



PostGIS Integration Tips

PG Session #7 - 2015 - Paris

A quoi sert un SIG ?

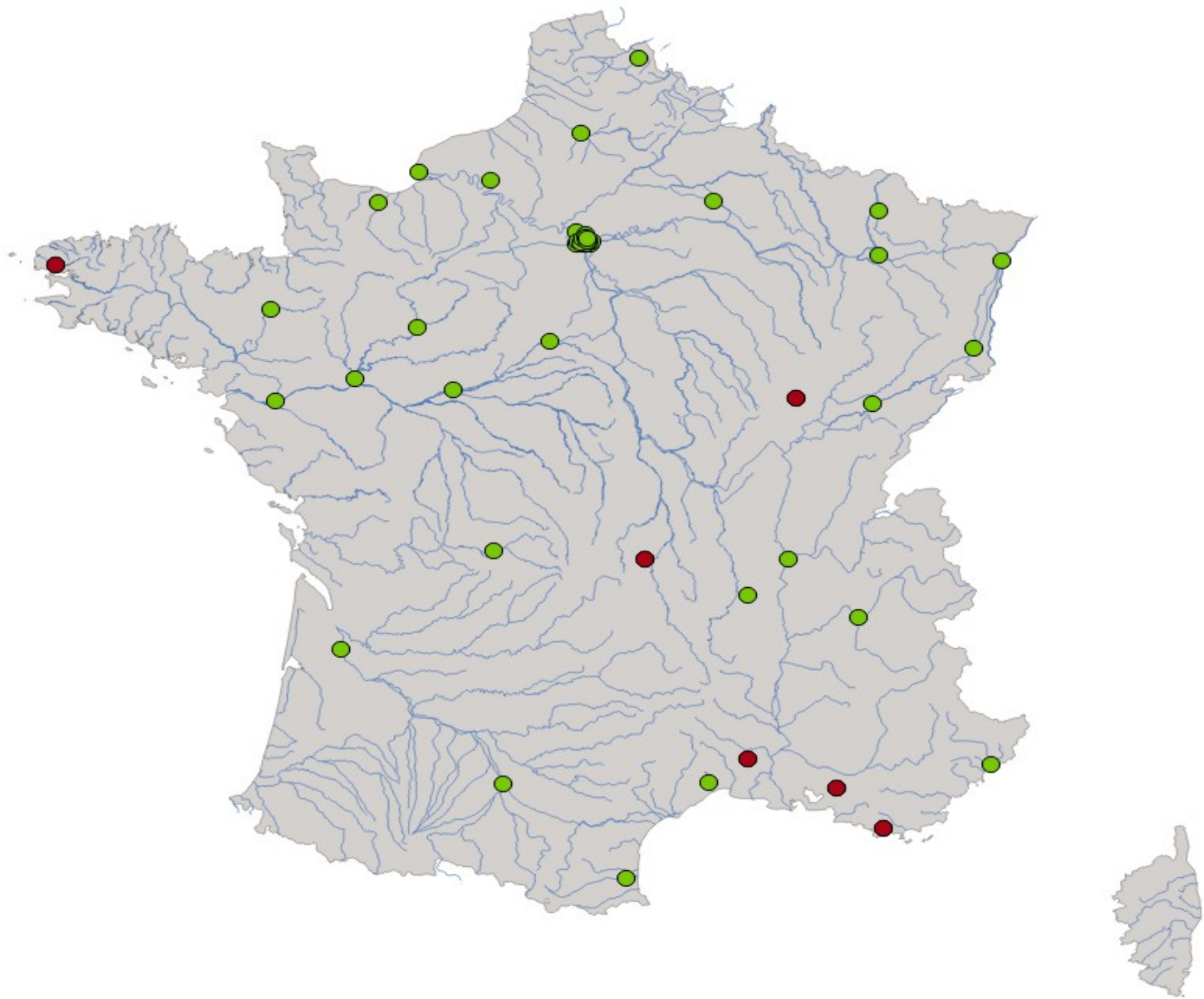
« Fleuve, Pont, Ville... »

```
SELECT nom_comm
FROM commune
WHERE      population * 1000 > 100000
          AND code_dept != '75'

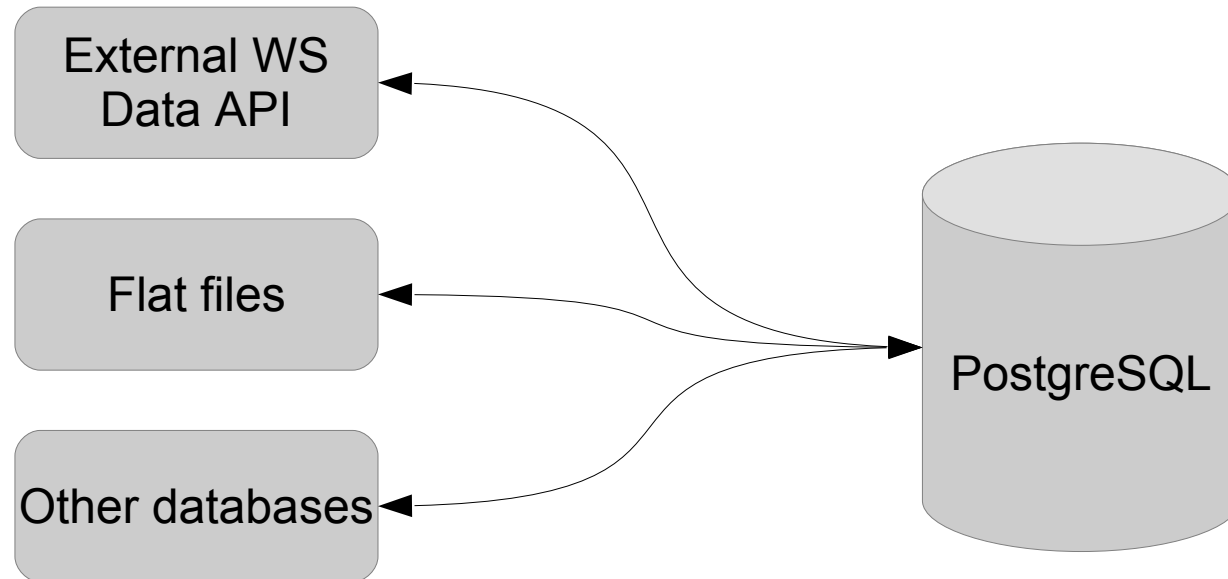
EXCEPT

SELECT DISTINCT nom_comm
FROM commune co, cours_eau ce
WHERE
      Population * 1000 > 100000
      AND ce.classe='1'
      AND co.toponyme != ''
      AND ST_Intersects(co.geom, ce.geom)
```

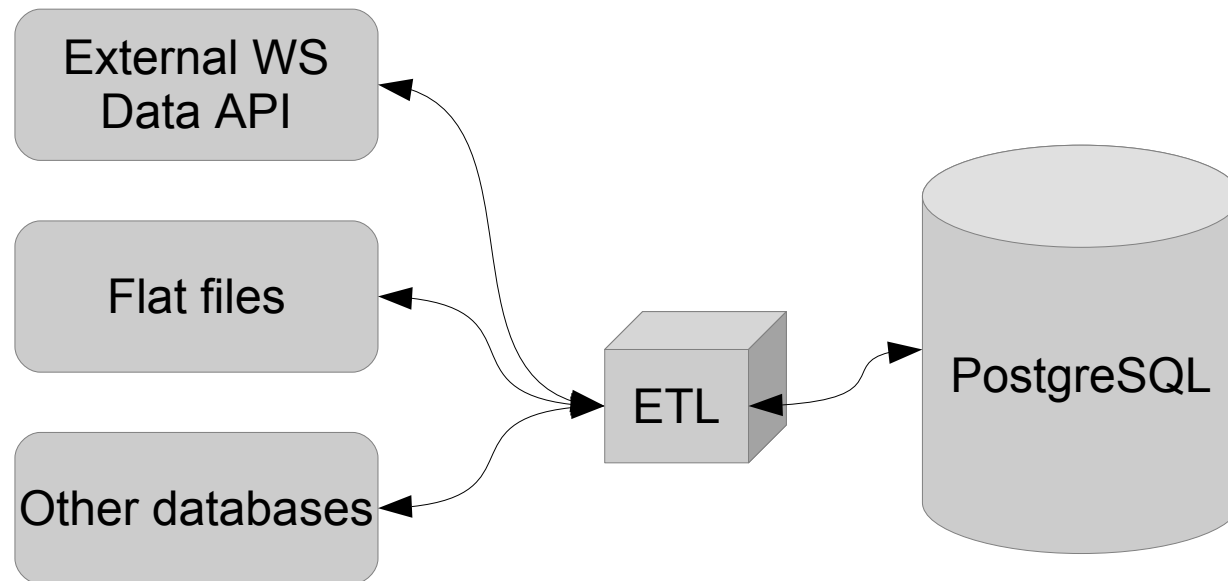
CLERMONT-FERRAND
NIMES
BREST
DIJON
AIX-EN-PROVENCE
TOULON



#1 Data Intégration

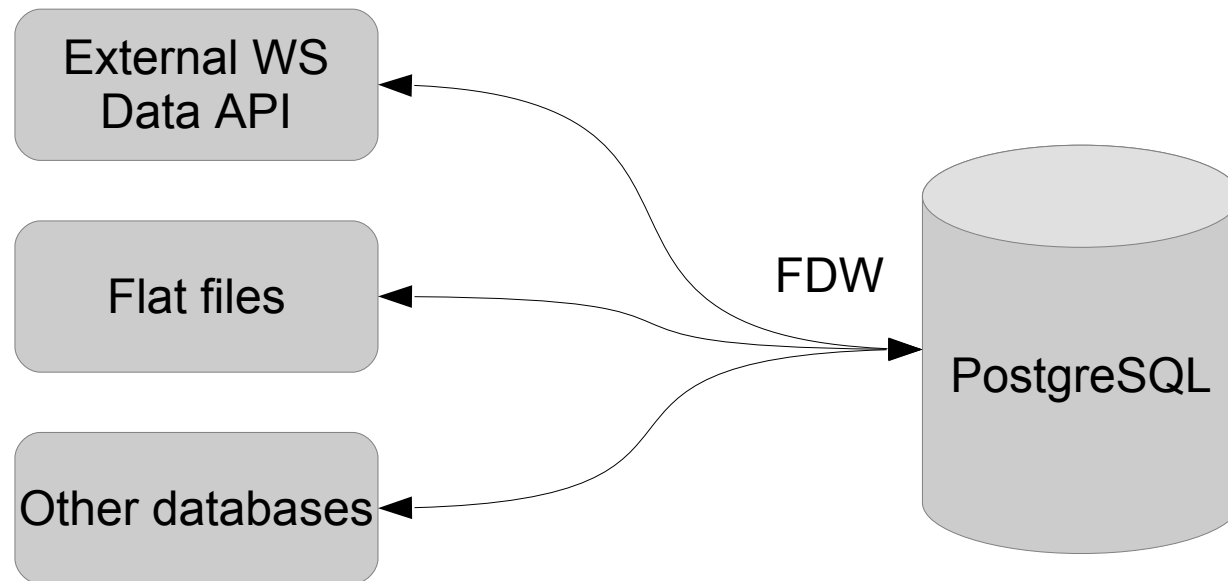


#1 Data Intégration



Réponse courante « Utilisez un ETL »

#1 Data Intégration



Réponse alternative :
« Utilisez les Foreign Data Wrapper de PostgreSQL »

https://wiki.postgresql.org/wiki/Foreign_data_wrappers

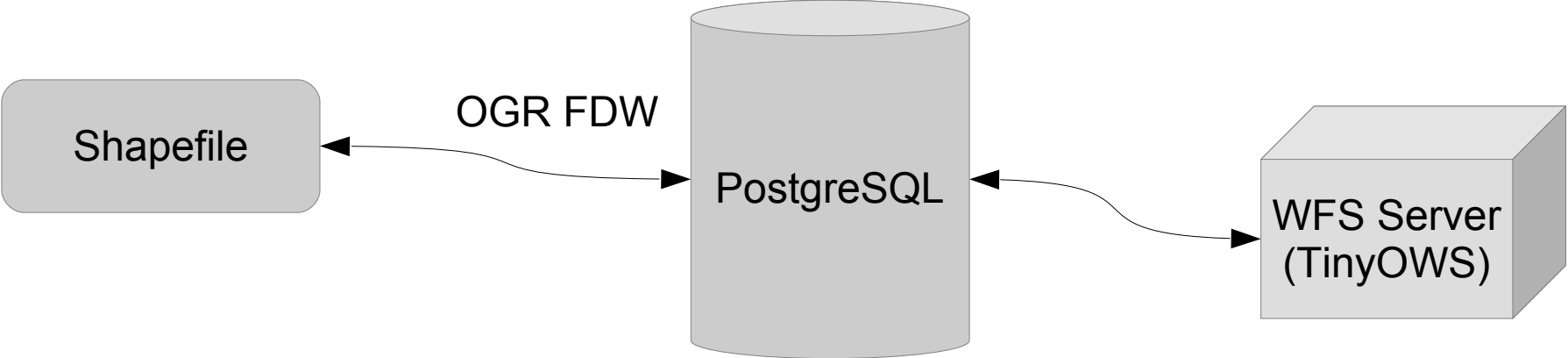
SQL Management of External Data (SQL/MED) ajouté au standard SQL
Permet l'accès à des objets distants à la base de données via du SQL
Disponible dans PostgreSQL depuis la 9.3

https://wiki.postgresql.org/wiki/Foreign_data_wrappers

SQL Management of External Data (SQL/MED) ajouté au standard SQL
Permet l'accès à des objets distants à la base de données via du SQL
Disponible dans PostgreSQL depuis la 9.3

- 1 Foreign Data Wrappers
 - 1.1 Generic SQL Database Wrappers
 - 1.2 Specific SQL Database Wrappers
 - 1.3 NoSQL Database Wrappers
 - 1.4 File Wrappers
 - 1.5 Geo Wrappers
 - 1.6 LDAP Wrappers
 - 1.7 Generic Web Wrappers
 - 1.8 Specific Web Wrappers
 - 1.9 Big Data Wrappers
 - 1.10 Column-Oriented Wrappers
 - 1.11 Scientific Wrappers
 - 1.12 Operating System Wrappers
 - 1.13 Exotic Wrappers
 - 1.14 Example Wrappers

~50 connecteurs natifs déjà disponibles
(et plus encore via l'extension Multicorn)



<https://github.com/pramsey/pgsql-ogr-fdw>

Installation de OGR FDW

```
git clone https://github.com/pramsey/pgsql-ogr-fdw.git
cd postgresql-ogr-fdw
make
sudo make install
```

Définition d'un FDW wrapper

```
CREATE EXTENSION postgis;  
CREATE EXTENSION ogr_fdw;  
  
CREATE SERVER shapefile_france  
  FOREIGN DATA WRAPPER ogr_fdw  
  OPTIONS (  
    datasource '/tmp/fdw_ogr/france.shp',  
    format 'ESRI Shapefile'  
  );
```

Récupération des données attributaires du shapefile

```
ogrinfo -al -so /tmp/fdw_ogr/france.shp
```

Création de la Foreign Table

```
CREATE SCHEMA shp;  
  
CREATE FOREIGN TABLE shp.france (  
    id_geofla integer,  
    geom geometry,  
    code_chf_l varchar,  
    nom_chf_l varchar,  
    x_chf_lieu varchar,  
    y_chf_lieu varchar,  
    x_centroid integer,  
    y_centroid integer,  
    nom_dept varchar,  
    code_reg varchar,  
    nom_region varchar,  
    code_dept varchar  
)  
  
SERVER shapefile_france  
OPTIONS (layer 'france');
```

Vérification

```
SELECT id_geofla, ST_AsEWKT(ST_Centroid(geom)) AS geom  
FROM shp.france LIMIT 1 ;
```

Création d'une vue pour la Foreign Table

<https://github.com/pramsey/pgsql-ogr-fdw/issues/11>

```
CREATE OR REPLACE VIEW shp.france_wfs AS  
  
SELECT id_geofla,  
       ST_Multi(ST_SetSRID(geom,27572))::geometry(MultiPolygon,27572) AS geom,  
       code_dept,  
       nom_dept  
FROM france;
```

Configuration TinyOWS

```
<tinyows online_resource="http://127.0.0.1/cgi-bin/tinyows"
  schema_dir="/usr/local/share/tinyows/schema/"
  estimated_extent="1"
  display_bbox="0">

  <pg host="127.0.0.1" user="pggis" password="***" dbname="db" />

  <metadata name="TinyOWS WFS Server"
    title="TinyOWS Server – OGR FDW Service" />

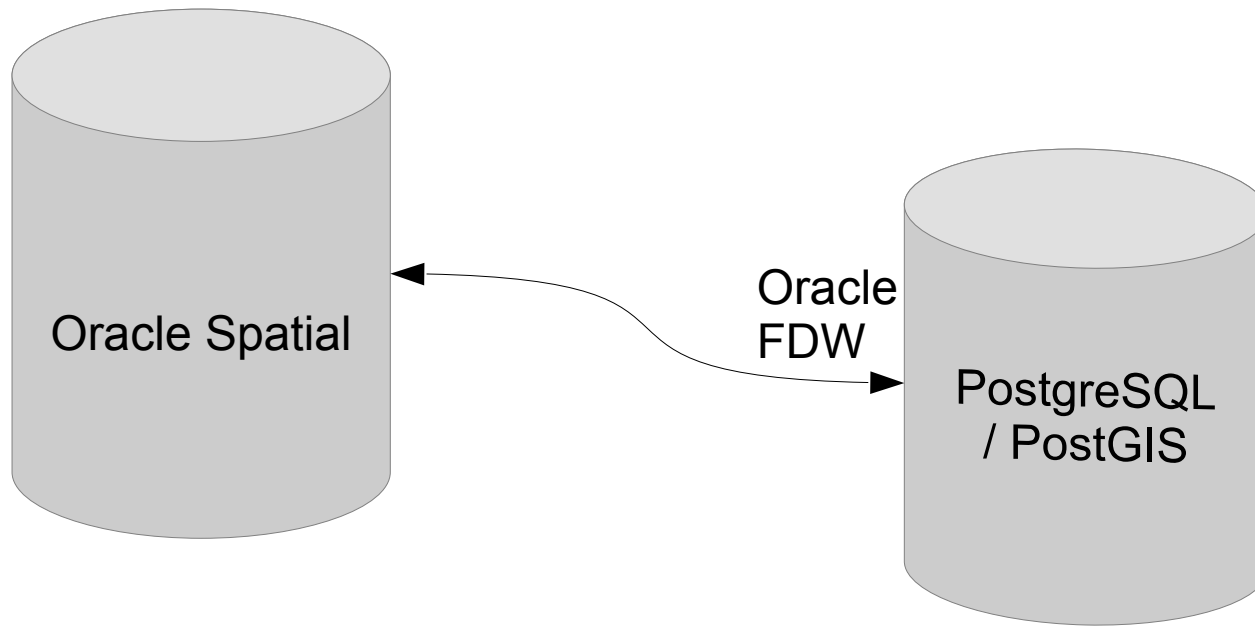
  <layer retrievable="1"
    writable="0"
    ns_prefix="tows"
    ns_uri="http://www.tinyows.org/"
    schema="shp"
    name="france_wfs"
    title="france" />

</tinyows>
```

Vérification

```
wget -O out http://127.0.0.1/cgi-bin/tinyows?
SERVICE=WFS&REQUEST=GetFeature&Typename=tows:france_wfs
```

#2 Data intégration : Oracle Spatial



http://pgxn.org/dist/oracle_fdw/


```
CREATE EXTENSION postgres_fdw;  
CREATE EXTENSION oracle_fdw;  
  
CREATE SERVER orcl FOREIGN DATA WRAPPER oracle_fdw  
OPTIONS (dbserver '${ORACLE_URI}');
```

Mapping utilisateur Oracle

```
GRANT USAGE ON FOREIGN SERVER orcl TO ${PGUSER};  
  
CREATE USER orcl_map FOR ${PGUSER}  
SERVER orcl  
OPTIONS (user '${ORAUSER}', password '${ORAPWD}');
```

```
CREATE SCHEMA fdw;
```

```
CREATE FOREIGN TABLE fdw.foo (  
    id double precision,  
    label varchar,  
    last_update date,  
    geom geometry(POINT, 2154),  
)
```

```
SERVER orcl
```

```
OPTIONS (schema '${ORAUSER}', table 'F00');
```

```
CREATE SCHEMA mat;  
CREATE MATERIALIZED VIEW mat.foo AS SELECT * FROM fdw.foo;
```

```
CREATE UNIQUE INDEX ON mat.foo(id);  
CREATE INDEX ON mat.foo USING GIST(geom);
```

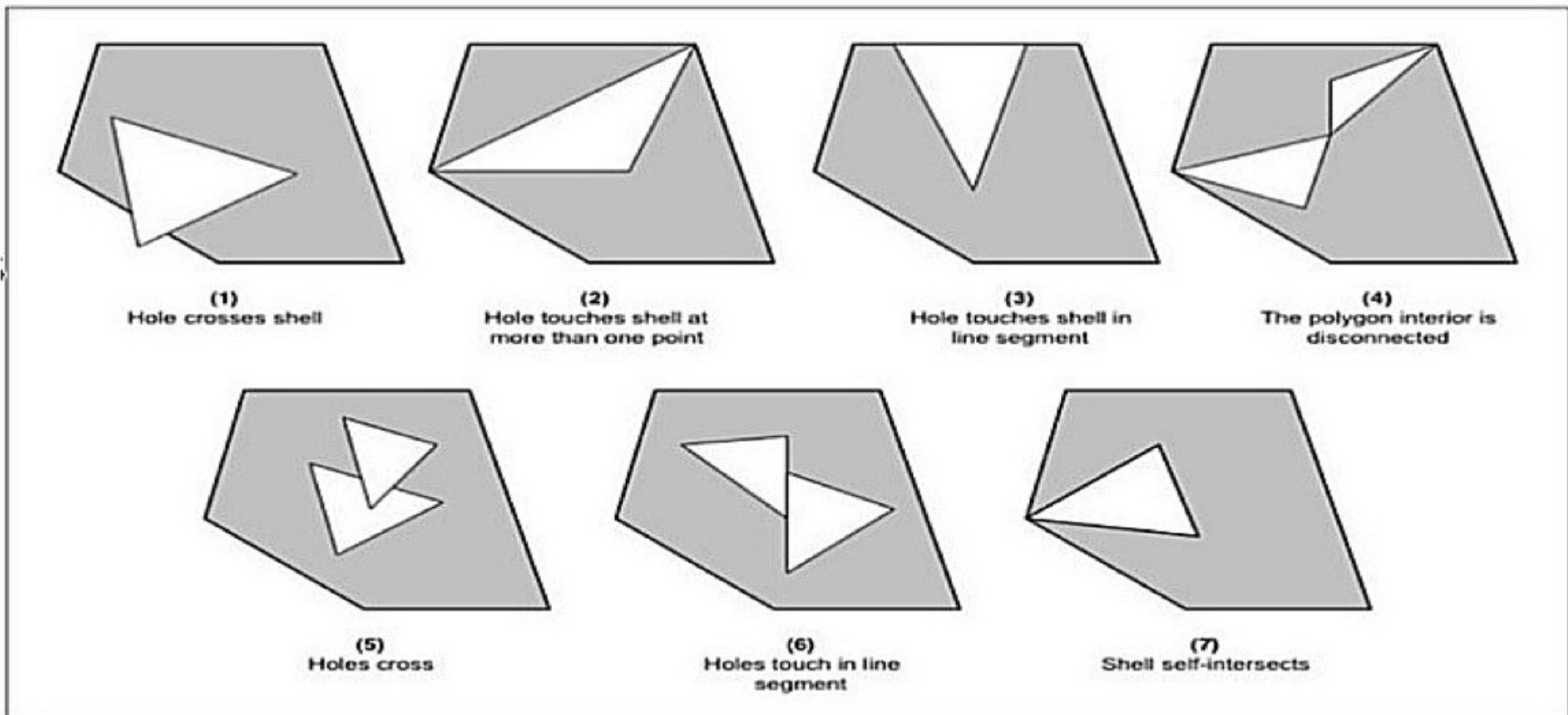
```
REFRESH MATERIALIZED VIEW CONCURRENTLY mat.foo;
```

#3 Data Validity

```
SELECT count(*) FROM my_schema.my_table WHERE NOT ST_IsValid(geom);
```

#3 Data Validity

```
SELECT count(*) FROM my_schema.my_table WHERE NOT ST_IsValid(geom);
```



```
UPDATE my_schema."my_table"
```

```
SET geom=ST_CollectionExtract(ST_MakeValid(geom), 3)
```

```
WHERE ST_IsValidReason(geom) != 'Valid Geometry'
```

```
AND (GeometryType(geom) = 'POLYGON'
```

```
OR GeometryType(geom) = 'MULTIPOLYGON');
```

```
UPDATE my_schema."my_table"
```

```
SET geom=ST_CollectionExtract(ST_MakeValid(geom), 3)
```

```
WHERE ST_IsValidReason(geom) != 'Valid Geometry'  
      AND (GeometryType(geom) = 'POLYGON'  
          OR GeometryType(geom) = 'MULTIPOLYGON');
```

Autres cas à gérer :

- Surface nulle → Empty
- Linéaire avec un seul point → infime translation ending point

#4 Traitement données avec PL/Python

Utilisation librairie Python existante depuis PostgreSQL
Appel depuis fonction SQL

Un exemple avec GeoPy, Installation :

```
sudo apt-get install postgresql-plpython-9.4 python3-geopy  
  
createdb db  
createlang plpython3u db  
psql db -c "CREATE EXTENSION postgis"
```

*S'inscrire sur GeoNames
Activer free Webservice sur votre compte*

Function de géocodage basique en PL/Python

```
CREATE OR REPLACE FUNCTION geoname(toponym text)
    RETURNS geometry(Point,4326)
AS $$

    from geopy import geocoders
    g = geocoders.GeoNames(username="YOUR_USERNAME")

    try:
        place, (lat, lng) = g.geocode(toponym)

        result = plpy.execute(
            "SELECT 'SRID=4326;POINT(%s %s)::geometry(Point, 4326) AS geom"
            % (lng, lat), 1)

        return result[0]["geom"]

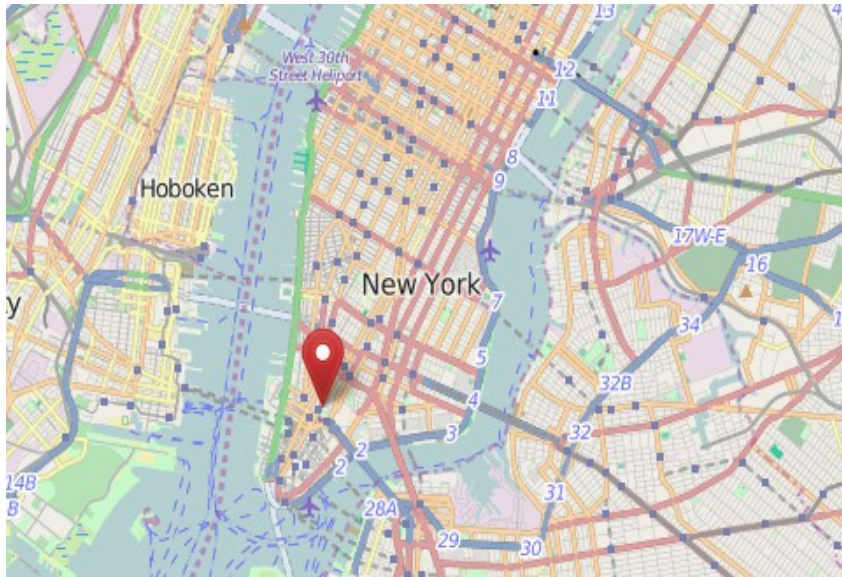
    except:
        plpy.warning('Geocoding Error')
        return None

$$ LANGUAGE plpython3u;
```

Vérification :

```
psql db -c  
"SELECT ST_AsGeoJSON(geoname('New York, NY 10022'))"  
  
{"type":"Point","coordinates":[-74.00597,40.71427]}
```

<http://www.openstreetmap.org/?mlon=-74.00597&mlat=40.71427&zoom=12>



#5 QGIS Db Manager

The screenshot shows the QGIS 2.4.0 Db Manager interface. The main window is titled "SQL window - foo [PostGIS]". On the left, there is a "Layers" panel. The central area contains an SQL query editor with the following text:

```
1 WITH n AS (  
2  
3 SELECT nom_comm FROM commune WHERE population * 1000 > 100000  
4 AND code_dept = '75'  
5  
6 EXCEPT  
7  
8 SELECT DISTINCT nom_comm  
9 FROM commune co, cours_eau ce  
10 WHERE  
11 population * 1000 > 100000  
12 AND ce.classe = '1'  
13 AND toponyme != ''  
14 AND ST_Intersects(co.geom, ce.geom)
```

Below the query editor, there is a button "Execute (F5)" and the text "6 rows, 0.3 seconds". To the right of this is a "Clear" button. Below the execution information is a "Result:" section containing a table with 6 rows and 3 columns: gid, nom_comm, and geom.

	gid	nom_comm	geom
1	1556	BREST	010600002...
2	1882	TOULON	010600002...
3	3253	CLERMONT...	010600002...
4	6547	DIJON	010600002...
5	13666	AIX-EN-PR...	010600002...
6	17592	NIMES	010600002...

Below the table, there is a checkbox "Load as new layer" which is checked. Underneath, there are two dropdown menus: "Column with unique integer values" set to "gid" and "Geometry column" set to "geom". To the right of these is a "Retrieve columns" button. Below that is a "Layer name (prefix)" text input field and a "Load now!" button. At the bottom, there is an unchecked checkbox "Avoid selecting by feature id".

```
;
```

```
(row_number() OVER())::integer AS gid
```



www.oslandia.com

<http://www.oslandia.com/pages/jobs.html>