

# Bases de données spatiales PostgreSQL/PostGIS

Claire Lelong

Septembre 2016



OpenCourseWare  
ocw.unesco-ihe.org



**UNESCO-IHE**  
Institute for Water Education



# Géobase

*Un outil pérenne pour l'archivage, l'interrogation et l'exploitation des informations*

## Données sources



Attributaires



Géographiques

## Géobase



## Clients



Interface de **gestion**  
*administration, mise à jour, traitement, génération de documents*



Client **SIG**  
*Créer des cartes thématiques*



Client **Web**  
*Visualisation via un navigateur*

## Base de données PostgreSQL

PostgreSQL est un puissant système de gestion de données relationnel objets (SGBDRO). Dès le début, il a été conçu de façon à pouvoir accepter des extensions particulières. Une extension de PostgreSQL peut être développée par une équipe de développement indépendante



## Module spatial PostGIS

PostGIS confère au SGBD PostgreSQL le statut de base de données spatiales en ajoutant les 3 supports suivants :

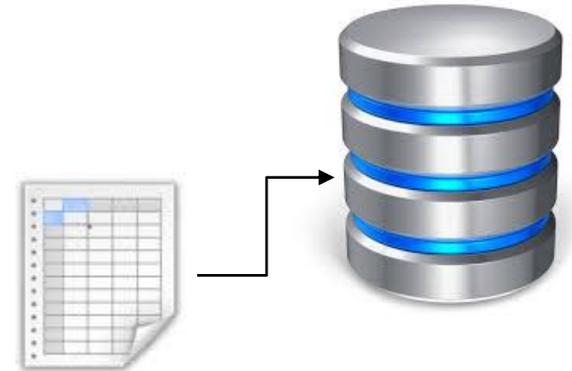
- types de données spatiales,
- index
- Fonctions



PostgreSQL et PostGIS sont développés dans le respect des standards OGC

# Qu'est-ce qu'une base de données ?

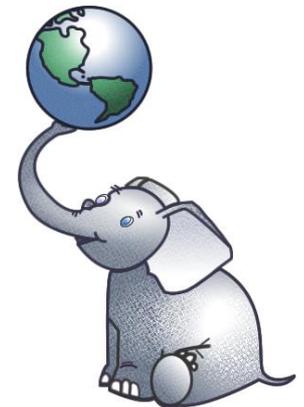
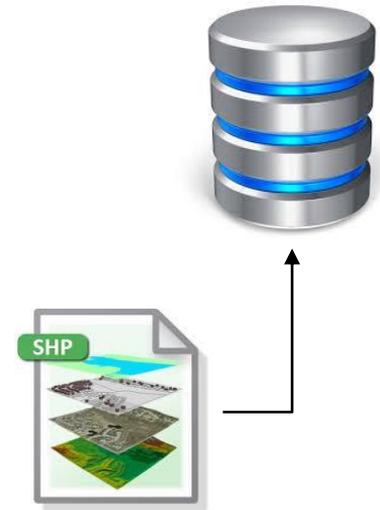
- Ensemble de données structurées
- Données centralisées
- Organisation des données en tables
- Typage des données
- Implémentation de contraintes pour garantir la cohérence du modèle
- Possibilité d'interrogation très puissante avec un langage simple SQL
- Gestion des accès concurrents, transactions
- Gestion de la sécurité



 PostgreSQLFr

# Qu'est-ce qu'une base de données spatiale ?

- Permet de stocker et de manipuler les objets spatiaux comme tout autre objet de base de données
- Bénéficie de la cohérence du modèle des données et de l'organisation en thèmes
- Centralisation des données spatiales
- Interpolation possible des données spatiales avec les données attributaires avec les requêtes SQL
- Nombreuses fonctions spatiales



# Notion de base de données

## Organisation logique des données en tables

### Colonnes

Définition du type de données : texte, numérique, date, géométrie, ...

### Lignes

Une ligne par entité, nombre illimité de lignes

	gid integer	code_bss character varying(50)	aquifere character varying(100)	resultat numeric(7,2)	date_prelevement date	geom geometry(Point,2154)
1	137	07947X0045/PUITS	Alluvions	30.00	2013-11-25	01010000206A080000D4A6E9880FDB29418AF10AEDC08F5841
2	6	07956X0037/D	Alluvions	5.00	2014-04-01	01010000206A080000CC62A5AAB48D2A41945545A6A1925841
3	7	08191X0060/D	Alluvions	5.00	2014-03-20	01010000206A0800003DC66E6063702A418FA71EAF098C5841
4	8	07947X0045/PUITS	Alluvions	8.00	2013-11-25	01010000206A080000D4A6E9880FDB29418AF10AEDC08F5841
5	9	07955X0029/P	Alluvions	5.00	2014-04-01	01010000206A08000054CE96F1784B2A41A2AEC6E89B925841
6	10	07955X0110/P1	Alluvions	5.00	2014-04-01	01010000206A080000A97935735D602A41EDA647BA4E935841

Quelles sont les différences entre une table et une feuille Excel ?

# Modèle des données

Le modèle de données est réalisé lors de la phase de conception de la base de données

## Garantir la cohérence du modèle

### Contraintes

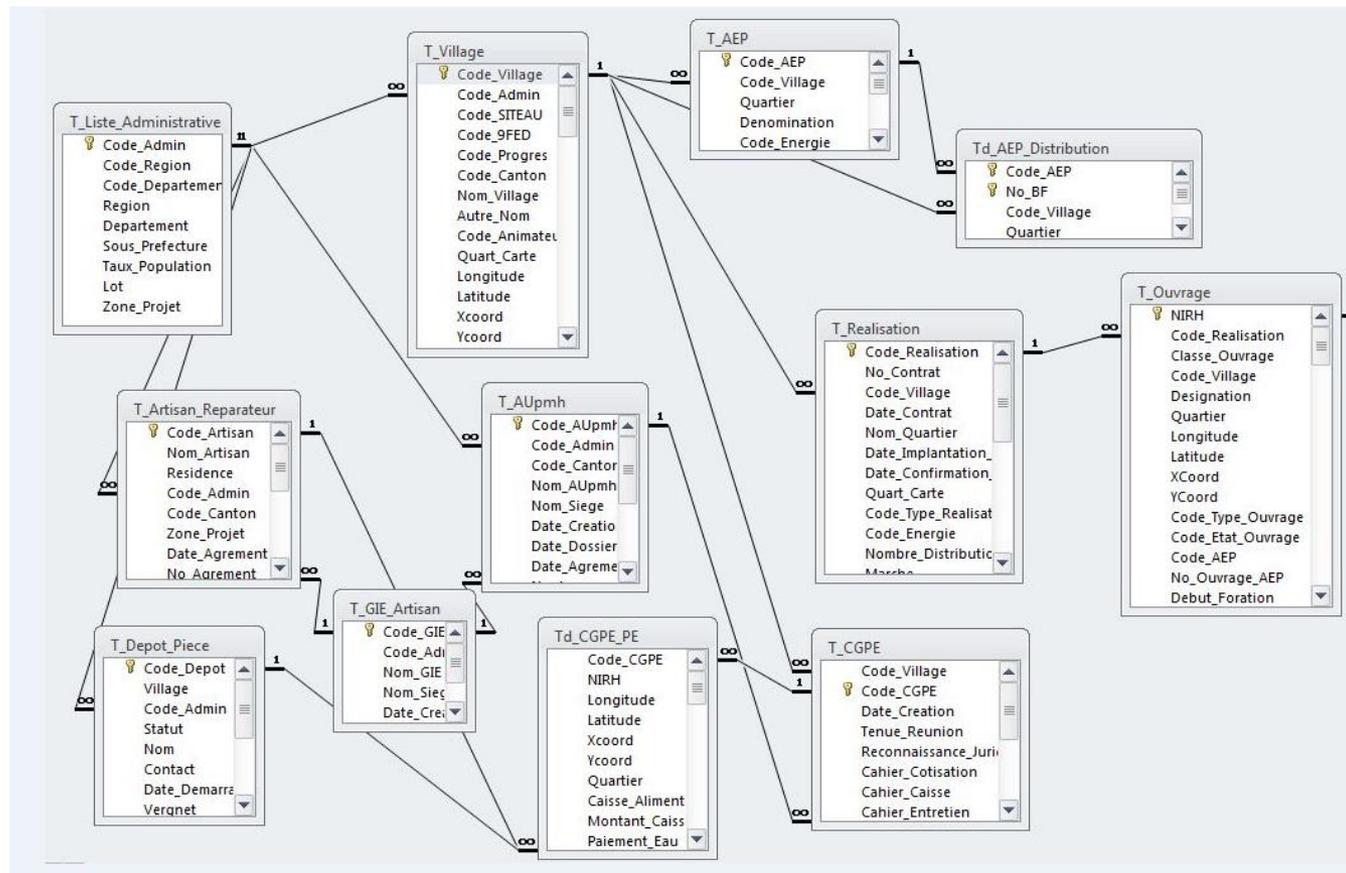
Clé primaire

identifiant

unique et obligatoire

Clé étrangère

lien entre les tables



# Langage SQL

- Structured Query Language
- Seul langage permettant d'agir sur les données
- Créé en 1980, normalisé ISO
- Langage simple et puissant, logique
- Non procédural, exécuté ligne à ligne
- 2 groupes de commandes :

## LDD

Définition des données  
Actions sur les structures  
de données

## LMD

Manipulation des données  
Actions sur les données elles  
mêmes

# Langage SQL

## LDD

Définition des données

**CREATE** création d'objet

**ALTER** modification d'objet

**DROP** destruction d'objet

## LMD

Manipulation des données

**INSERT** ajout de ligne

**UPDATE** mise à jour de ligne

**DELETE** suppression de ligne

**SELECT** consultation des données

**95% des requêtes SQL utilisent la commande SELECT**

# La commande SELECT

## Les clauses de la commande SELECT

**SELECT** <données à afficher>  
**FROM** <tables contenant les données>  
**WHERE** <conditions pour l'affichage des données>  
**ORDER BY** <critères de tri>

Exemple :

```
SELECT gid, code_bss, resultat, date_prelevement
FROM eau.qualite_eso
WHERE aquifere = 'Alluvions'
ORDER BY gid
```

gid integer	code_bss character varying(50)	resultat numeric(7,2)	date_prelevement date
1	08423X0051/P	5.00	2014-05-26
2	08187X0162/P	5.00	2014-05-26
3	08187X0239/F	32.00	2013-09-10
4	08184X0066/DRAIN	5.00	2013-11-19
5	08424X0033/F	5.00	2013-09-11
6	07956X0037/D	5.00	2014-04-01

# Requête de jointure

## Requête sur plusieurs tables

La jointure permet de mettre des tables en relation

Les critères de jointure sont précisés dans la clause FROM

```
FROM table1 INNER JOIN table2 ON table1.gid = table2.gid
```

Exemple :

```
SELECT      c.no_insee, commune, population
```

```
FROM        contexte.commune c INNER JOIN contexte.commune_detail cd
```

```
ON c.no_insee = cd.no_insee
```

```
ORDER BY   commune
```

no_insee character varying(5)	commune character varying(150)	population integer
26004	Alixan	2461
26006	Allex	2576
26007	Ambonil	126
26014	Arthémonay	555
38018	Auberives-en-Royans	371
26021	Autichamp	146
26023	Barbières	844

# Requête d'agrégation

La clause **GROUP BY** permet d'utiliser les fonctions d'agrégations telles que **somme, compter, moyenne, minimum, maximum, ...**

```
SELECT    <critère de groupement et/ou fonctions de groupe>
FROM      <tables>
GROUP BY  <critère de groupement>
```

Exemple :

```
SELECT    bassin_versant, count(gid), sum(volume_capte), avg(volume_capte)
FROM      pression.prelevement
GROUP BY  bassin_versant
```

bassin_versant character varying(150)	count bigint	sum numeric	avg numeric
Véore	274	25259.90	99.0584313725490196
Sud Grésivaudan	374	59143.00	165.2039106145251397
Galaure	199	4877.00	26.9447513812154696
Herbasse	376	17327.40	48.2657381615598886
Drôme	31	692.00	23.8620689655172414

# Particularités d'une base de données spatiale ?

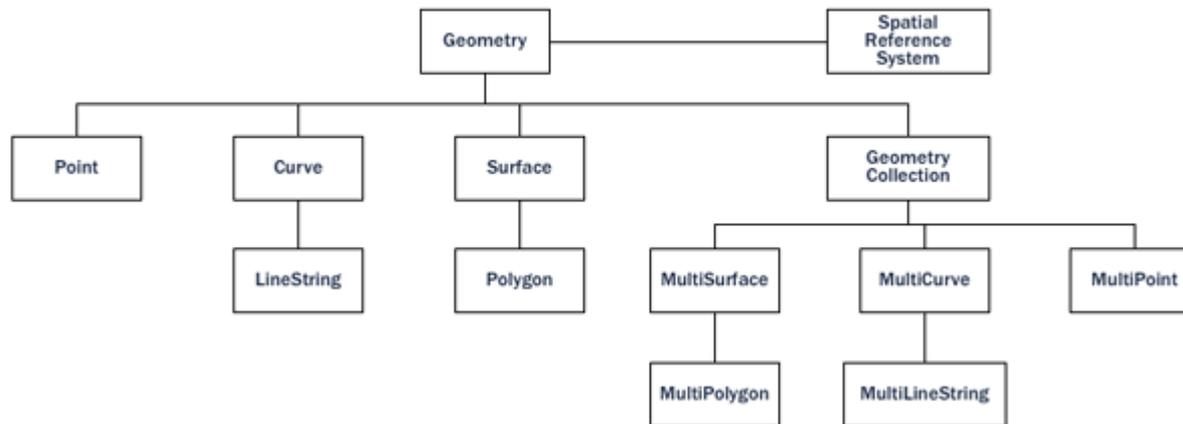
- **Les bases de données spatiales permettent le stockage et la manipulation des objets spatiaux comme les autres objets de la base de données.**
- **Types de données spatiales** fait référence aux géométries de type point, ligne et polygone
- L'**indexation spatiale** est utilisée pour améliorer les performances d'exécution des opérations spatiales
- Les **fonctions spatiales**, au sens SQL, sont utilisées pour accéder à des propriétés ou à des relations spatiales
- Utilisés de manière combinée, les types de données spatiales, les index et les fonctions fournissent une structure flexible pour optimiser les performances et les analyses.

# Type de données spatiales

## ➤ Les types de données spatiales

- Représentent les **entités géographiques**.
- Permettent d'accéder à des propriétés de l'entité géographique : contours, dimension.
- Hiérarchie de type avec héritage de la structure et du comportement.

Geometry Hierarchy



# Données spatiales

Éditeur SQL Constructeur graphique de requêtes

Requêtes précédentes  Supprimer Tout supprimer

```
select * from geometry_columns where type <> 'GEOMETRY' order by f_table_schema;
```

Panneau sortie

Sortie de données Expliquer (Explain) Messages Historique

	f_table_catalog character varying(256)	f_table_schema character varying(256)	f_table_name character varying(256)	f_geometry_column character varying(256)	coord_dimension integer	srid integer	type character varying(30)
1	sage molasse	contexte	principale ville	geom	4	2154	POINT
2	sage molasse	contexte	syndicat aep	geom	2	2154	MULTIPOLYGON
3	sage molasse	contexte	commune sage	geom	4	2154	MULTIPOLYGON
4	sage molasse	contexte	commune	geom	4	2154	MULTIPOLYGON
5	sage molasse	contexte	contrat riviere	geom	2	2154	MULTIPOLYGON
6	sage molasse	contexte	perimetre sage	geom	2	2154	MULTIPOLYGON
7	sage molasse	eau	masse eau superficielle	geom	2	2154	MULTILINESTRING
8	sage molasse	eau	alluvion valence	geom	2	2154	MULTIPOLYGON
9	sage molasse	eau	captage prioritaire	geom	2	2154	POINT
10	sage molasse	eau	bac	geom	2	2154	MULTIPOLYGON
11	sage molasse	eau	qualite esu	geom	2	2154	MULTILINESTRING

# La colonne geom

La géométrie est stockée dans une colonne généralement appelée geom

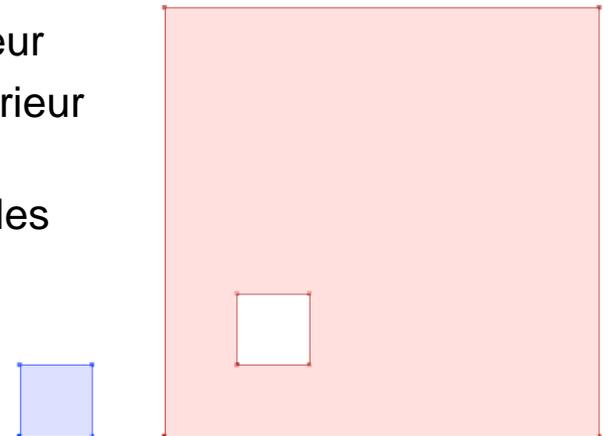
- **Point** : représenté par ses coordonnées X, Y  
SELECT ST\_AsText(geom) FROM geometries WHERE name = 'Point';  
POINT(0 0)  
**ST\_X(geom)** retourne la composante X  
**ST\_Y(geom)** retourne la composante Y
- **Ligne** : chemin entre plusieurs points représenté par un tableau de points  
SELECT ST\_AsText(geom) FROM geometries WHERE name = 'Linestring';  
LINESTRING(0 0, 1 1, 2 1, 2 2)  
**ST\_Length(geometry)** retourne la longueur d'une ligne  
**ST\_StartPoint(geometry)** retourne le premier point d'une ligne  
**ST\_EndPoint(geometry)** retourne le dernier point d'une ligne  
**ST\_NPoints(geometry)** retourne le nombre de points dans une ligne

# La colonne geom

- **Polygone**

POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)) POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),(1 1, 1 2, 2 2, 2 1, 1 1))

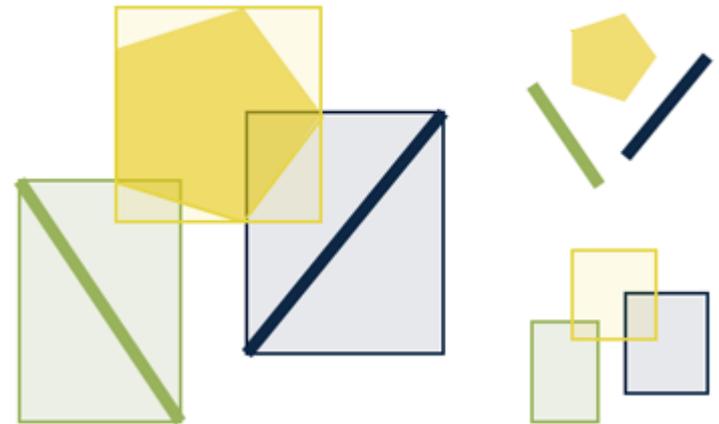
- **ST\_Area(geometry)** retourne l'aire d'un polygone
- **ST\_NRings(geometry)** retourne le nombre de contours (habituellement 1, plus lorsqu'il y a des trous)
- **ST\_ExteriorRing(geometry)** retourne le contour extérieur
- **ST\_InteriorRingN(geometry,n)** retourne le contour intérieur numéro n
- **ST\_Perimeter(geometry)** retourne la longueur de tous les contours



# Index spatiaux et étendue

- Un index est une méthode d'accès à un sous-ensemble de données
- L'indexation des nombres, des chaînes de caractères et des dates est faite sur l'ordre naturel, simple à déterminer et à gérer sous forme d'arbre
- Les bases de données spatiales fournissent un "index spatial" qui répond à la question : "quel objet se trouve dans une étendue spécifique ?"
- Une **étendue** correspond au rectangle de plus petite taille capable de contenir un objet géographique.

Bounding Boxes



# Fonctions spatiales

Pour pouvoir traiter et effectuer des calculs sur les types géométriques, PostGIS fournis de nombreuses fonctions.

Parmi les plus utiles, on peut trouver celles-ci, classées en trois groupes :

- **Récupération** : fonctions qui permettent de *récupérer* les propriétés et les mesures d'une géométrie.
- **Comparaison** : fonctions qui permettent de *comparer* deux géométries en respectant leurs relations spatiales.
- **Construction**: fonctions qui permettent de *construire* de nouvelles géométries à partir d'autres.

La liste des fonctions possibles est très vaste, mais un ensemble commun à l'ensemble des implémentations est défini par la spécification term: *OGC SFSQL*. Cet ensemble commun est implémenté dans PostGIS.

Toutes les fonctions Postgres : <http://docs.postgresql.fr/9.2/fonctions.html>

Toutes les fonctions PostGIS : <http://postgis.net/docs/reference.html>

# Fonctions de récupération

- **ST\_Area**

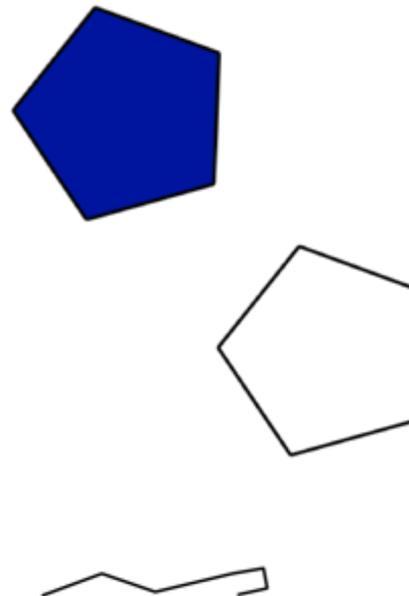
Retourne l'aire d'une géométrie dans l'unité du système de référence spatiale.

- **ST\_Perimeter :**

Retourne le périmètre de la géométrie dans l'unité du système de référence spatiale.

- **ST\_Length :**

Retourne la longueur de la géométrie dans l'unité du système de référence spatiale.



Exemple :

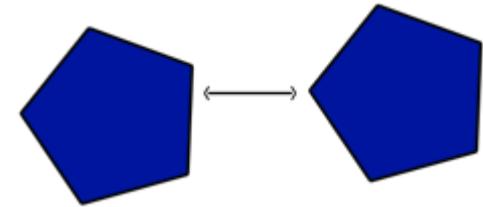
```
SELECT commune, st_area(geom)
FROM contexte.commune
```

commune character varying(150)	st_area double precision
Alixan	28123531.5714136
Allex	20323421.0664053
Ambonil	1172362.68251201
Arthémonay	5699136.14231461
Autichamp	6329293.40046936
Bathernay	5686480.94679473

# Fonctions de comparaison

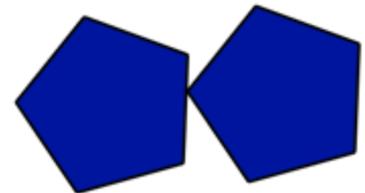
- **ST\_Distance :**

Retourne la distance entre deux géométries dans l'unité du système de référence spatiale.



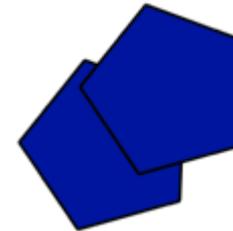
- **ST\_Touches :**

Retourne vrai si les géométries se "touchent spatialement"



- **ST\_Overlaps :**

Retourne vrai si les géométries se chevauchent spatialement.



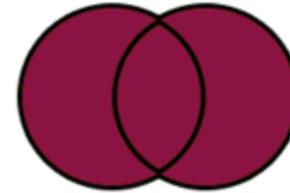
Exemple : `SELECT p1.nom_pe, p2.nom_pe, st_distance(p1.geom, p2.geom)`  
`FROM pression.prelevement_2012 p1, pression.prelevement_2012 p2`  
`WHERE p1.gid < p2.gid`

nom_pe character varying(100)	nom_pe character varying(100)	st_distance double precision
FORAGE LIEU-DIT CAMBEDEAU	SOURCE	47752.7897509664
FORAGE LIEU-DIT CAMBEDEAU	PRISE DANS ISERE LIEU-DIT COUPINIERE	47374.7980259547
FORAGE LIEU-DIT CAMBEDEAU	FORAGE LIEU-DIT CHATELARD	48359.8765300326
FORAGE LIEU-DIT CAMBEDEAU	PRISE DANS VEZY LIEU-DIT BARNETIERE	48522.4249290985
FORAGE LIEU-DIT CAMBEDEAU	PRISE LIEU-DIT GUETTE	49726.5891852639

# Fonctions de construction

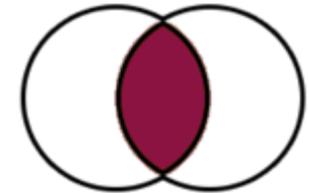
- **ST\_Union :**

Retourne l'union des deux géométries.



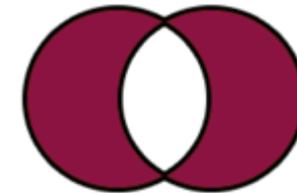
- **ST\_Intersection :**

Retourne la zone géographique (ou une ligne, ou un point) que les deux géométries ont en commun. Si les géométries sont disjointes, la fonction retourne une géométrie vide.



- **ST\_Difference :**

Retourne une géométrie qui représente la partie de la géométrie A qui ne croise pas la géométrie B.



Exemple :

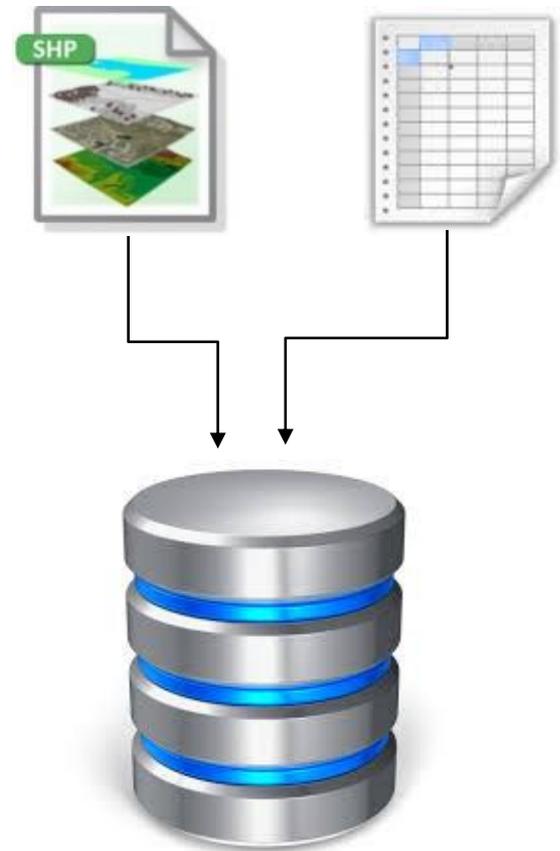
```
SELECT nom_epci, st_union(geom)
FROM contexte.commune_sage
GROUP BY nom_epci
```

nom_epci character varying(250)	st_union geometry
CA Valence-Romans Sud Rhône Alpes	01030000A06A08000001000000770900003461D16680892A41255B317C0A
CC Val de Drôme	01030000A06A080000010000003E03000028CFBC3ED1F729412A380A7436
CC de la Raye	01030000A06A080000010000005E0200007A3001805E622A412A9A99CE6A
CC du Crestois et du pays de Saillans	01060000E06A080000010000001030000C001000000A0100001A9F5035
CC de la Bourne à l'Isère	01030000A06A0800000100000073020000E5642FADC1EF2A4197C910A2E1

# Organisation des données en schéma

## Définition de thèmes

- Contexte
- Socio-économie
- AEP
- Eaux pluviales
- Eaux usées
- Déchets solides
- Raster



# Les outils

- pgAdmin
- Utilitaires de transfert des données
  - shp2pgsql
  - postgis shapefile and dbf loader
  - ogr2ogr
  - ...

Visite guidée de pgAdmin



# Exercices

1. Créer une base de données spatiales
2. Importer des données attributaires et des données spatiales
3. Créer des requêtes et des vues
4. Créer des cartes à partir de la géobase

# La géobase depuis QGIS

- Extension db manager



# Postgis versus fichiers shape

## Inconvénients des fichiers plats :

- **Les fichiers au formats SIG requièrent un logiciel spécifique pour les lire et les écrire.** Le langage SQL est une abstraction de l'accès aléatoire aux données et à leur analyse. Sans cette abstraction, vous devrez développer l'accès et l'analyse par vos propres moyens.
- **L'accès concurrent aux données peut entraîner un stockage de données corrompues.** Il est indispensable d'écrire du code supplémentaire afin de garantir la cohérence des données
- **Les questions compliquées nécessitent des logiciels compliqués pour y répondre.** Les questions intéressantes et compliquées (jointures spatiales, agrégations, etc) qui sont exprimables en une ligne de SQL grâce à la base de données, nécessitent une centaine de lignes de code spécifiques pour y répondre dans le cas de fichiers.
- La plupart des utilisateurs de PostGIS ont mis en place des systèmes où diverses applications sont susceptibles d'accéder aux données, et donc d'avoir les méthodes d'accès SQL standard, qui simplifient le déploiement et le développement.
- Certains utilisateurs travaillent avec de grands jeux de données sous forme de fichiers, qui peuvent être segmentés en plusieurs fichiers, mais dans une base de données ces données peuvent être stockées dans une seule grande table.

