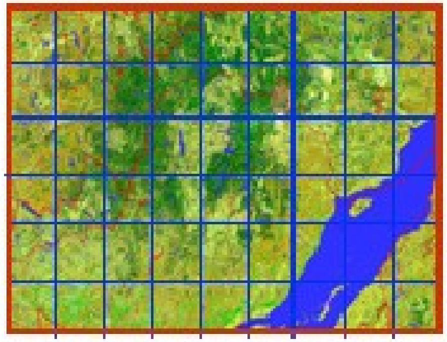
a) warehouse of untiled and unrelated images (4 images)



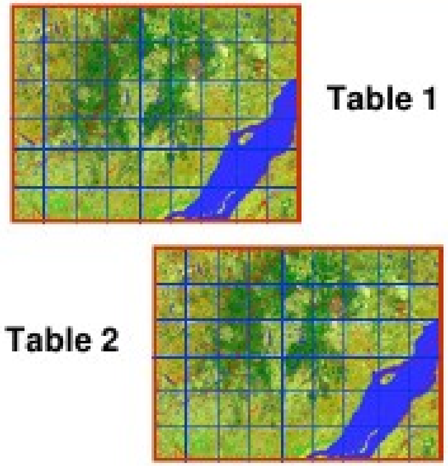
b) irregularly tiled raster coverage (36 tiles)



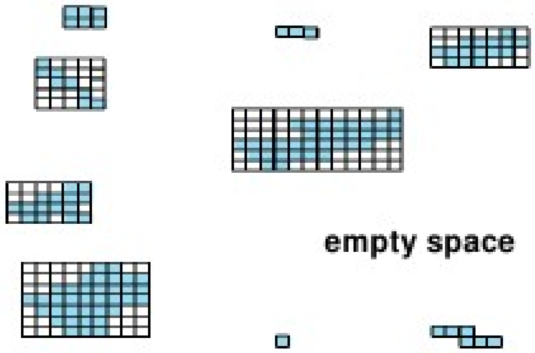
c) regularly tiled raster coverage (36 tiles)



d) rectangular regularly tiled raster coverage (54 tiles)

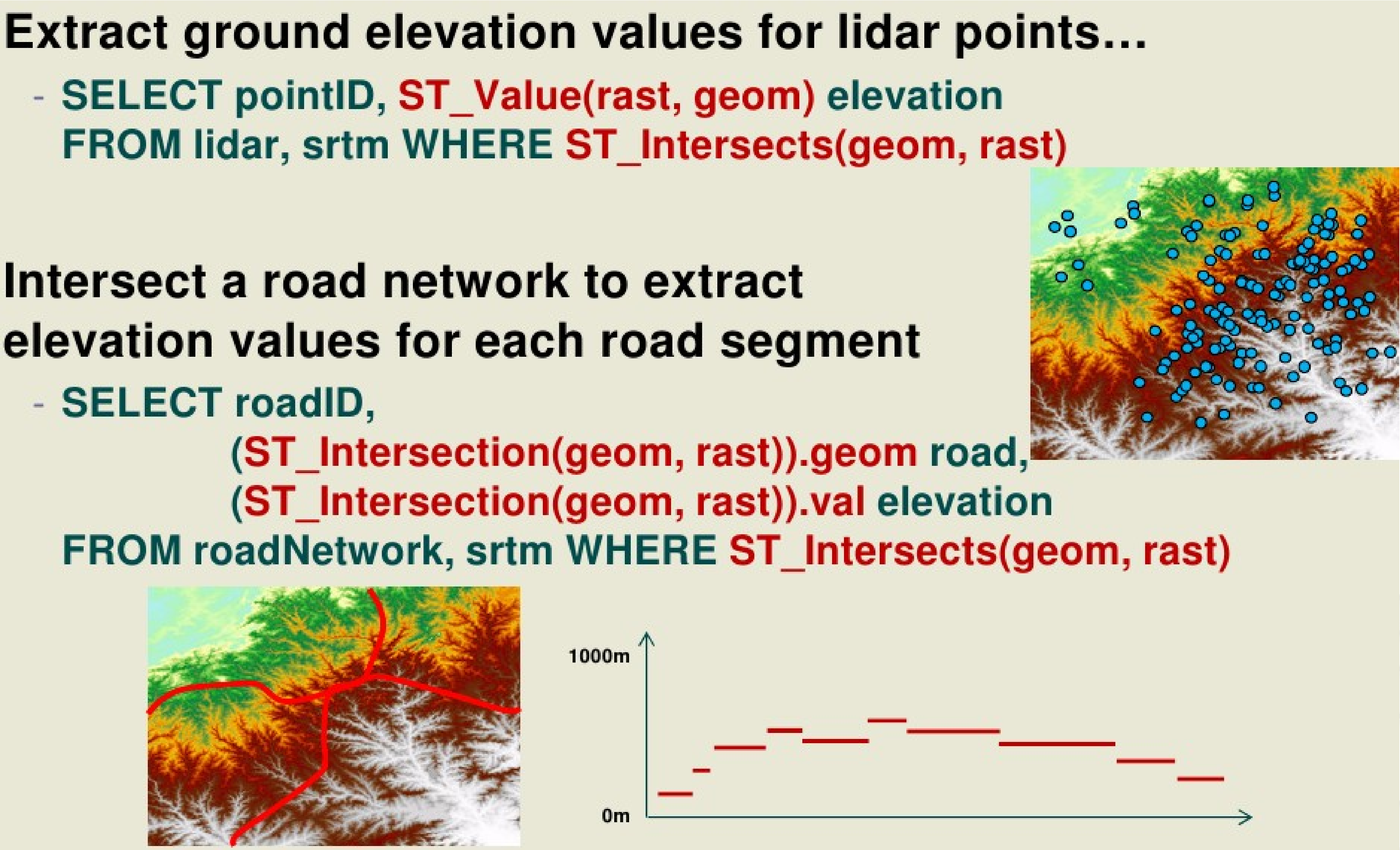


e) tiled images (2 tables of 54 tiles)



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 : plus proches voisins

KNN-GIST dans PostgreSQL 9.1

Utilise des index !

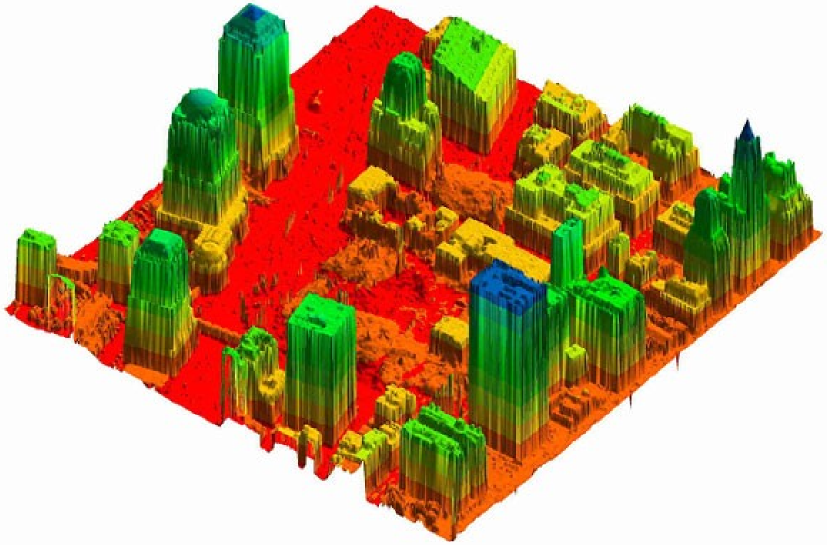
Plus proches voisins

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

Opérateur

- <-> ou <#> : centre or bbox (indexé)
- Besoin d'affiner après bbox

Nuage de points



- Support de données LIDAR
- Comme extension
  - Et extension PostGIS
- Points et « patches »
- Nombre arbitraire de dimensions
  - « Schéma » de point cloud
- Compression de données (reshape, RLE)
- Lien avec lib PDAL

Nuage de points

Analyse spatiale

```
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

En chantier :

Distance Arc-geometry

Distance with cached tree

R-Tree index improvement (pick-split)

SP-Gist Index :

- Nouveau avec PG 9.1
- Bien plus rapide à construire
- Et à lire

ST_MapAlgebra avec N rasters

PostGIS 2.1

Améliorations sur la topologie

Tiger geocoder comme extension PG

PgRouting comme extension

+ et support Windows

Et … au delà ?

PostGIS 3.0 ?

Paris codesprint et barcamp Mai 2012

Boston codesprint Mars 2013

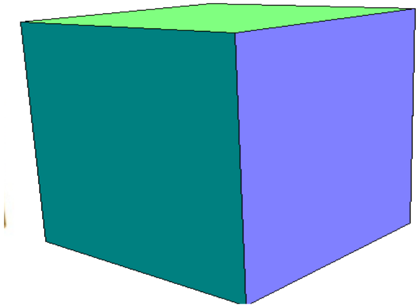
Directions futures :

- Git, build system (pour Windows)
- Geometry backend (GEOS vs BGL vs ?)
- Améliorations Raster
- 3D topology & processing (CGAL?)
- Nuages de points
- Performance, performance, performance

PostGIS x.y : prochaine dimension

Let's go 3D !

Stockage 3D



- 3D « réelle » dans PostGIS
- Standards ISO et OGC
  - ISO 19125, SQL/MM, SFS 1.2.0
- Premières implémentations
- Nouvelles données et fonctions

3D

Nouveaux types

- TRIANGLE, POLYHEDRALSURFACE, TIN

Input/Output

- ST_AsGML, ST_AsX3D...

Nouveaux opérateurs

- &&&

Spatial index : nd-indexes

PostGIS 3D

Des traitements 3D !

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

PostGIS 3D

NKOT$_{\text{(geo)}}$B : CGAL

Calculs géométriques 2D & 3D

C++

Calculs **exacts**

Générique, performant, extensible ...

PostGIS 3D

...et maintenant GPL !

PostGIS 3D

e-PLU

Financement partiel FEDER

Coopération IGN et urbanistes

Naissance de **SFCGAL** (OGC-SF)

Inclusion dans PostGIS ?

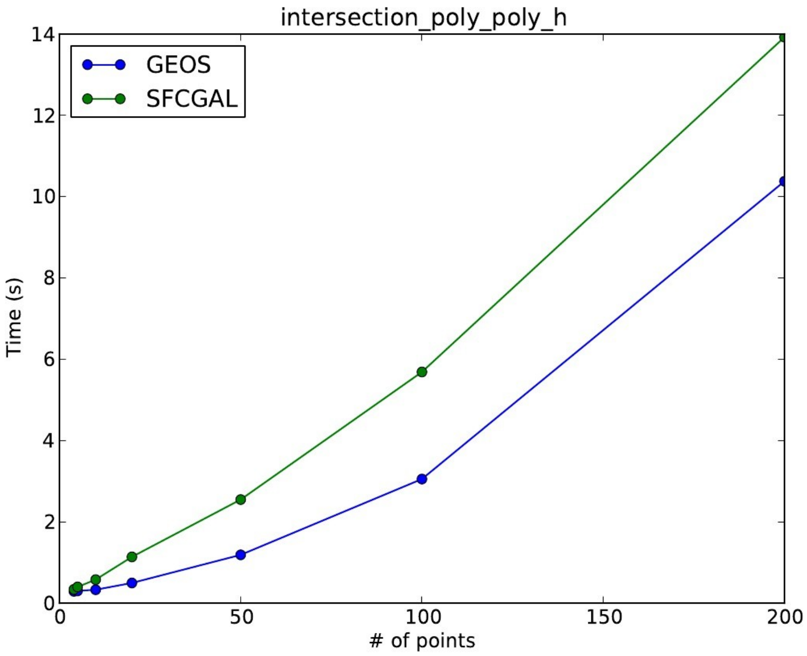
Comparaison avec GEOS (2D) ?

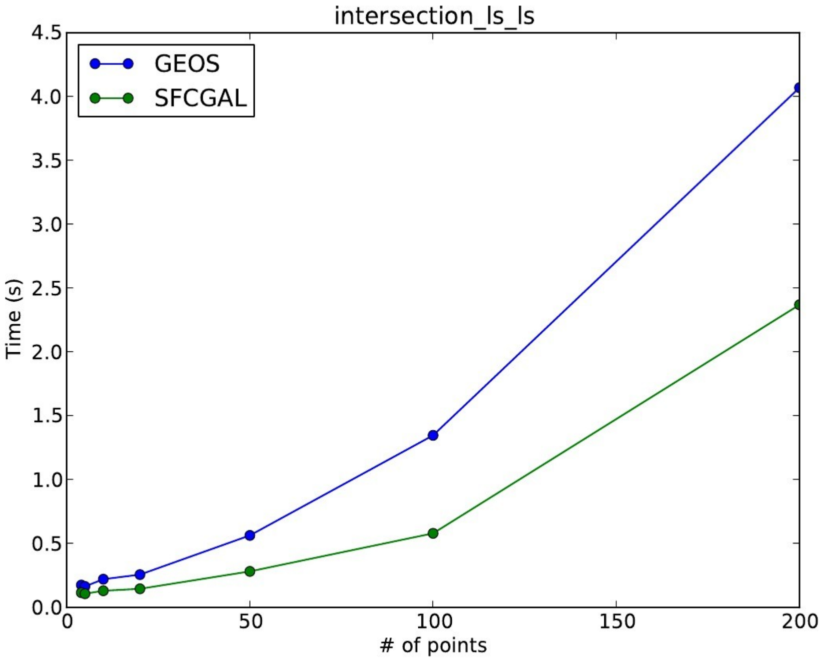
GABRIEL PERI
Volumétries
et contraintes de prospect*
3800m² SHON
(47 logts)

PostGIS 3D

Comparaison de performance avec GEOS

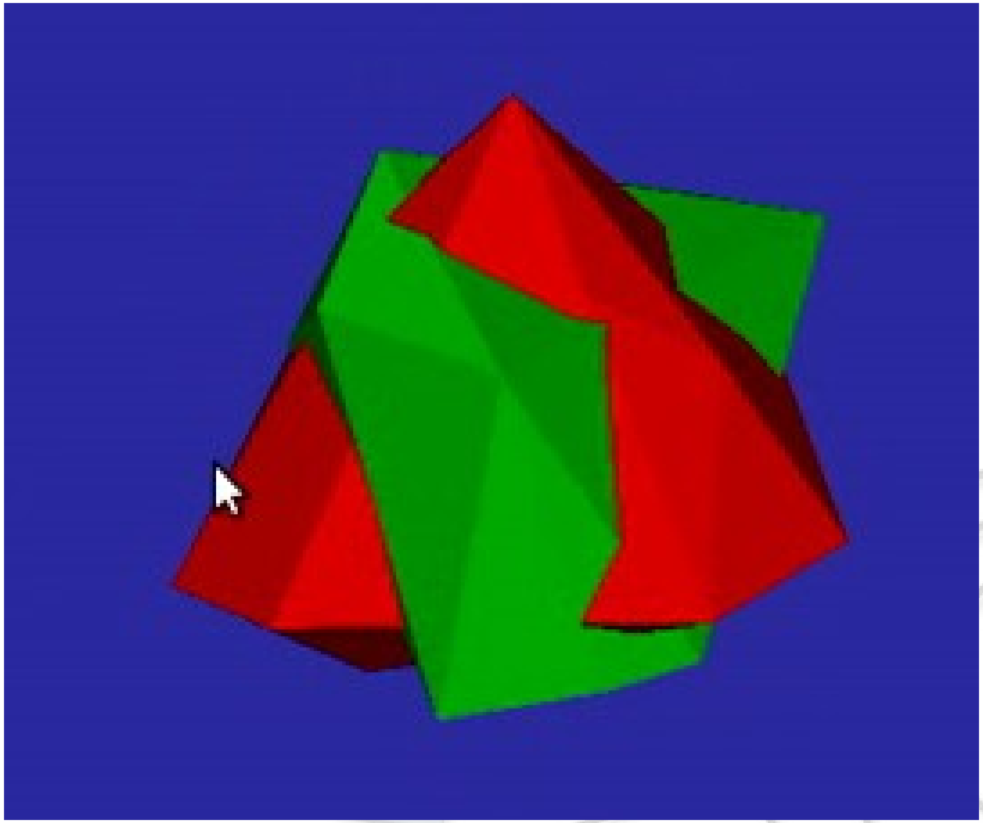
-> Comparable !





Récemment

- Client Quantum GIS (Globe)
- Quelques fonctions
  - ST_Extrude
  - ST_3DConvexhull
  - ST_3DIntersection
    - Surfaces
    - Solides
- **Dans PostGIS 2.1 !**



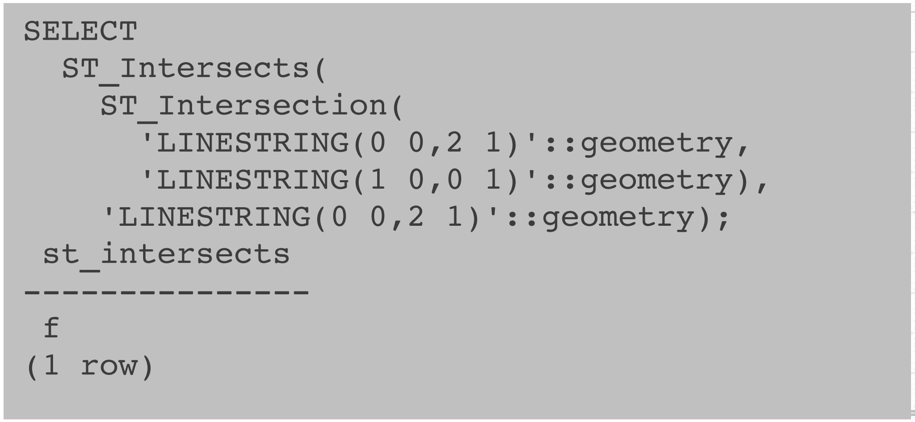
Une petite vidéo ?

PostGIS 3D : ensuite ?

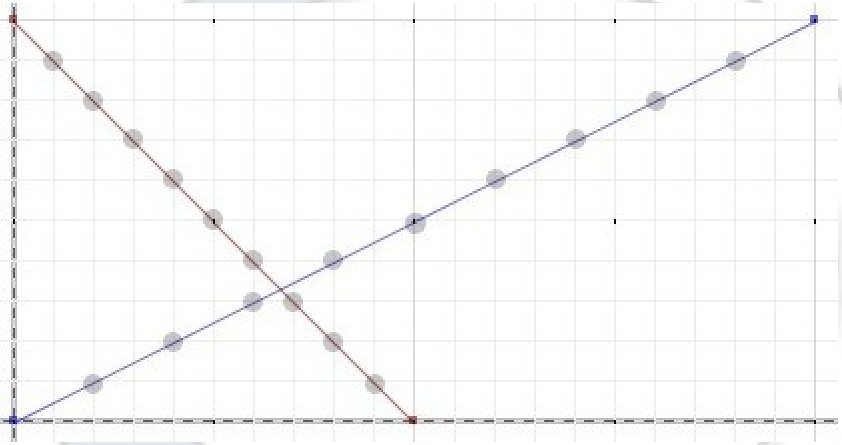
CGAL : calculs exacts

Nouveaux objets : géométries exactes

- Tenter d'éviter la sérialization
- (patch PostgreSQL)



```
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)
```



3D Next steps

- Plus de fonctions CGAL
  - Alpha shapes
  - 3D Minkowski sum
  - 3D snap rounding ?
  - ...
- Meilleur intégration QGIS
- Loaders / exporters CityGML & Collada
- Textures ?
- Trouver des €€€€€ pour accélérer le dev.

That's it...

Questions ?

hugo.mercier@oslandia.com

http://www.oslandia.com

http://ww.github.com/Oslandia

OSLANDIA