



PgDay – PostGIS ou comment stocker et manipuler l'information géographique

Nicolas Ribot - Olivier Courtin - Licence GNU FDL

Qui sommes-nous

- **Société** Camptocamp: spécialiste des solutions géomatiques Opensource
- **Orateurs/rédacteurs**
 - Olivier Courtin - olivier.courtin@camptocamp.com
 - Nicolas Ribot - nicolas.ribo@camptocamp.com

Contexte

- Utilisation de PostGIS comme SGBD spatial depuis plusieurs années
- Participation à la communauté PostGIS
- Position dominante de PostGIS dans le monde des SGBD Spatiaux libres

Plan

- Historique de PostGIS
- Rappel sur les normes et standards: OGC SFSQL, ISO SQL/MM
- Principaux SGBD spatiaux
- Objets géographiques supportés
 - Hiérarchie d'objets géographique
 - Définition mathématiques
 - Représentation interne, binaire et textuelles des objets
- Index spatiaux
 - Intérêt des index
 - Mode de fonctionnement
 - Mise en oeuvre, utilisation dans des requêtes
 - Limitations
- Fonctions et opérateurs spatiaux
 - Présentation de GEOS, bibliothèques géographique
 - Différence entre fonctions et opérateurs
 - Principales fonctions spatiales
 - Utilisation des fonctions et opérateurs pour des traitements complexes
 - Agrégation, fusion
 - Analyses spatiales

Plan (suite)

- Formats d'export géographiques
 - GML
 - KML
 - SVG
 - GeoJson
 - Utilitaire d'import/export de shapefiles: shp2pgsql, pgsql2shp
- Outil SIG OpenSource
 - Bureautique
- Serveur d'application
 - MapServer
 - GeoServer
- Futur de PostGIS
 - Roadmap court terme
 - Roadmap long terme
- Conclusion



Historique de PostGIS

SGBD spatial, concepts et enjeux

- Stockage de l'information géographique et de ses attributs
- Récupération de tout ou partie de l'information sur requêtes géographiques ou non
- Gestion d'index optimisés pour les requêtes spatiales
- Respect des standards existants (OGC SFS ou ISO SQL/MM)
- Permet de faire tourner de très gros volumes de données (plusieurs To)

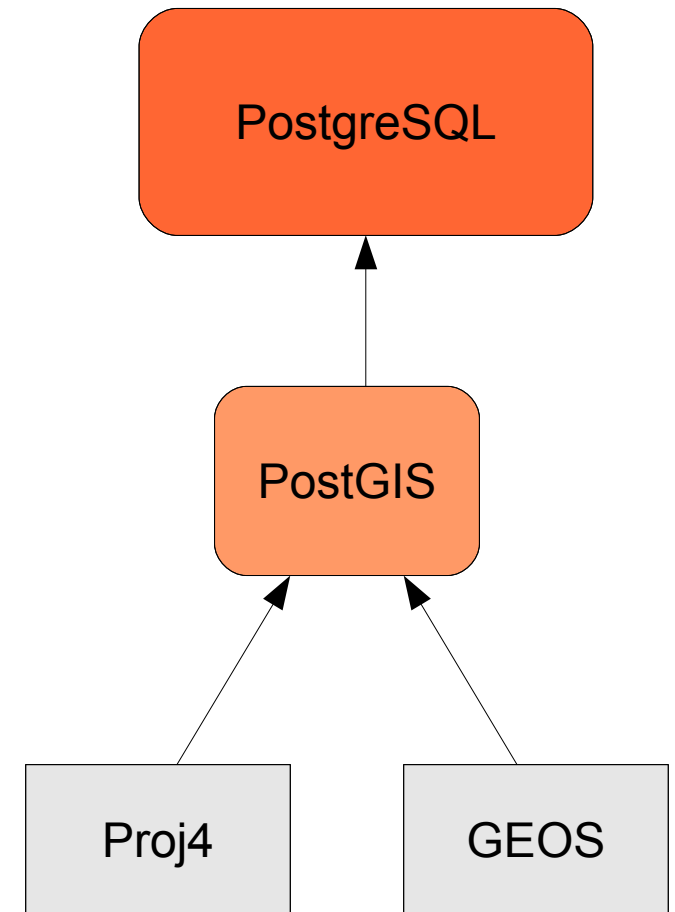
Présentation de PostGIS

- Site Web officiel : <http://postgis.refractions.net/>
- Site français (non officiel): <http://postgis.fr/>
- Version actuelle 1.3.4
- Cycle de sortie de nouvelles versions rapide
- Développé en C
 - Sous forme de plugin pour PostgreSQL
 - Lisibilité et concision remarquable du code
- Société éditrice Canadienne : Refractions
- Utilisé dans de très nombreux projets
 - Références prestigieuses (IGN, SwissTopo, IRSN, etc.)
 - Communauté large et technique
- Respect du standard OGC SFS (Simple Feature for SQL).
- Nombreuses fonctionnalités additionnelles à SFS



PostGIS Architecture et librairies

- PostGIS comme plugin PostgreSQL
- Couplable (optionnel) à Proj4
 - pour gestion de très nombreux systèmes de projections
- Couplable (optionnel) à GEOS
 - pour gestion d'opérateurs spatiaux



Historique de PostGIS

- Version publique initiale (2001)
- Version 0.8 (2004)
 - Version déjà utilisable en production, nombreuses fonctionnalités avancés également présentes
- Version 1.0 (2005)
 - Validation OGC SFS 1.0
 - Gestion de data en 4D
 - Réécriture de toute la sous couche de sérialisation des géométries en base (passage de WKT à HEWKB pour le stockage natif)
- Version 1.1.x et 1.2.x (2006 et 2007)
 - Nombreuses fonctionnalités additionnelles
 - Débug et optimisations diverses
 - Connecteur JDBC, ARCSDE
- Version 1.3.x (2007)
 - Préfixe ST_ pour compatibilité SQL/MM

Support et documentation PostGIS

- Documentation
 - <http://postgis.refractions.net/doc/>
 - <http://www.postgis.fr/book>
- Mailing list officielles:
 - postgis-users
 - postgis-devel
- Autre sites francophones
 - postgis.fr
 - Georezo
 - forumsig.org
- Support commercial
 - Camptocamp
 - Refractions
 - ...





Standards internationaux OGC et ISO

Le concept de 'feature'

- Une 'feature' comme abstraction géographique de la réalité,
 - Géopositionnée
 - Peut être doté d'attributs

Le concept d'API spatiale standardisé

- Permet de définir pour un SGBD existant (SQL92 ou SQL99)
 - La liste des types géométriques possibles
 - La manière de représenter les données spatiales
 - WKT
 - WKB
 - Les méthodes spatiales disponibles
 - Prototype
 - Comportement
 - Les contraintes d'intégrité spatiales
 - Typage géométrique
 - Système de projection

Standards existants: OGC SFS 1.1

- Disponible depuis 99
- <http://www.opengeospatial.org/standards/sfs>
- Logique 2D (X Y)
- Types géométrique définis
 - POINT
 - POLYGON
 - LINESTRING
 - (+ Logique de MULTI et d'agrégat)
- Environ 80 fonctions d'accès et de manipulation des données spatiales
- Schéma d'intégrité référentielle spatiale
 - geometry_columns
 - spatial_ref_sys

Standards existants: ISO SQL/MM

- Standard ISO
- Logique de stockage X Y [Z] [M]
- Types spatial additionel:
 - Curve
- Logique additionnelle
 - Network et routing
 - Représentation topologique
 - Fonctions géocentriques
- Nombreuses fonctionnalités spatiales additionnelles à OGC SFS

Principaux SGBD spatiaux

- PostGIS
- Oracle Spatial
- Arc SDE
- Oracle Locator
- MySQL Spatial

LINAGORA

CNES / DSI / EA / AI

Centre National d'études Spatiales



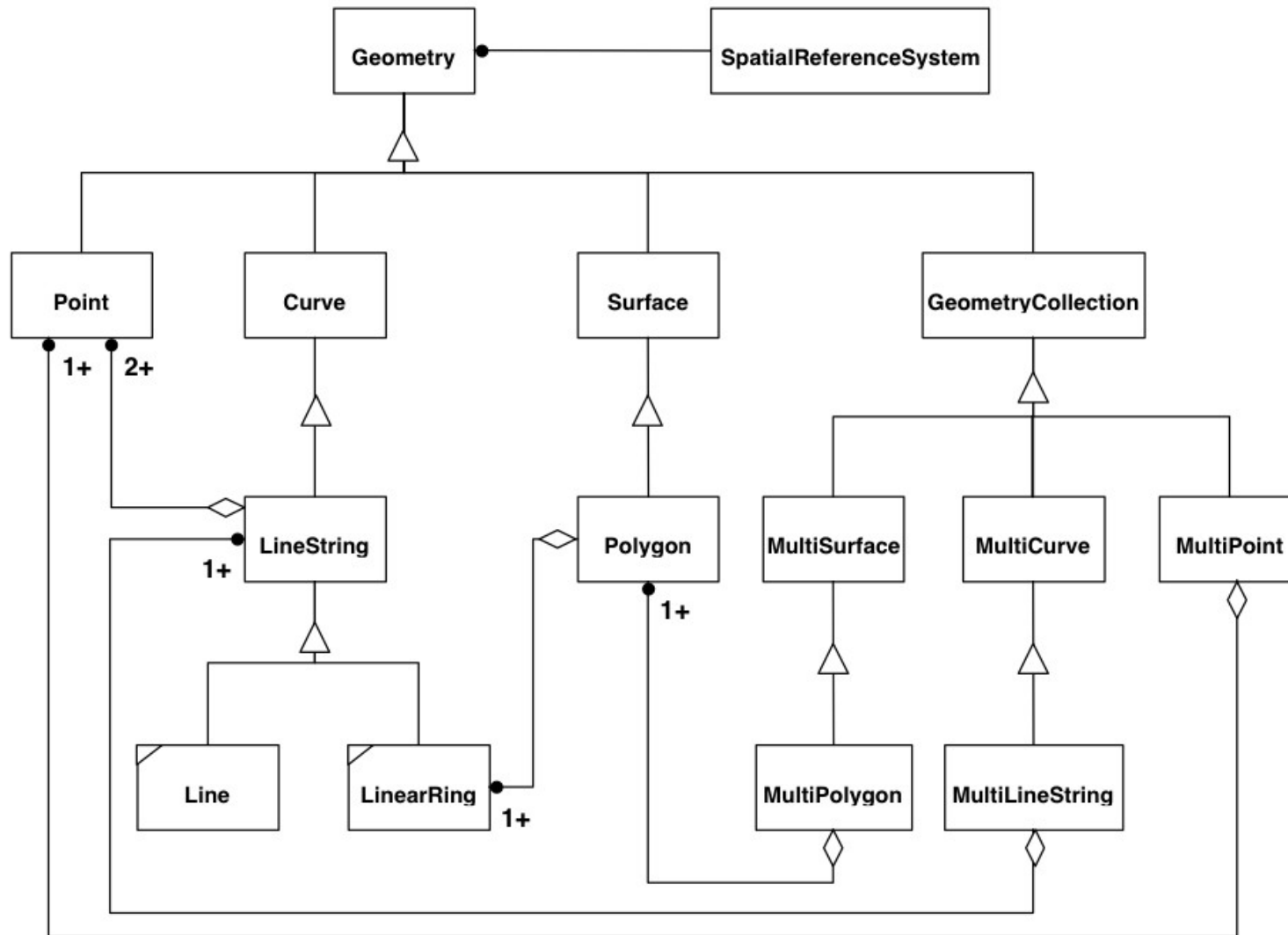
Etude des bases de données spatialisées

- Etude mandaté par le CNES sur un comparatif fonctionnel entre les principaux SGBD spatiaux:
 - cct.cnes.fr/cct05/public/2007/documents/Etude_comp_bases_donnees_spatialisees/rapport_etude_spatiale_final.pdf



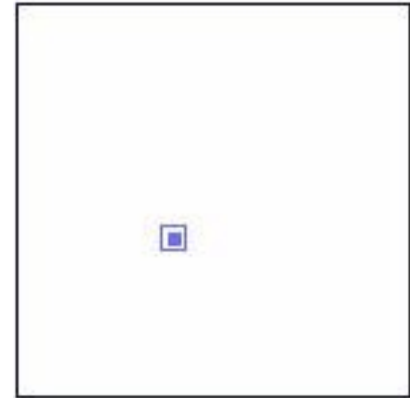
Objets géographiques supportés

Modèle Objet PostGIS



Représentation WKT

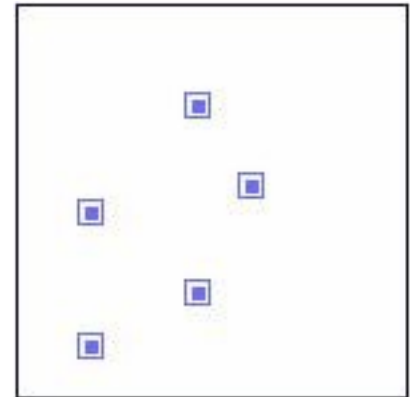
POINT (10 10)



WKT: MULTIPOINT

MULTIPOINT

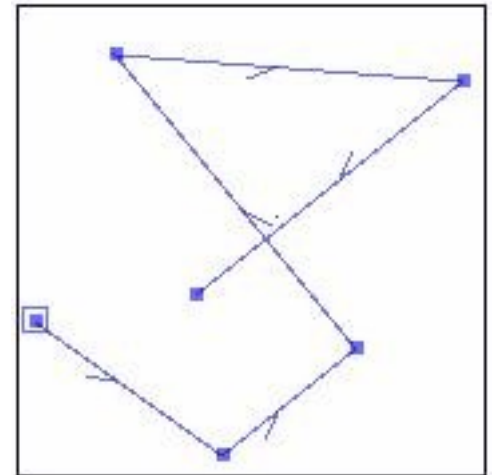
```
(  
  5 5, 5 15, 10 7, 10 30, 14 17  
)
```



WKT: LINESTRING

LINESTRING

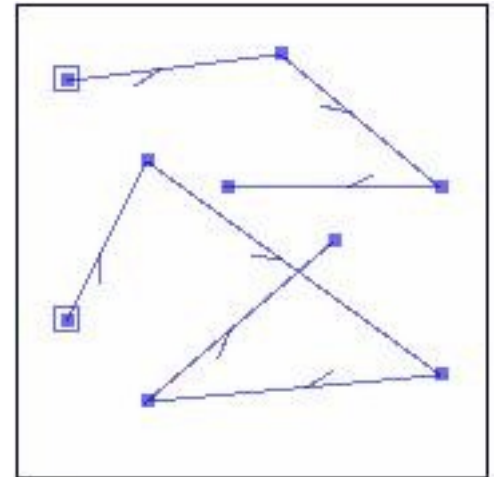
```
(  
  0 5, 5 1, 9 4, 2 14, 14 13, 4 4  
)
```



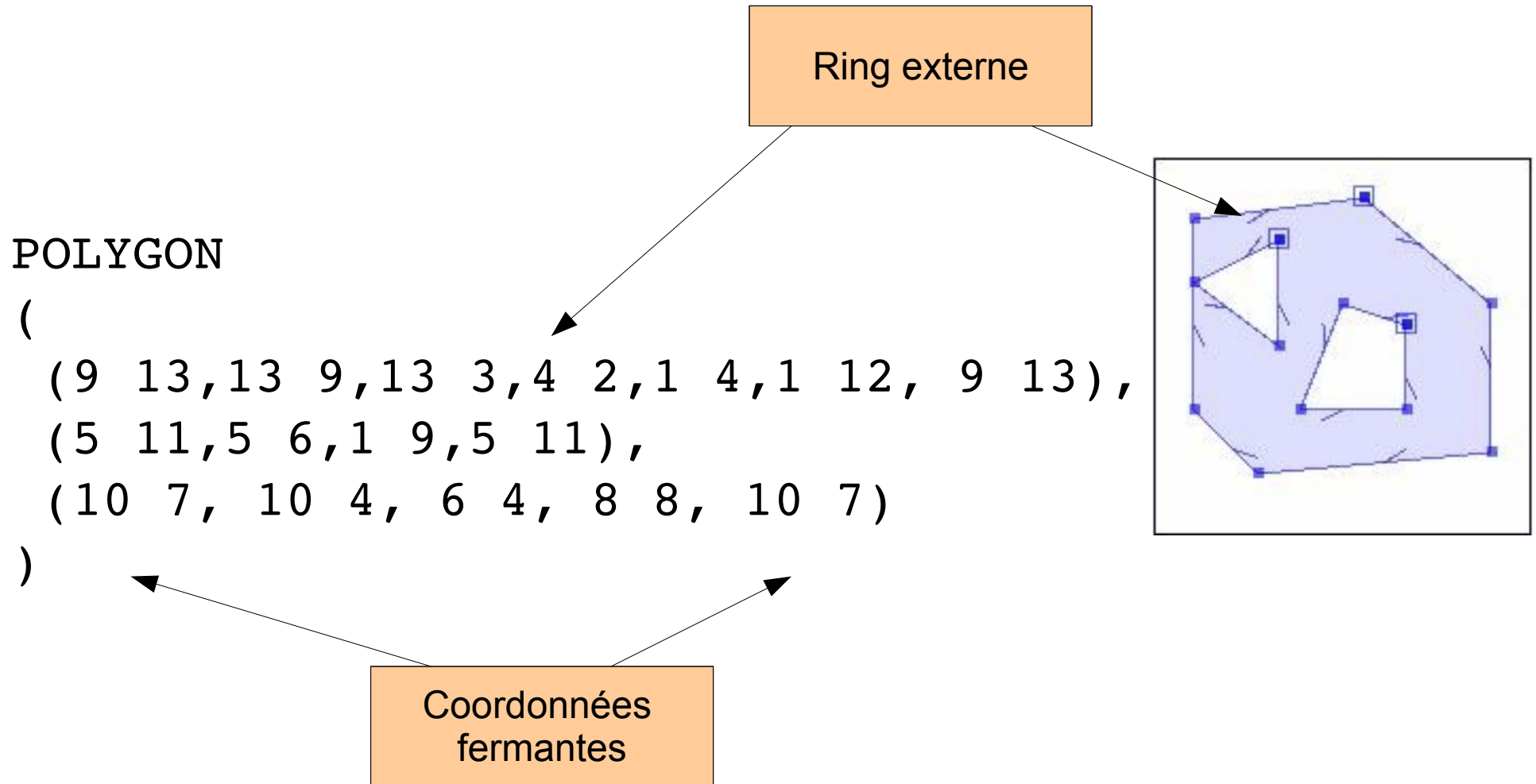
WKT: MULTILINESTRING

MULTILINESTRING

```
(  
  (1 5, 3 9, 14 3, 3 2, 10 8),  
  (1 13, 8 13, 14 9, 7 9)  
)
```



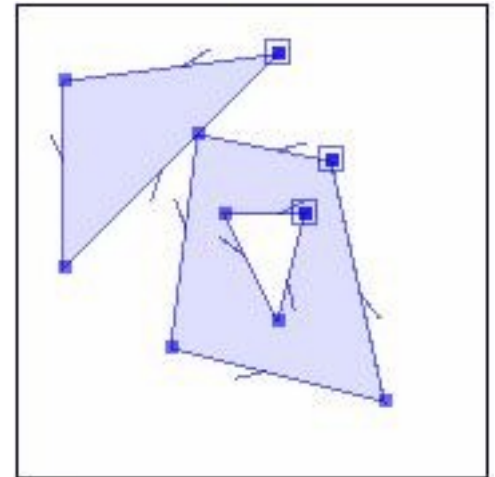
WKT: POLYGON



WKT: MULTIPOLYGON

MULTIPOLYGON

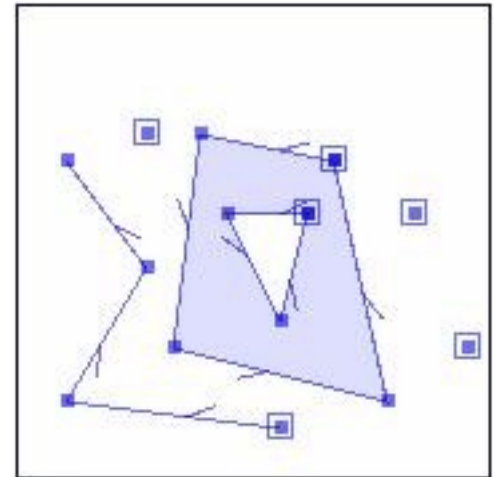
```
(  
  ((10 14,1 8, 1 13,10 14)),  
  (  
    (12 9, 13 3, 5 5, 7 10, 12 9),  
    (13 7, 12 8, 11 7, 13 7)  
  )  
)
```



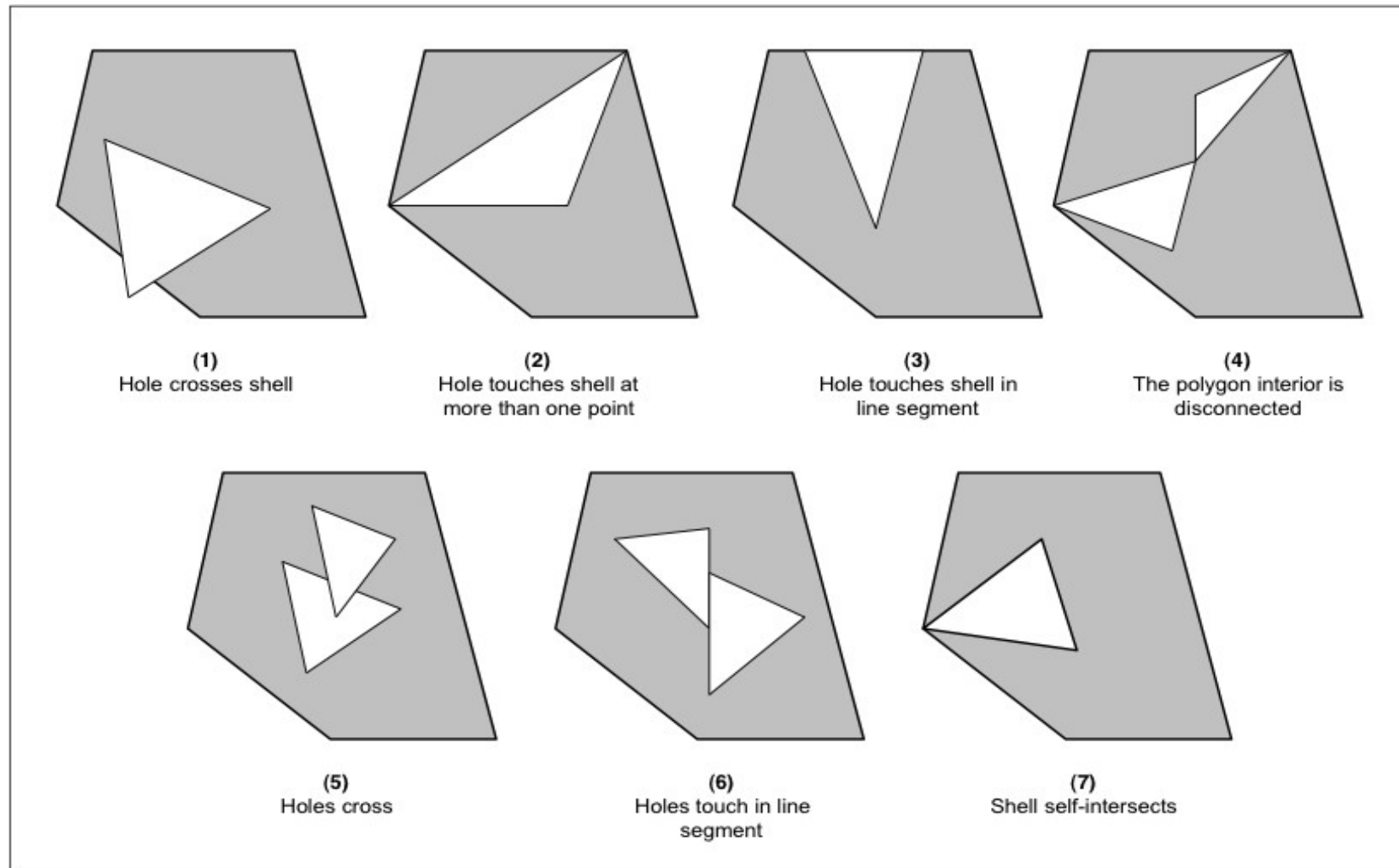
WKT: GEOMETRYCOLLECTION

GEOMETRYCOLLECTION

```
(  
  MULTIPOINT(4 10, 12 9, 14 4),  
  LINESTRING(8 2, 2 3, 4 7, 2 9),  
  POLYGON  
  (  
    (10 9, 11 3, 5 5, 6 10, 10 9),  
    (9 7, 8 6, 7 7, 9 7)  
  )  
)
```



Limites du modèle objet SFS



Objets non représentables par des polygones



Format binaire et natif

- Besoin format binaire pour optimiser stockage et accès aux données:
 - Pour WKT: WKB
 - Pour EWKT: EWKB
- En natif PostGIS utilise pour le stockage
 - Un encodage Héxa de EWKB: HEWKB
 - Depuis la version 1.0

Interfaces entre (E)WKT et Geometry

- Les interfaces suivantes permettent de manipuler les données WKT et EWKT :

```
Text WKT = ST_AsText(geometry);
```

```
Text EWKT = ST_AsEwkt(geometry);
```

```
Geometry = ST_GeomFromText(text WKT, SRID);
```

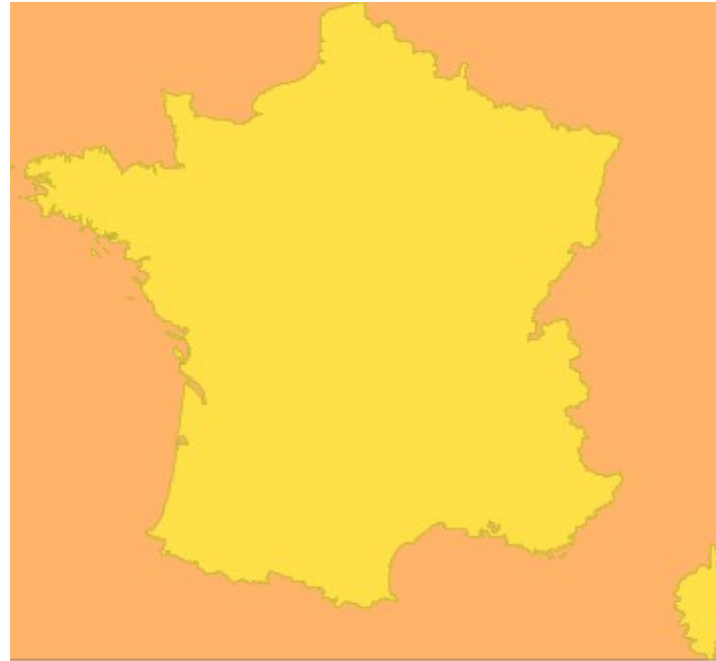
```
Geometry = ST_GeomFromEWKT(text EWKT);
```



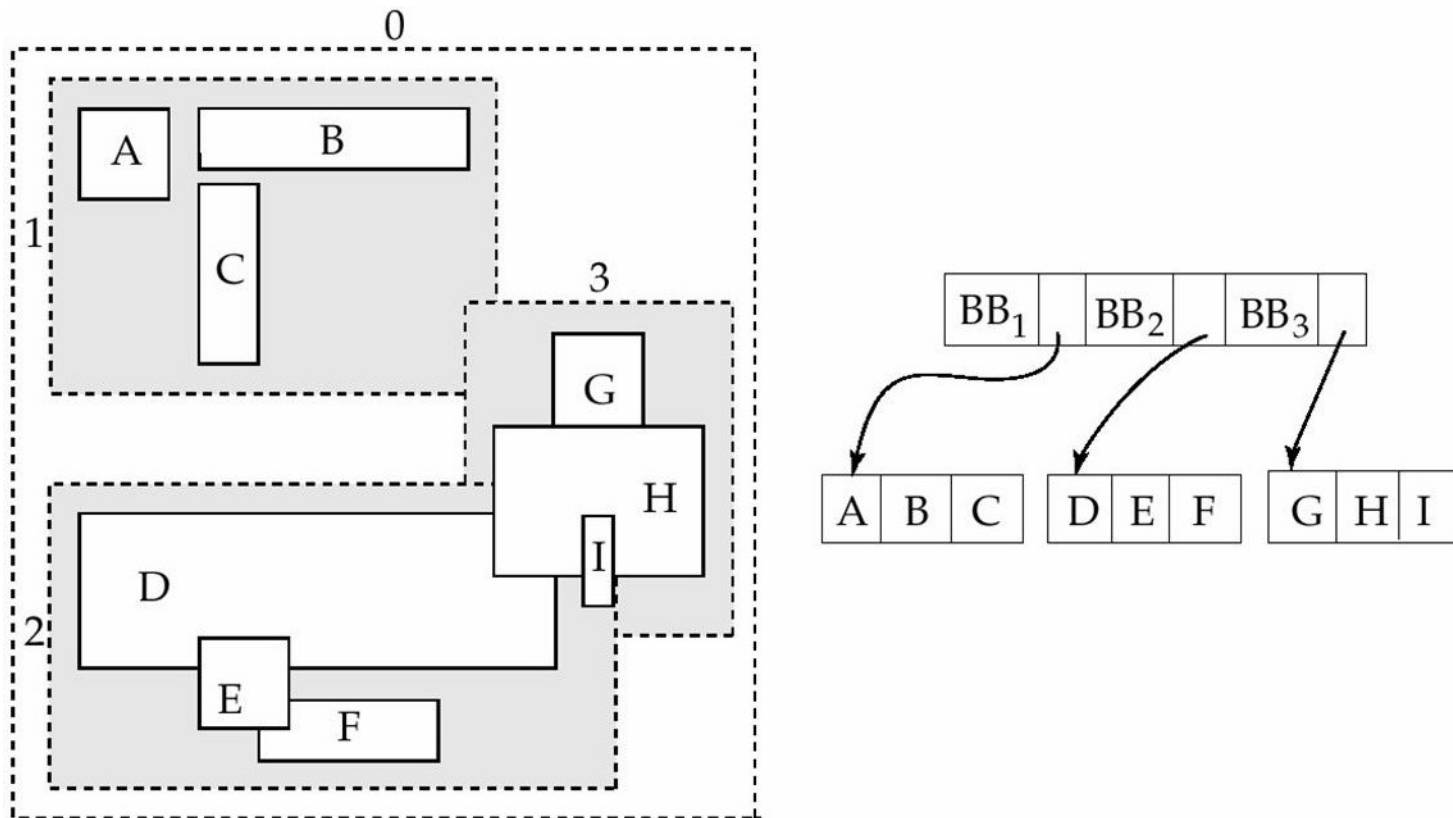
Index Spatiaux

Index Spatiaux - Bbox

- Rectangle englobant = bounding box = bbox
- Surface rectangulaire recouvrant un objet
- Utilisé comme approximation de la géométrie (index, opérateurs primaires, etc)



Index Spatiaux - Principes



Principe du R-Tree: regroupement des rectangles englobants des objets dans des régions de l'index

Index Spatiaux - Utilisation

- Cas d'utilisation
 - SELECT avec des clauses WHERE sur colonne spatiale
 - A évaluer précisément dans le cas de table avec des requêtes en RW importantes
- Création d'un index spatial:

Nom de l'index

Nom de la table spatiale

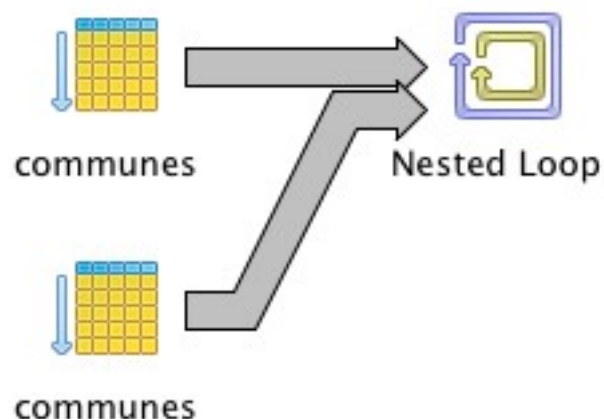
```
CREATE INDEX index_dept ON departement USING GIST  
( the_geom GIST_GEOMETRY_OPS );
```

Nom de la colonne spatiale

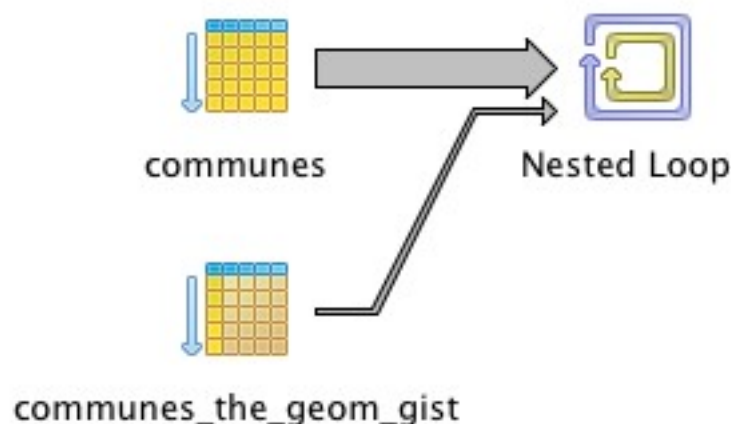
Index Spatiaux - Limitations

- Mise à jour est gourmande: Ils ralentissent de fait les tâches d'écriture en base (INSERT / UPDATE)
- Pas toujours utilisés par le moteur de requête (exemple de peu d'objets géographiques volumineux)
- Vérifier leur utilisation avec la commande EXPLAIN:

```
EXPLAIN SELECT c1.nom FROM com.communes c1, com.communes c2  
WHERE c2.nom = 'Toulouse' and st_touche(c1.the_geom,  
c2.the_geom);
```



Sans index: scan séquentiel
des tables

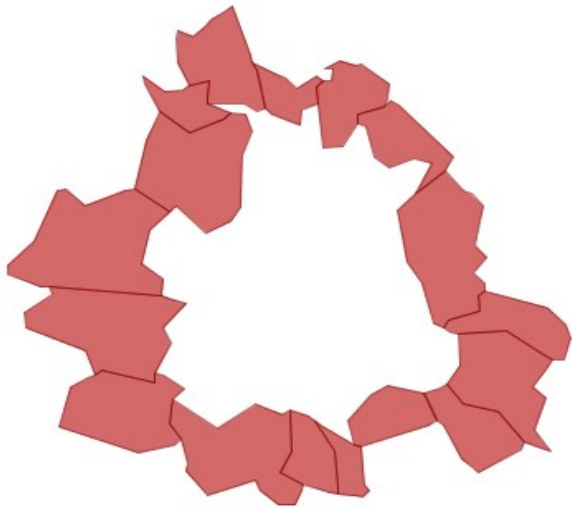


Avec index: filtre géographique
sur la table indexée

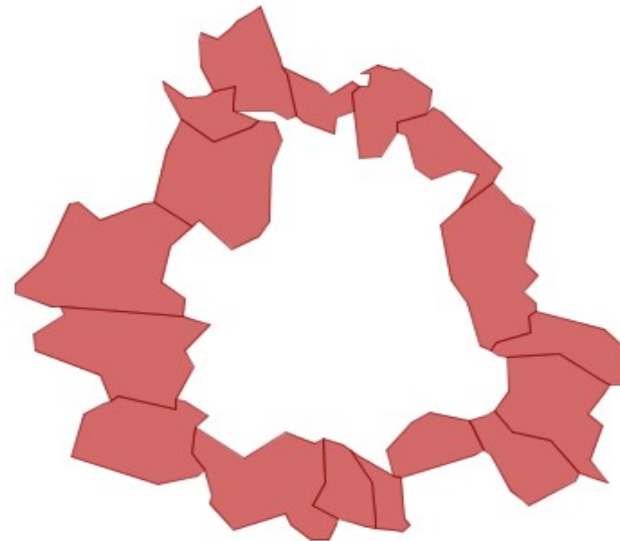
Index – exemple

- Trouver les communes qui touchent Toulouse:

```
SELECT c1.nom FROM communes c1, communes c2 WHERE c2.nom =  
'Toulouse' AND st_touches(c1.the_geom, c2.the_geom);
```



Sans index: temps = 150 ms



Avec index: temps = 30 ms

Meme résultat, mais gain de temps important grâce aux index



Fonctions et opérateurs spatiaux

Opération 'Topologique'

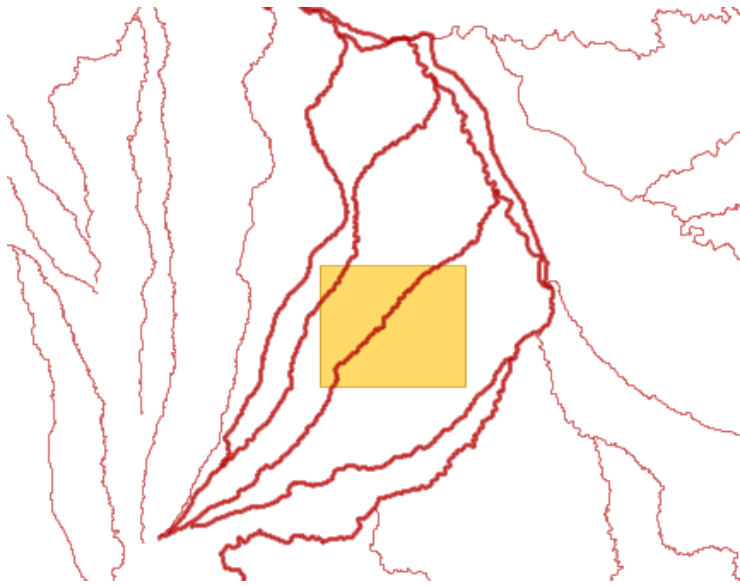
- Ensemble des calculs réalisés par PostGIS et GEOS
- GEOS
 - Portage en C++ de JTS
 - geos.refractory.net
- Permet de réaliser des opérations complexes: buffer, union, découpage, etc.

Opérateurs ou fonctions ?

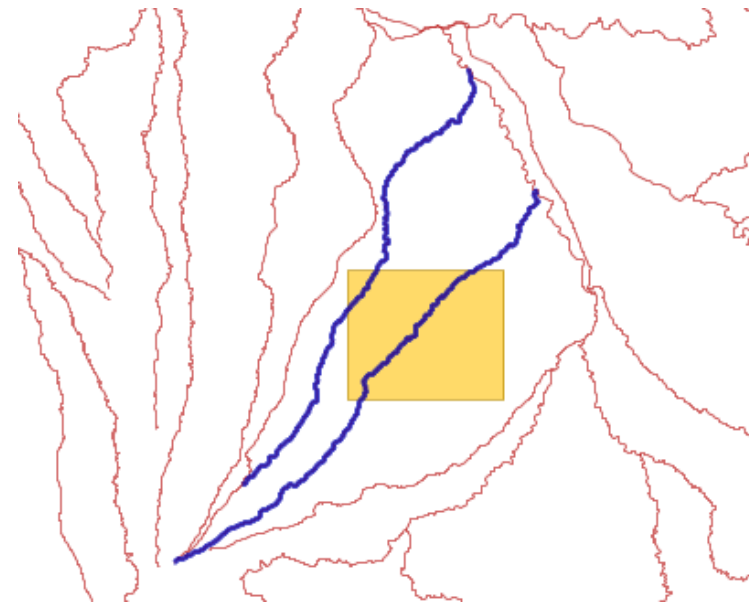
- Les opérateurs fonctionnent à partir des rectangles englobants minimums des géométries (bounding box)
- Les fonctions utilisent la géométrie exacte.
- Les opérateurs mettent également à profit les index spatiaux basés sur les bbox.

Bbox vs objets

- Rivières intersectant une zone d'intérêt



```
select * from rivieres  
where the_geom &&  
'BOX(476309 1823131, 508515 1849654)')
```



```
select * from rivieres  
where st_intersects(  
the_geom ,  
'BOX(476309 1823131, 508515 1849654)')
```

Fonctions et Opérateurs spatiaux

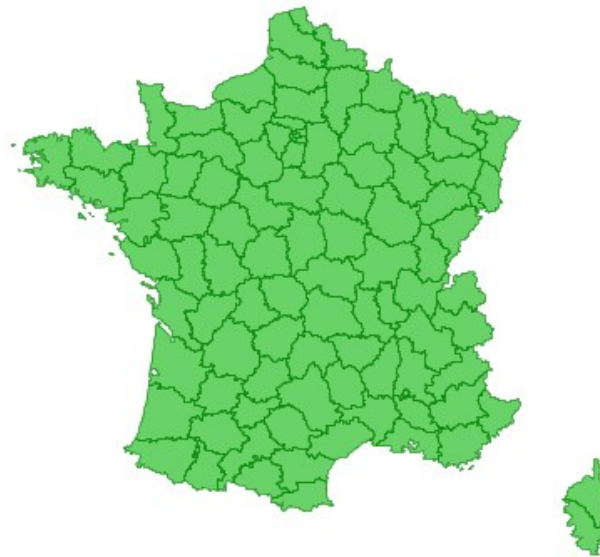
- **Opérations booléennes de localisation** des objets entre eux: (touches, intersects, overlaps, relate, etc.)
- **Calcul sur les objets** (distance, area, distance_sphere, etc.)
- **Construction de nouveaux objets** (buffer, intersection, union, difference, symmetric_difference, etc.)

Exemple de Fonctions: agrégation

- But: unir des objets géographiques liés par un même attribut
- Equivalent du « dissolve » d'ESRI
- Ex: construire les départements de France a partir des communes
- `select st_union(the_geom) from communes group by code_dep;`



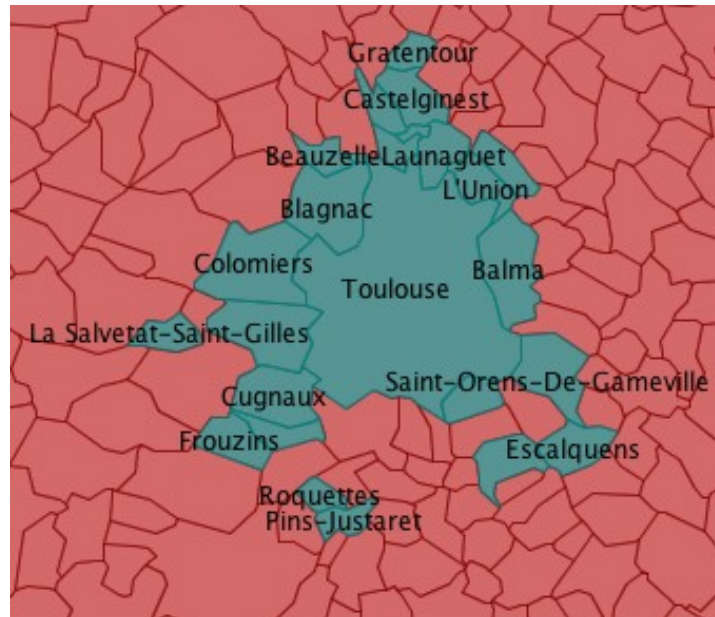
Les communes de France



Les communes de France fusionnées par département

Exemple de Fonctions: analyse spatiale

- But: Trouver toutes les communes de la Haute-Garonne dont la densité de population est supérieure à 600 habitants/km²
- ```
select nom, pop_tot / (st_area (the_geom) / 1000000) as densite, the_geom from com.communes where code_dep = '31' and pop_tot / (st_area (the_geom) / 1000000) > 600 order by densite;
```





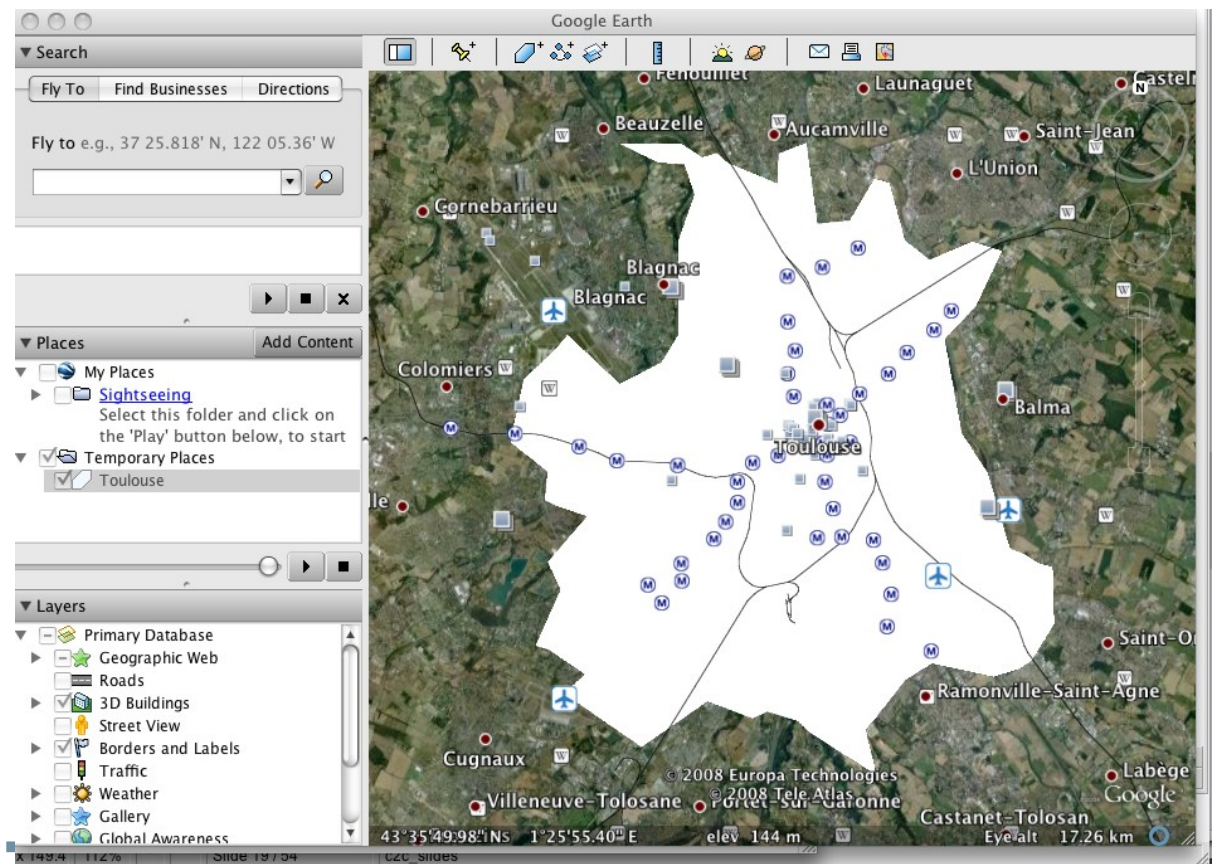
---

# Import et Export de données

---

# Formats d'export

- GML (Geography Markup Language): référence OGC, utilisé pour le WFS par exemple
  - Select st\_asgml(the\_geom) from communes;
- KML (Format Google Earth pour les objets géographiques)
  - Select st\_askml(the\_geom from communes where nom = 'Toulouse';



# Formats d'export

- GeoJson (format compact d'echange de données géographiques, essentiellement utilisé dans des applications Web)
  - `Select st_asgeojson(the_geom) from communes;`

## Vector Formats Example

Shows the wide variety of vector formats that open layers supports.



Use the drop-down below to select the input/output format for vector features. New features can be added by using the drawing tools above or by pasting their text representation below.

Format  Pretty print

Input Projection:

Output Projection:

# Shp2pgsql: Présentation

- Commande fournie avec PostGIS
- Conversion Shapefile vers du SQL
- Chargement en deux étapes:
  - Conversion SHP vers SQL
  - Chargement du SQL dans Postgres/PostGIS
- Dispose de multiples options
  - `-s` précise le système de projection
  - `-I` génère un index spatial
  - `-S` utilise des géométries 'simples'
  - `-d` Drop la table avant de la récréer
  - `-D` Utilise le format DUMP de PostgreSQL
  - `-W` Encodage des données attributaires
- `shp2pgsql -iID -s 27582 commune.shp communes | psql -U user -d ma_base`



# Pgsql2shp: Présentation

- Commande fourni avec PostGIS
- Pendant de shp2pgsql
- Permet de générer un shapefile à partir de:
  - Table spatiale
  - Requête spatiale
- `pgsql2shp -u postgres ma_base communes`
- `pgsql2shp -u postgres ma_base 'SELECT code_dep, st_union(the_geom) as the_geom FROM communes group by code_dep'`



---

# Outils SIG OpenSource

---



# Bureautiques

- Commerciaux ou libres supportent PostGIS

|               | Lecture | Écriture | Requête spatiales <sup>(1)</sup> |
|---------------|---------|----------|----------------------------------|
| MapInfo       | Oui     | Non      | Non                              |
| ArcGIS        | Oui     | Oui      | Non                              |
| FME           | Oui     | Oui      | Oui                              |
| CadCord SIS   | Oui     | Oui      | Non                              |
| Ionic         | Oui     | Non      | Non                              |
| ...           |         |          |                                  |
| Udig          | Oui     | Oui      | Non                              |
| Grass         | Oui     | Oui      | Oui                              |
| Jump/openJump | Oui     | Oui      | Oui <sup>(2)</sup>               |
| GvSig         | Oui     | Oui      | Oui <sup>(2)</sup>               |
| SDI           | Oui     | Oui      | Oui                              |
| Qgis          | Oui     | Oui      | Non                              |
| ...           |         |          |                                  |

(1) exécution de Requêtes spatiales arbitraires - (2) plugin nécessaire

# Serveurs cartographiques

- MapServer:
  - CGI écrit en C,
  - Outil 'historique',
  - Très bon support de PostGIS
  - Mise en oeuvre simple
  
- GeoServer:
  - Servlet Java,
  - Implémentation stricte des normes OGC (WMS, WFS, WFS-T)
  - Permet l'édition de features stockées dans PostGIS (WFS-T)
  - Mise en oeuvre plus délicate



---

# Futur de PostGIS

---

# RoadMap Court Terme

- 1.3.4 est en phase de finalisation
- 1.4.0 est en phase de développement
  - Dissociation entre librairies bas niveau SFS et librairie de fonctions spatiales
  - Travail important de refactoring
  - ...

# RoadMap Long Terme

- PostGIS 2.0
- Points marquants
  - Continuer à se rapprocher de SQL/MM
  - Compatible SFS 2.0
  - Logique de Topologie et Routing native
  - ...
- Roadmap intégrale :
  - [http://docs.google.com/View?docid=dg99qr76\\_2dgt26j](http://docs.google.com/View?docid=dg99qr76_2dgt26j)



---

# Conclusion

---

# En guise de conclusion

- PostGIS est la solution de référence en terme de cartouche spatial OpenSource
- Elle s'intègre et enrichie les possibilités d'un SGBDR
- Cette application permet de très nombreux traitements SIG sur des géométries vectorielles
- Elle se couple à de nombreux autres outils SIG existants
- Perspectives sur l'utilisation de Spatial OLAP et/ou de traitements géostatistiques.

# Questions

- En direct...
- Par mail:
  - Olivier Courtin - [olivier.courtin@camptocamp.com](mailto:olivier.courtin@camptocamp.com)
  - Nicolas Ribot - [nicolas.ribo@camptocamp.com](mailto:nicolas.ribo@camptocamp.com)