UNIVERSITÉ RENNES 2

Traitements géomatiques avancés

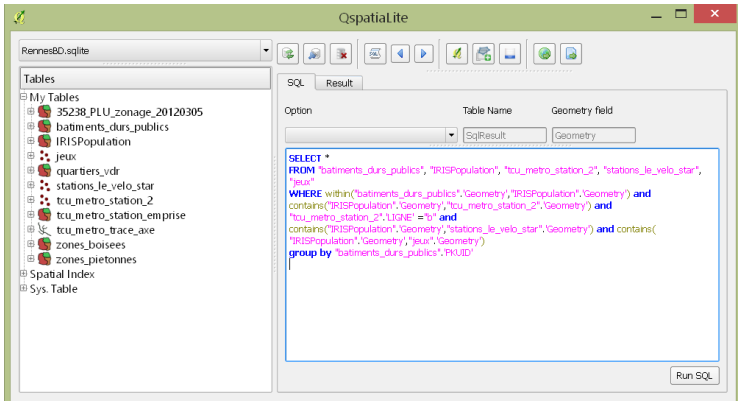
Introduction au SGBD spatial avec QSpatiaLite (SQL spatial)

M2 SIGAT
Automne 2020

@Boris Mericskay

Objectifs séances

- **Premier pas avec un SGBD spatial (QSpatiaLite)**
  - Créer une base de données spatiales
  - Gérer une base de données spatiales
  - Agir sur la structuration des données
  - Effectuer des requêtes attributaires
  - Effectuer des jointures attributaires et spatiales
  - Effectuer des requêtes spatiales
  - Effectuer des requêtes complexes (spatiales et attributaires)



Bases de données et système de gestion de bases de données (SGBD)

Définition

- **Base de données**
  - Ensemble structuré d'éléments d'information, généralement agencés sous forme de tables, dans lesquelles les données sont organisées selon certains critères en vue de permettre leur exploitation
    - Les Bases de données offrent plusieurs avantages sur les données stockées dans un format de fichier simple, comme un shapefile
    - Les avantages incluent des requêtes complexes, des relations complexes, l'évolutivité, la sécurité, et l'intégrité des données, pour n'en nommer que quelques-uns
  - Bref c'est de cette manière que l'on gère vraiment des données !

Définition

Flat File vs. Relational Databases

Flat File Database

A database consisting of a single **Table**

Represented using a **Data Dictionary**

Field Name	Data Type	Data Format	Field Size	Description	Example
License ID	Integer	NNNNNN	6	Unique number ID for all drivers	12345
Surname	Text		20	Surname for Driver	Jones
First Name	Text		20	First Name for Driver	Arnold
Address	Text		50	First Name for Driver	11 Rocky st Como 2233
Phone No.	Text		10	License holders contact number	0400111222
D.O.B	Date / Time	DD/MM/YYYY	10	Drivers Date of Birth	08/05/1956

Contains **files, records, fields** and **characters**

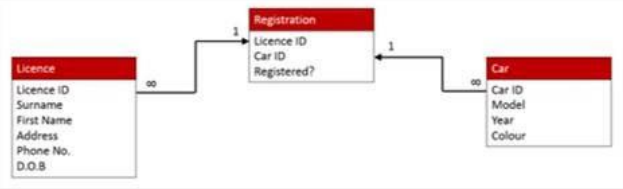
Advantages:
Simple to create, easy to use, inexpensive

Disadvantages:
Increased data redundancy and inconsistency

Relational Database

A database comprised of multiple **Entity's**

Represented Using a **Schema**, such as an **Entity Relationship Diagram**



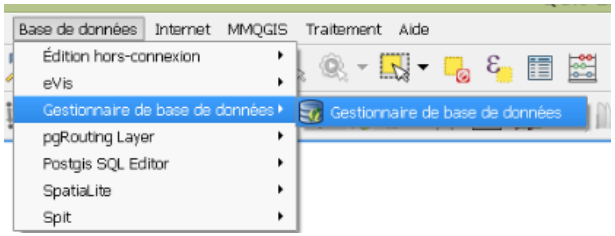
Contains **entity's, attributes** and **relationships**

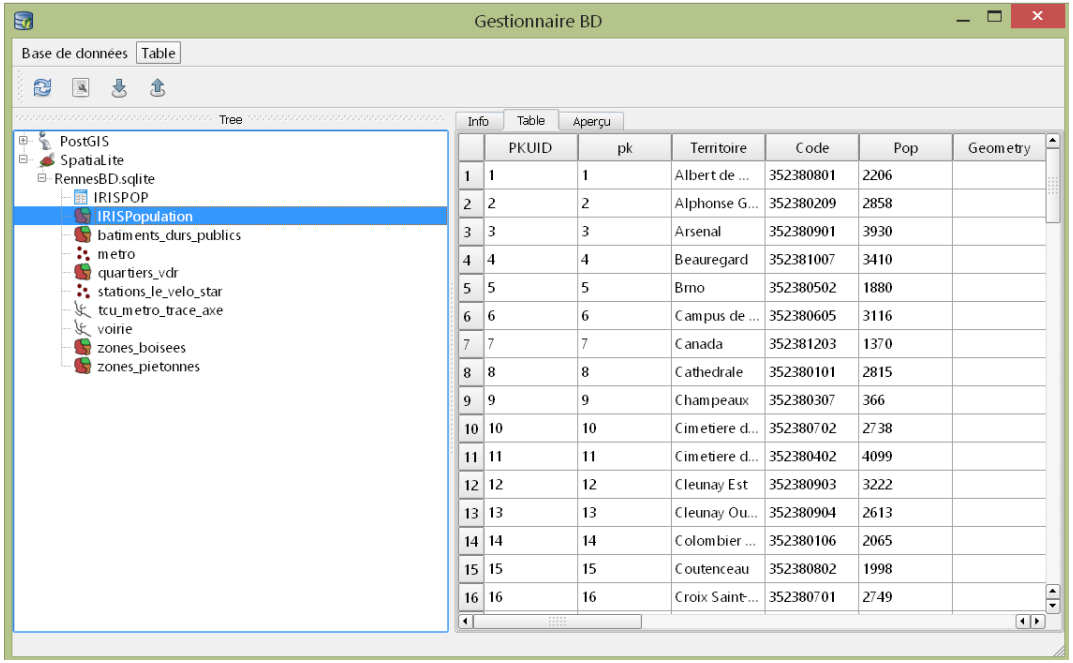
Advantages:
Reduced data redundancy, consistency, shared data, centralised security

Disadvantages:
Takes time to set up

Définition

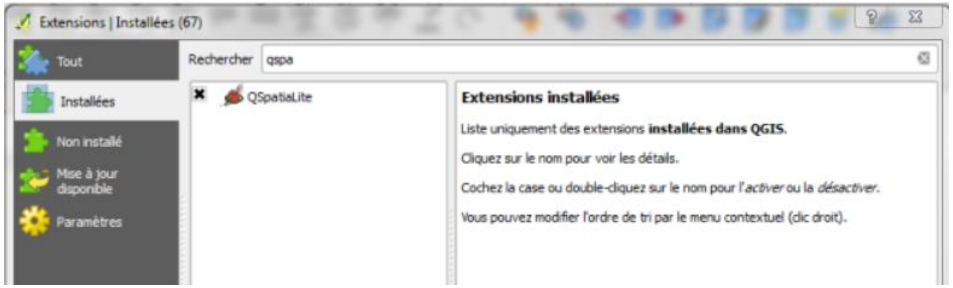
- **Système de gestion de base de donnée (SGBD)**
  - Système qui permet de gérer une BD
    - La BD peut-être personnelle ou partagées (plusieurs utilisateurs)
  - Le SGBD est l'interface de gestion des bases de données





SQLITE et QGIS

- **QSpatiaLite est l'extension spatiale de Sqlite**
  - Permet une gestion optimisée des données
    - Toutes les couches sont centralisée dans un même système
    - Ouvre la porte à la réalisation d'analyses spatiales au-delà des fonctions natives de QGIS
  - QSpatiaLite vs PostGIS = pas multi-utilisateurs
  - QSpatiaLite est disponible de manière native dans QGIS
    - Vérifier que l'extension est bien installée



Ici encore, un choix propre à SQLite:

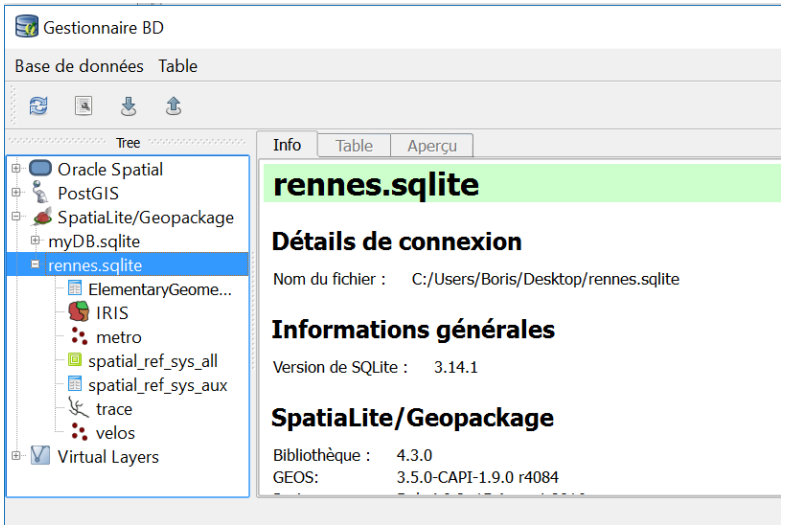
- la suppression de colonne n'est **pas autorisé**.
- renommer les colonnes n'est **pas autorisé**.

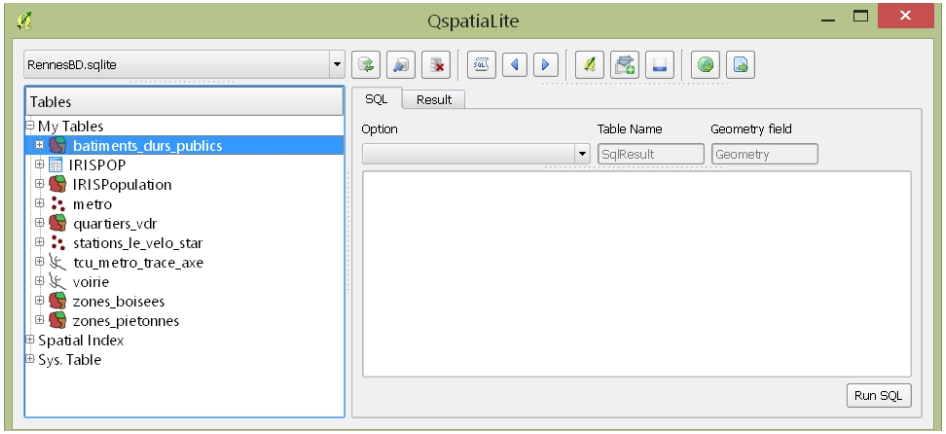
i.e. une fois la colonne créée, vous ne pouvez plus modifier sa configuration.

SQLITE et QGIS

Interface de QSpatiaLite

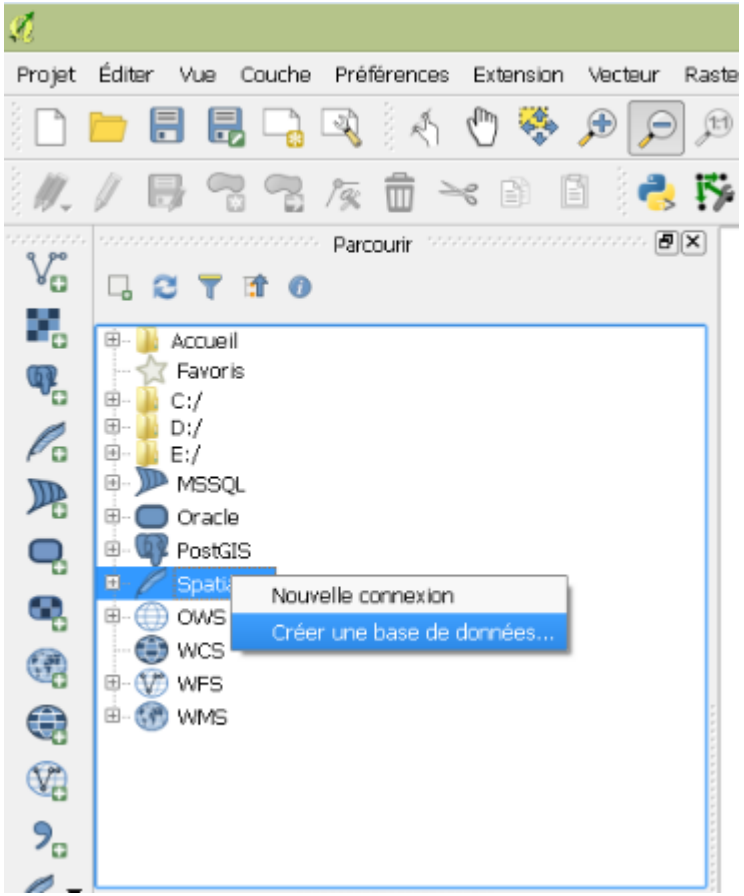
- QSpatiaLite est un SGBD spatial qui permet d'effectuer une multitude de manipulations sur les bases de données
- C'est une déclinaison du gestionnaire BD de QGIS
- **IMPORTANT** : pour bien gérer les données il faut utiliser les deux interfaces de manière complémentaire

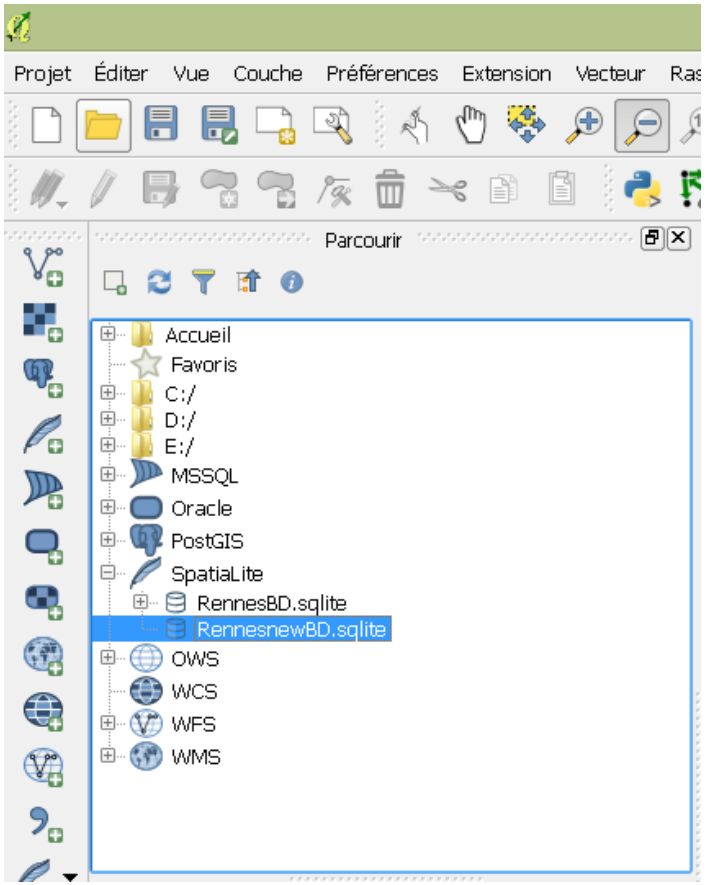




Créer une nouvelle BD spatiale

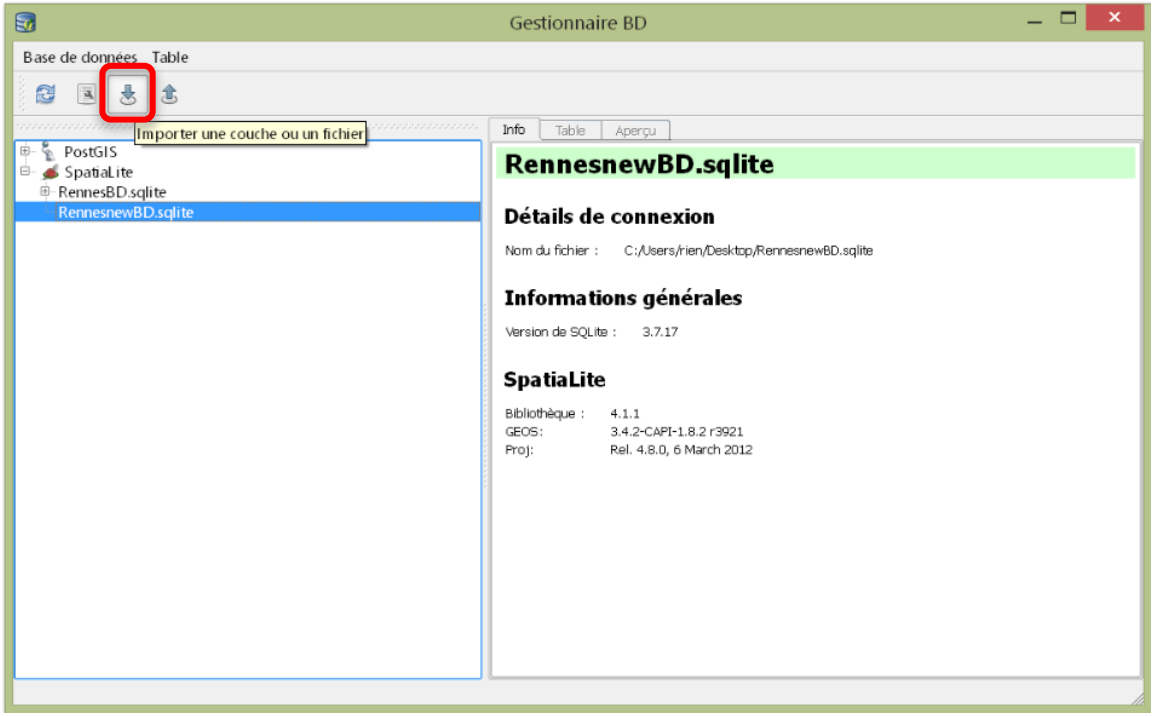
- Créer une nouvelle base de données spatiale





Importer des couches

- **Ajouter des couches dans votre base de données**
  - Cela se passe dans l'interface du gestionnaire BD de QGIS
  - Cliquer sur le bouton pour importer votre première couche



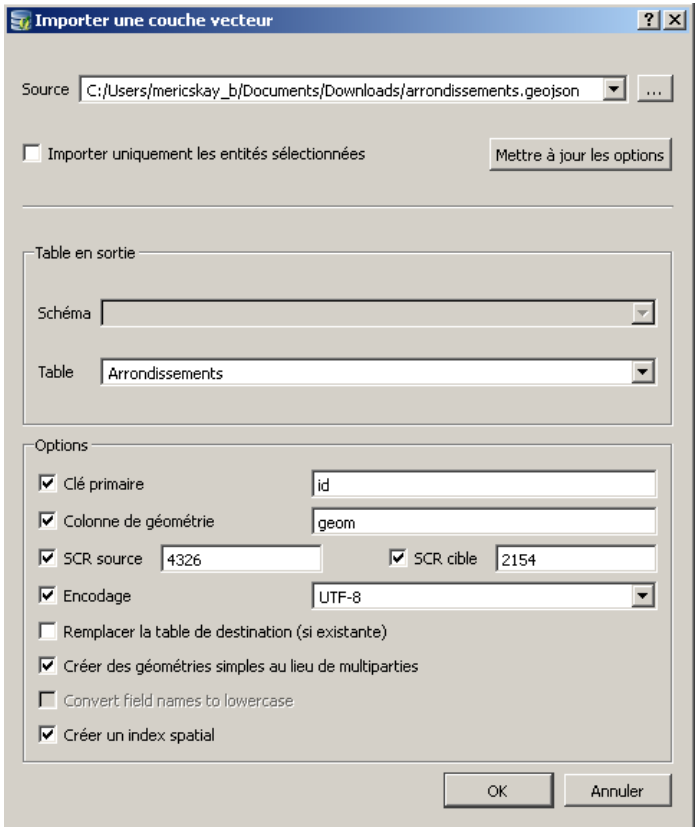
Importer des couches

Ajouter des couches dans base de données

1. **Définissez l'emplacement de votre fichier géographique**
   - Couche chargée dans QGIS
   - Fichier stocké en local
2. **Nommer votre table**
3. **Ajouter une colonne de géométrie**
4. **Définir le SRC source**
5. **Définir le SRC cible**

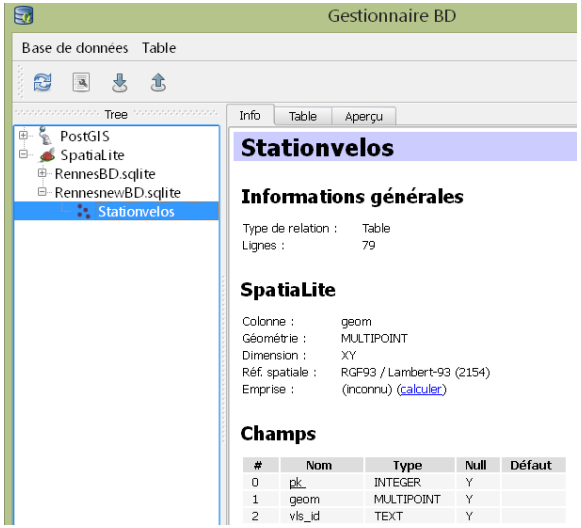
   → Le même pour toutes vos tables
6. **Paramétrer l'encodage (UTF8)**
7. **Créer un index spatial**

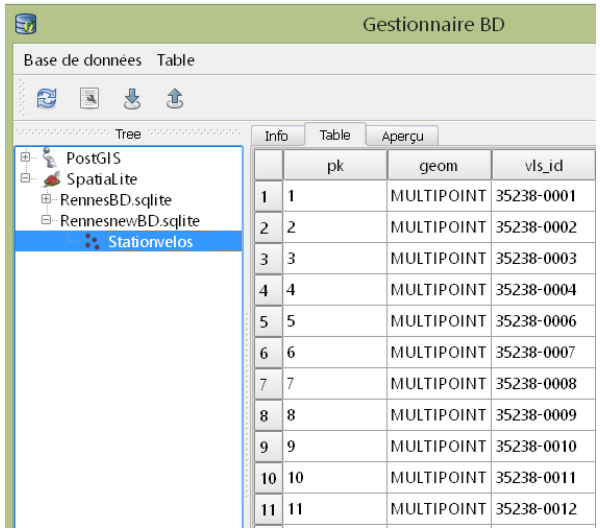
Optimisation des géotraitements

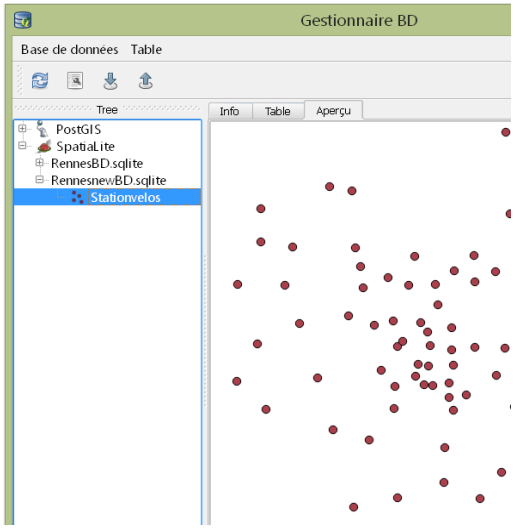


Importer des couches

- **Voilà la première couche de votre base de données spatiale**
  - Dans le gestionnaire BD vous pouvez voir :
    - Informations sur la table
    - Table attributaire
    - Aperçu des entités géographiques

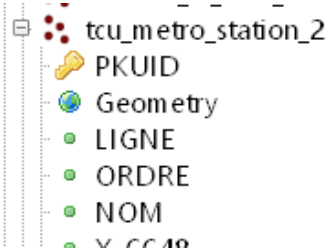






Importer des couches

- **ATTENTION à ne pas utiliser les deux méthodes d'importation pour la même BD**
  - Les deux solutions proposent deux modes de structuration différents de vos données
    - Avec Gestionnaire DB (nomdelacouche.geom)
    - Avec QSpatiaLite (nomdelacouche.geometry)



→ IMPOSSIBLE DE FAIRE DES ANALYSES SPATIALES SI LES GEOMETRIES SONT RENSEIGNEES DE MANIERES DIFFERENTES

Les données

- Communes
- Arrêts de bus
- PLU
- Quartiers
- IRIS
- Stations de métro
- Stations de vélos
- Bâtiments

- Données carroyées INSEE

RENNES METROPOLE EN ACCÈS LIBRE

QSpatiaLite

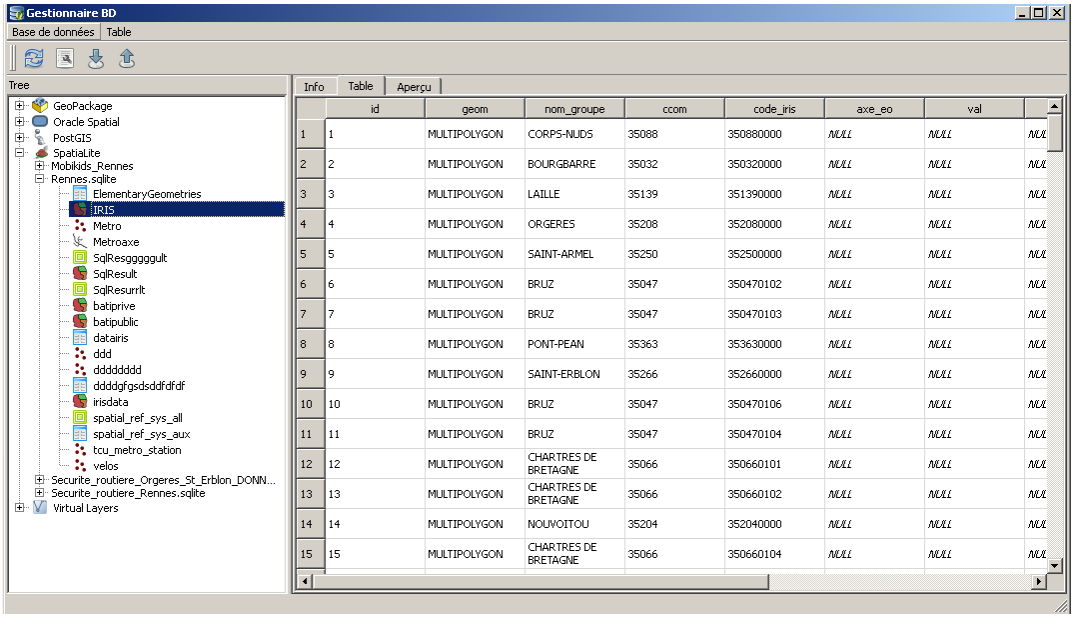
- Le plus grand défi pour l'utilisation de QSpatiaLite est d'apprendre les rudiments de syntaxe de requêtes SQL pour interroger et transformer la base de données
- Documentation Spatialite :

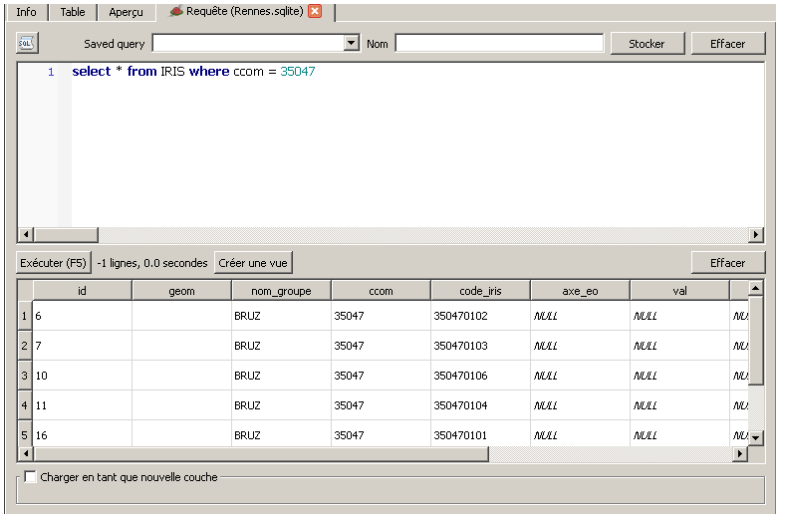
  http://www.gaia-gis.it/spatialite-2.4.0-4/spatialite-sql-2.4-4.html

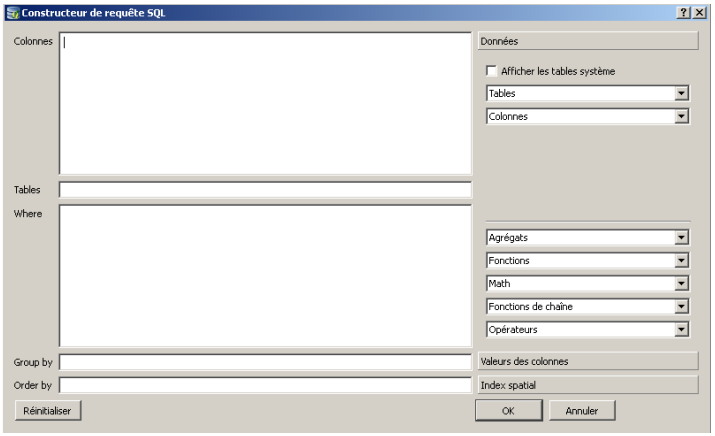
- Afin d'être bon en SQL, il faut y aller doucement et complexifier sa requête au fur et à mesure !
- C'est un langage UNIVERSEL pour gérer et interroger des données (+ de 50 ans…)
- Bref prenez le temps et pratiquez !

DB MANAGER

L’interface de DB Manager







Requêtes SQL

http://cours-fad-public.ensg.eu/mod/imscp/view.php?id=384

Requêtes SQL

La sélection

Syntaxe générale

La requête de sélection est la base de la recherche de données en SQL.

Une requête SQL respecte une syntaxe de type :

SELECT (liste des attributs) **FROM** (liste des tables) **WHERE** (Conditions)

- la partie **SELECT** indique le sous-ensemble des attributs (les colonnes) qui doivent apparaître dans la réponse ;
- la partie **FROM** décrit les relations (les tables) qui sont utilisées dans la requête. Les attributs de la clause SELECT doivent appartenir aux tables listées dans la clause FROM ;
- la partie **WHERE** exprime les conditions, elle est optionnelle.

Nous verrons d'autres options plus tard...

ex 1: `SELECT * FROM commune WHERE population > 1000`

sélectionne les enregistrements de la table COMMUNE dont la population est supérieure à 1000 avec tous les attributs (c'est le sens de *) de la table COMMUNE

ex 2 : `SELECT nom_comm, insee_comm, population FROM commune`

sélectionne tous les enregistrements de la table COMMUNE (cf pas de conditions) et renvoi une table avec les attributs NOM_COM, INSEE_COMM et POPULATION.

Résultat :

	NOM_COMM	INSEE_COMM	POPULATION
1	SAINT-JEAN-DE-LA-MOTTE	72291	900
2	ARTHEZE	72009	400
3	VAULANDRY	49380	300
4	CLEFS	49101	900

Requêtes SQL

Utiliser des alias

Il est possible de donner un nom d'alias aux attributs en sortie avec le mot clef AS.

Ex 3 : `SELECT nom_comm AS COMMUNE , insee_comm AS INSEE, population FROM commune`

on peut également écrire directement (on omet le AS) :

```
SELECT nom_comm COMMUNE , insee_comm INSEE, population FROM commune
```

Résultat :

	COMMUNE	INSEE	POPULATION
1	SAINT-JEAN-DE-LA-MOTTE	72291	900
2	ARTHEZE	72009	400
3	VAULANDRY	49380	300
4	CLEFS	49101	900

Utilisation des alias de nom de colonne

Modifier la structuration des tables

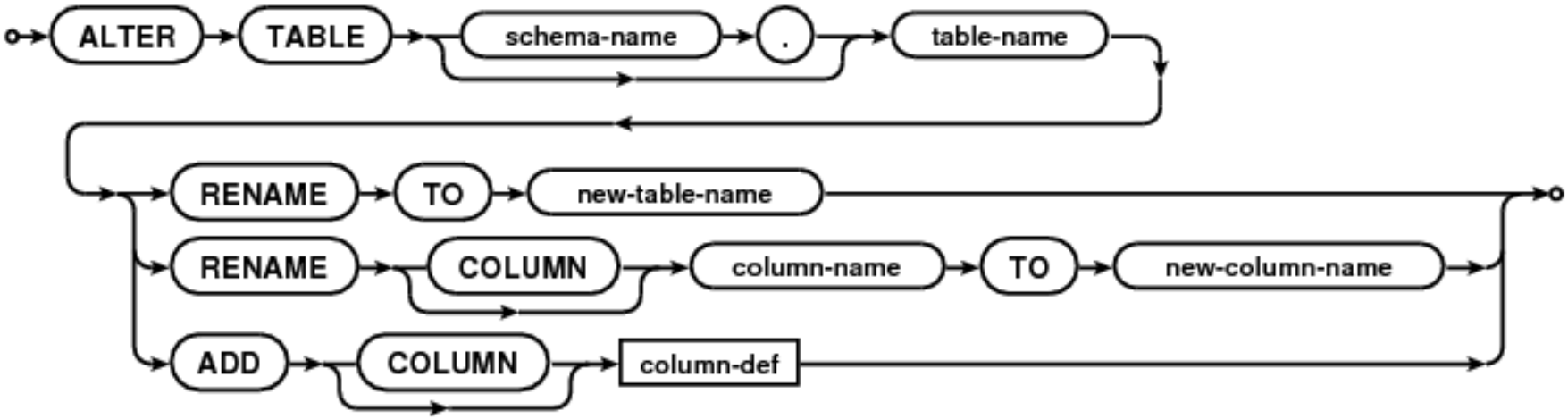
Les types de données et les fonctions

Les types de données

Les principaux types de données en SQL sont :

- CHARACTER (ou **CHAR**) : valeur alpha de longueur fixe ;
- CHARACTER VARYING (ou **VARCHAR**) : valeur alpha de longueur maximale fixée ;
- **TEXT** : suite longue de caractères (sans limite de taille) ;
- **NUMERIC** (ou DECIMAL ou DEC) : décimal ;
- INTEGER (ou **INT**) : entier long ;
- **REAL** : réel à virgule flottante dont la représentation est binaire ;
- **BOOLEAN** (ou LOGICAL) : vrai/faux ;
- **DATE** : date du calendrier grégorien.

Modifier la structuration des tables

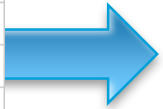


Changer le nom d'une colonne

- **Changer la nom de la colonne avec les nom des communes dans la table IRIS**
  - Passage de *nom_groupe* à *commune*

```
alter table IRIS rename column nom_groupe to commune
```

	id	geom	gml_id	objectid	nom_groupe
1	1	MULTIPOLYGON	iris.15	15.0	CHARTRES DE BRETAGNE
2	2	MULTIPOLYGON	iris.6	6.0	BRUZ
3	3	MULTIPOLYGON	iris.24	24.0	RENNES
4	4	MULTIPOLYGON	iris.25	25.0	CHAVAGNE
5	5	MULTIPOLYGON	iris.13	13.0	CHARTRES DE BRETAGNE



	id	geom	gml_id	objectid	commune
1	1	MULTIPOLYGON	iris.15	15.0	CHARTRES DE BRETAGNE
2	2	MULTIPOLYGON	iris.6	6.0	BRUZ
3	3	MULTIPOLYGON	iris.24	24.0	RENNES
4	4	MULTIPOLYGON	iris.25	25.0	CHAVAGNE
5	5	MULTIPOLYGON	iris.13	13.0	CHARTRES DE BRETAGNE

Changer le nom d'une colonne

- Changer les noms des colonnes relatives aux équipements dans la table des bus

	id	geom	ap_timeo	ap_datemaj	equip_mobil	banc	eclairage	poubelle	PMR
1	*NULL*	POINT	1004	2011-08-24Z	Abri simple	OUI	OUI	NON	NON
2	*NULL*	POINT	1020	2011-08-24Z	Abri simple	OUI	OUI	NON	NON
3	*NULL*	POINT	1024	2011-08-24Z	Abri bâti	NON	NON	NON	NON
4	*NULL*	POINT	1025	2011-08-24Z	Abri simple	OUI	OUI	NON	NON
5	*NULL*	POINT	1026	2011-08-24Z	Abri auvent	NON	NON	NON	NON
6	*NULL*	POINT	1032	2011-08-24Z	Abri simple	OUI	OUI	NON	NON
7	*NULL*	POINT	1035	2011-08-24Z	Abri simple	OUI	OUI	NON	NON
8	*NULL*	POINT	1038	2011-08-24Z	Abri simple	OUI	OUI	NON	NON

Créer et mettre à jour des champs

- Créer un nouveau champ

```
Alter table velos add column nbvelos numeric
```

```
alter table bus add column quartier text
```

- Mettre à jour un champ

Ajouter un nombre ou une chaine de caractère commune

```
update velos set nbvelos='coucou'
```

Ajouter la valeur d'un champ existant

```
update velos set nbvelos=nombreemplacementstheorique
```

Ajouter un champ calculé

```
update table set densite= pop/area
```

Reclassifier un champ

- **Mettre à jour un champ selon des conditions**

Utilisation de la fonction **Case when then…**

> Reclassification selon une colonne

```
update table set champ = case when field_4>30 then 'pasok' when field_4<=30 then 'ok' end
```

- **Reclassifier le champ du nombre d'emplacements de vélos par station en 3 classes**
  - 0-20 = 'peu'
  - 20-30 = 'moyen'
  - 30 = 'beaucoup'

Reclassifier

- **Créer un champ en se basant sur les modalités de plusieurs variables**
  - Créer une nouvelle variable avec 3 modalités pour classifier le type des arrêts de bus
    - Top (abri + tous les équipements)
    - Bof (abri)
    - Nul (tous les autres)

	type	nb
1	bof	636
2	nul	613
3	top	287

119	NULL	POINT	2041	2015-02-09Z	Abri simple	OUI	OUI	OUI	NON	NULL	6 30	Cesson-Sévigné	35051	top
120	NULL	POINT	2046	2012-03-26Z	Abri simple	OUI	OUI	OUI	NON	NULL	64	Cesson-Sévigné	35051	top
121	NULL	POINT	2049	2014-08-11Z	Abri simple	OUI	NON	NON	NON	NULL	30	Cesson-Sévigné	35051	bof
122	NULL	POINT	2052	2015-06-17Z	Abri simple	OUI	OUI	OUI	NON	NULL	34	Cesson-Sévigné	35051	top
123	NULL	POINT	2054	2014-08-11Z	Poteau	NON	NON	NON	NON	NULL	34	Cesson-Sévigné	35051	nul
124	NULL	POINT	2055	2014-08-13Z	Abri simple	OUI	NON	OUI	NON	NULL	34	Cesson-Sévigné	35051	bof
125	NULL	POINT	2057	2012-03-19Z	Abri simple	OUI	OUI	OUI	NON	NULL	35 67 34	Cesson-Sévigné	35051	top

Reclassifier

```
update bus set type = case when equip_mobil like 'Abri%' and equip_banc
= 'OUI' and equip_eclai= 'OUI' and equip_poube ='OUI' then  'top'
when equip_mobil like 'Abri%' then  'bof'
 else 'nul' end
```

Reclassifier un champ

- **Reclassification selon les modalités de plusieurs colonnes**

```
CASE WHEN

conso_totale_tertiaire_mwh > conso_totale_industrie_mwh OR
conso_totale_tertiaire_mwh > conso_totale_professionnel_mwh THEN 'tertiare'

WHEN conso_totale_industrie_mwh > conso_totale_tertiaire_mwh OR
conso_totale_industrie_mwh > conso_totale_professionnel_mwh THEN 'industriel'

WHEN conso_totale_professionnel_mwh > conso_totale_tertiaire_mwh OR
conso_totale_professionnel_mwh > conso_totale_industrie_mwh THEN 'professionel'

END
```

Modifier la structuration des tables

- **Diviser un champ = SUBSTR**
  - Diviser le champ de code commune

```
SELECT nomcommune as Commune, codeinseeco as CodeINSEE, SUBSTR(codeinseeco, 1, 2) as Departement, SUBSTR(codeinseeco, 3) as CodeComm
FROM bus group by nomcommune
```

Commune	CodeINSEE	Departement	CodeComm
Acigné	35001	35	001
Gévezé	35120	35	120
Gévezé	35120	35	120
Gévezé	35120	35	120
Gévezé	35120	35	120
L'Hermitage	35131	35	131
L'Hermitage	35131	35	131
L'Hermitage	35131	35	131

Modifier la structuration des tables

- **A partir de la variable *id_docurba* créer une nouvelle variable avec le code INSEE de la commune**

	id	geom	objectid	id_docurba	libelle	libelong	typezone	destdomi	codecommune
7	7	MULTIPOLYGON	1512.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024
8	8	MULTIPOLYGON	1516.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024
9	9	MULTIPOLYGON	1518.0	0000350242015...	A	Agricole	A	NULL	35024
10	10	MULTIPOLYGON	1521.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024
11	11	MULTIPOLYGON	1527.0	0000350242015...	1AUD2	A Urbaniser alternatif	AUc	NULL	35024
12	12	MULTIPOLYGON	1535.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024
13	13	MULTIPOLYGON	1539.0	0000350242015...	1AUD2	A Urbaniser alternatif	AUc	NULL	35024
14	14	MULTIPOLYGON	1540.0	0000350242015...	Nh1	Naturel constructible (Art...	Nh	NULL	35024
15	15	MULTIPOLYGON	1548.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024
16	16	MULTIPOLYGON	1556.0	0000350242015...	Ng	Naturel constructible (Art...	Nh	NULL	35024
17	17	MULTIPOLYGON	1574.0	0000351962015...	N	Naturel	N	NULL	35196
18	18	MULTIPOLYGON	1588.0	0000351962015...	N	Naturel	N	NULL	35196
19	19	MULTIPOLYGON	1576.0	0000351962015...	N	Naturel	N	NULL	35196

Modifier la structuration des tables

```
alter table PLU add column codecommune text
```

```
update PLU set codecommune = SUBSTR(id_docurba, 5, 5)
```

Modifier la structuration des tables

- **Concaténer deux champs**
  - La fonction CONCAT n'est pas disponible dans SQLITE mais Ok dans PostGIS
  - Alternative utiliser l'opérateur de concaténation || (barre verticale, *tube*, *pipe*) > altgr – 6
  - Concatener le champ *typezone* et *objectid* dans la table PLU

```
SELECT typezone, objectid, id as ID, typezone || '-' || objectid as
Concat FROM PLU
```

	typezone	objectid	ID	Concat
1	Nh	1495.0	1	Nh-1495.0
2	A	1499.0	2	A-1499.0
3	Nh	1500.0	3	Nh-1500.0
4	A	1940.0	4	A-1940.0
5	N	1503.0	5	N-1503.0
6	Nh	1507.0	6	Nh-1507.0

Modifier la structuration des tables

- **Concaténer dans la table PLU la variable *typezone* & *id_docurba* dans une nouvelle colonne**

Info | Table | Aperçu | Requête (M2Base.sqlite) | Requête (M2Base.sqlite)

	id	geom	objectid	id_docurba	libelle	libelong	typezone	destdomi	codecommune	IdUrba
1	1	MULTIPOLYGON	1495.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024	Nh-00003502420151231
2	2	MULTIPOLYGON	1499.0	0000350242015...	A	Agricole	A	NULL	35024	A-00003502420151231
3	3	MULTIPOLYGON	1500.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024	Nh-00003502420151231
4	4	MULTIPOLYGON	1940.0	0000353532015...	Ah	Agricole	A	NULL	35353	A-00003535320151231
5	5	MULTIPOLYGON	1503.0	0000350242015...	N	Naturel	N	NULL	35024	N-00003502420151231
6	6	MULTIPOLYGON	1507.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024	Nh-00003502420151231
7	7	MULTIPOLYGON	1512.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024	Nh-00003502420151231
8	8	MULTIPOLYGON	1516.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024	Nh-00003502420151231
9	9	MULTIPOLYGON	1518.0	0000350242015...	A	Agricole	A	NULL	35024	A-00003502420151231
10	10	MULTIPOLYGON	1521.0	0000350242015...	Nh	Naturel constructible (Art...	Nh	NULL	35024	Nh-00003502420151231

Modifier la structuration des tables

```
alter table PLU add column IdUrba text
```

```
update PLU set  IdUrba =  typezone  || '-' ||  id_docurba
```

Opérateurs logiques

Les opérateurs logiques

- **OR** : pour séparer deux conditions dont au moins une doit être vérifiée.

Ex : `SELECT * FROM commune WHERE statut = 'Commune simple' OR STATUT = 'Chef-lieu de canton'`

Cette requête sélectionne les communes pour lesquelles le statut est "commune simple" ou "chef-lieu de canton".

Bien penser dans l'exemple ci-dessus que le OR lie deux conditions. Une condition contient nécessairement un des opérateurs de comparaison. Ainsi on ne peut écrire

`SELECT * FROM commune WHERE statut = 'Commune simple' OR 'Chef-lieu de canton'`

- **AND** : pour séparer deux conditions qui doivent être vérifiées simultanément.

Ex : `SELECT * FROM commune WHERE statut = 'Sous-préfecture' AND population > 10000`

Seules les sous-préfectures de plus de 10 000 habitants sont sélectionnées.

- **NOT** : permet d'inverser une condition.

Ex : `SELECT * from commune WHERE NOT (statut = 'Commune simple' OR statut = 'Chef-lieu de canton')`

Sélectionne les communes qui ne sont ni commune simple, ni chef lieu de canton.

Opérateurs de comparaison

Les opérateurs de comparaison et les opérateurs logiques

Les opérateurs de comparaison

La clause WHERE est définie par une condition qui s'exprime à l'aide d'opérateurs de comparaison et d'opérateurs logiques.

Les opérateurs de comparaison sont :

- A = B
- A <> B (différent)
- A < B
- A > B
- A <= B (inférieur ou égal)
- A >= B (supérieur ou égal)
- A BETWEEN B AND C (compris entre B et C)
- A IN (B1, B2,...) liste de valeurs

Ex : `SELECT nom_comm, insee_comm, population FROM commune WHERE statut IN('Commune simple', 'Chef-lieu de canton')`

- A LIKE 'chaîne' permet d'insérer des caractères jokers dans l'opération de comparaison, % désignant 0 à plusieurs caractères quelconques , _ désignant un seul caractère.

Sélections attributaires

- **Sélections basiques:**
  - Toutes les stations de vélos avec plus de 20 emplacements (37 stations)

```
SELECT *
FROM velos
WHERE velos.'NB_SOCLES' >20
```

  - Toutes les stations de vélos contenant entre 10 et 20 emplacements (42 stations)

```
SELECT *
FROM velos
WHERE velos.'NB_SOCLES' between 10 and 20
```

Opérateur Distinct

- **Utiliser l'opérateur Distinct**
  - Afficher la liste des noms de communes disponibles de la table des bus

```
select distinct(nomcommune) from bus
```

	nomcommune
1	Rennes
2	*NULL*
3	Cesson-Sévigné
4	Saint-Jacques-...
5	Bruz
6	Betton
7	Le Rheu
8	Saint-Grégoire
9	Chartres-de-Br...
10	Chantepie
11	Pacé

  - Afficher les types d'abri existants

```
select distinct (equip_mobil) from bus
```

	equip_mobil
1	Abri simple
2	Abri bâti
3	Abri auvent
4	Abri double
5	Poteau
6	*NULL*
7	Abri provisoire
8	Station métro
9	Prévoir abri
10	abri vélo
11	Abri mixte vélo

Opérateur Distinct

- **Utiliser l'opérateur Distinct**
  - Afficher le nombre de valeur unique d'une colonne, ici le nombre de communes

```
select count(distinct(nomcommune)) as nbCommunes from bus
```

	nbCommunes
1	43

  - Compter le nombre de combinaisons possibles en croisant la variable type d'arrêts de bus et commune

```
SELECT COUNT(*) as NbCombinaisons
FROM( SELECT DISTINCT equip_mobil, nomcommune
FROM bus )
```

	NbCombinaisons
1	156

Sélections attributaires

- Sélectionner toutes les zones Urbanisées (commencent par U)

```
select * from PLU where libelle like 'U%'
```

- Afficher les libellés de zones du PLU
- Compter le nombre de libellés différents

Sélections attributaires

- **Mobilisation de l'opérateur IN**

Requête avec plusieurs OR

```
SELECT prenom
FROM utilisateur
WHERE prenom = 'Maurice' OR prenom = 'Marie' OR prenom = 'Thimoté'
```

Requête équivalent avec l'opérateur IN

```
SELECT prenom
FROM utilisateur
WHERE prenom IN ( 'Maurice', 'Marie', 'Thimoté' )
```

Sélections attributaires

- **Mobilisation de la clause IN**
  - Sélectionner les arrêts de bus des communes de Rennes, Bruz et Betton

```
SELECT * FROM bus WHERE nomcommune IN ('Rennes', 'Bruz', 'Betton')
```

	id	geom	nom	nomcommune	code	mobilier	estaccessiblepmr	coordonnees	codeinseecommune
1	*NULL*		Longs Champs	Rennes	1001	Abri double	true	(2:48.129077,-1....	35238
2	*NULL*		Mirabeau	Rennes	1005	Abri double	true	(2:48.123081,-1....	35238
3	*NULL*		Roazhon Park	Rennes	9012	*NULL*	false	(2:48.107747,-1....	35238
4	*NULL*		Atalante Cham...	Rennes	1666	*NULL*	false	(2:48.113351,-1....	35238
5	*NULL*		Henri Fréville	Rennes	1659	*NULL*	false	(2:48.086765,-1....	35238
6	*NULL*		Jacques Cartier	Rennes	1651	Poteau provisoire	false	(2:48.097202,-1....	35238
7	*NULL*		Italie	Rennes	1646	Poteau	true	(2:48.086632,-1....	35238
8	*NULL*		Haut Sancé	Rennes	1633	Poteau	true	(2:48.096655,-1....	35238
9	*NULL*		Alma	Rennes	1622	Abri double	true	(2:48.084157,-1....	35238

Sélections attributaires

Mobilisation de la clause IN

- Sélectionner les transactions DVF des communes de Bruz, Betton Pacé et Chantepie

Exécuter | 3144 enregistrements, 0.0 secondes | Créer une vue | Effacer

	id	geom	date	annee	type	prix	surface	prixm2	codecommun	commune	PrixOk
1	NULL	b'\x00\x01j\x08...	24/11/2016	2016	Maison	190000	85	2235	35024	Betton	190000
2	NULL	b'\x00\x01j\x08...	02/10/2015	2015	Maison	194731	98	1987	35024	Betton	194731
3	NULL	b'\x00\x01j\x08...	12/10/2018	2018	Appartement	146000	75	1947	35024	Betton	146000
4	NULL	b'\x00\x01j\x08...	15/01/2016	2016	Maison	270000	93	2903	35024	Betton	270000
5	NULL	b'\x00\x01j\x08...	07/05/2014	2014	Appartement	125000	62	2016	35024	Betton	125000
6	NULL	b'\x00\x01j\x08...	27/05/2019	2019	Maison	250000	88	2841	35024	Betton	250000
7	NULL	b'\x00\x01j\x08...	11/01/2019	2019	Appartement	125000	66	1894	35024	Betton	125000
8	NULL	b'\x00\x01j\x08...	25/02/2014	2014	Maison	370000	144	2569	35024	Betton	370000
9	NULL	b"\x00\x01j\x0...	29/03/2016	2016	Appartement	153000	62	2468	35024	Betton	153000

Sélections attributaires

- **Utiliser la clause NOT**
  - Sélectionner les arrêts de bus ne se trouvant pas dans la commune de Rennes

```
select * from bus where NOT nomcommune='Rennes'
```

  - Sélectionner les arrêts de bus ne se trouvant pas dans la commune de Rennes et n'étant pas de type 'Abri simple'

```
select * from bus where NOT nomcommune='Rennes' and
NOT equip_mobilier = 'Abri simple'
```

Sélections attributaires

- **Sélections avec deux critères attributaires :**
  - Sélectionner les stations de vélos avec plus de 20 emplacements et situées à un arrêt de métro

```
SELECT *
FROM stations_le_velo_star
WHERE stations_le_velo_star.'NB_SOCLES' > 20 and
stations_le_velo_star.'METRO' = oui
```

  - Sélectionner les stations de vélos équipées d'un terminal de paiement électronique (TPE) et possédant entre 10 et 20 socles

```
SELECT *
FROM stations_le_velo_star
WHERE stations_le_velo_star.'TPE' =oui and stations_le_velo_star.'NB_SOCLES'
between 20 and 30
```

Sélections attributaires

- **Sélectionner les arrêts de bus**
  - de type Abri simple et abri auvent,
  - équipés d'un banc et d'un éclairage
  - dans les communes de Rennes, Betton et Bruz

Exécuter | 55 enregistrements, 0.0 secondes | Créer une vue | Effacer

	id	geom	ap_timeo	ap_datemaj	equip_mobil	equip_banc	equip_eclai	equip_poube	access_pmr	access_li_o	access_li_n	nomcommune	codeinseeco
1	NULL	b'\x00\x01j\x08...	1213	2011-12-21Z	Abri simple	OUI	OUI	OUI	NON	NULL	2 1 11	Rennes	35238
2	NULL	b'\x00\x01j\x08...	1570	2016-07-18Z	Abri simple	OUI	OUI	OUI	NON	NULL	6	Rennes	35238
3	NULL	b'\x00\x01j\x08...	2308	2011-08-24Z	Abri simple	OUI	OUI	OUI	NON	NULL	57	Bruz	35047
4	NULL	b'\x00\x01j\x08...	2314	2014-08-11Z	Abri simple	OUI	NON	OUI	NON	NULL	57	Bruz	35047
5	NULL	b'\x00\x01j\x08...	2320	2017-05-11Z	Abri simple	OUI	NON	OUI	NON	NULL	57	Bruz	35047
6	NULL	b'\x00\x01j\x08...	2327	2011-08-24Z	Abri simple	OUI	OUI	OUI	NON	NULL	63 91	Bruz	35047
7	NULL	b'\x00\x01j\x08...	2332	2014-12-08Z	Abri simple	OUI	OUI	OUI	NON	NULL	59	Bruz	35047
8	NULL	b'\x00\x01j\x08...	2404	2011-08-24Z	Abri simple	OUI	OUI	OUI	NON	NULL	51	Betton	35024
9	NULL	b'\x00\x01j\x08...	2406	2011-08-24Z	Abri simple	OUI	OUI	OUI	NON	NULL	51 94 71	Betton	35024
10	NULL	b'\x00\x01j\x08...	2414	2011-08-24Z	Abri simple	OUI	NON	OUI	NON	NULL	51	Betton	35024
11	NULL	b'\x00\x01j\x08...	2420	2011-08-24Z	Abri simple	OUI	NON	OUI	NON	NULL	78 51	Betton	35024

Sélections attributaires

- **Sélectionner les arrêts de bus**
  - de type Abri simple et abri auvent,
  - équipés d'un banc et d'un éclairage
  - dans les communes de Rennes, Betton et Bruz

```
select * from bus where equip_mobil in ('Abri auvent', 'Abri simple')
and equip_banc = 'OUI'
and equip_poube = 'OUI'
and nomcommune in ('Rennes', 'Bruz', 'Betton')
```

Sélections attributaires

- **Sélectionner les arrêts de bus**
  - de type Abri simple et abri auvent,
  - équipés d'un banc et d'un éclairage
  - dans les communes de Rennes, Betton et Bruz

```
select * from bus where equip_mobil in ('Abri auvent', 'Abri simple')
and equip_banc = 'OUI'
and equip_poube = 'OUI'
and nomcommune in ('Rennes', 'Bruz', 'Betton')
```

Sélections attributaires

- **Exercices PLU**
  - Toutes les zones commençant par U
  - Toutes les zones qui ne sont pas de type U
  - Toutes les zones A et N
  - Toutes les zones de type A,U et N
  - Toutes les zones de type AU

- **Exercices bus**
  - Tous les arrêts de bus équipés de tous les équipements à Rennes
  - Tous les arrêts de bus équipés de tous les équipements hors Rennes
  - Tous les arrêts de type « Abri auvent » à Rennes, Betton, Bruz et Chantepie

Requêtes imbriquées

- **Nous voulons utiliser ici la variable PMR mais elle n'est pas renseignée de manière correcte dans la table bus**
- **En revanche elle est bien renseignée dans la couche *Bus2020* (Geojson)**
- **Ajouter cette couche dans votre BD**

Requêtes imbriquées

- **Nous cherchons les arrêts de bus de type « abri auvent » accessible aux PMR en utilisant:**
  - La colonne *equip_mobil* dans la table bus
  - La colonne *estaccessiblepmr* dans la table bus2020
  - Pour faire lien il faut deux colonnes de jointure (?)

```
select * from bus where ap_timeo in (select code from bus2020 where estaccessiblepmr = 'true') and equip_mobil='Abri auvent'
```

Exécuter | 33 enregistrements, 0.0 secondes | Créer une vue | Effacer

	id	geom	ap_timeo	ap_datemaj	equip_mobil	equip_banc	equip_eclai	equip_poube	access_pmr	access_li_o	access_li_n	nomcommune	codeinseeco
1	*NULL*	b'\x00\x01j\x08...	1026	2011-08-24Z	Abri auvent	NON	NON	NON	NON	*NULL*	8 1 9 5	Rennes	35238
2	*NULL*	b'\x00\x01j\x08...	1082	2011-08-24Z	Abri auvent	NON	NON	NON	NON	*NULL*	2	Rennes	35238
3	*NULL*	b'\x00\x01j\x08...	1085	2014-01-07Z	Abri auvent	NON	OUI	NON	NON	*NULL*	1 2 11	Rennes	35238
4	*NULL*	b'\x00\x01j\x08...	1118	2011-08-24Z	Abri auvent	NON	NON	NON	NON	*NULL*	3	Rennes	35238
5	*NULL*	b'\x00\x01j\x08...	1151	2011-08-24Z	Abri auvent	NON	OUI	NON	NON	*NULL*	3	Rennes	35238
6	*NULL*	b'\x00\x01j\x08...	1184	2011-08-24Z	Abri auvent	NON	NON	NON	NON	*NULL*	31	Rennes	35238
7	*NULL*	b'\x00\x01j\x08...	1233	2011-08-24Z	Abri auvent	NON	OUI	NON	NON	*NULL*	11 56 54 55	Rennes	35238
8	*NULL*	b'\x00\x01j\x08...	1264	2011-08-24Z	Abri auvent	NON	OUI	NON	NON	*NULL*	57 5	Rennes	35238
9	*NULL*	b'\x00\x01j\x08...	1370	2011-08-24Z	Abri auvent	NON	NON	NON	NON	*NULL*	8	Rennes	35238

Requêtes imbriquées

- Trouver toutes les ventes maisons (table DVF) dans une commune équipée d'arrêts de bus de type « Abri double » (table bus)

Exécuter | 8751 enregistrements, 0.1 secondes | Créer une vue | Effacer

	id	geom	date	annee	type	prix	surface	prixm2	codecommun	commune	PrixOk
1	NULL	b'\x00\x01j\x08...	24/11/2016	2016	Maison	190000	85	2235	35024	Betton	190000
2	NULL	b'\x00\x01j\x08...	02/10/2015	2015	Maison	194731	98	1987	35024	Betton	194731
3	NULL	b'\x00\x01j\x08...	15/01/2016	2016	Maison	270000	93	2903	35024	Betton	270000
4	NULL	b'\x00\x01j\x08...	27/05/2019	2019	Maison	250000	88	2841	35024	Betton	250000
5	NULL	b'\x00\x01j\x08...	25/02/2014	2014	Maison	370000	144	2569	35024	Betton	370000
6	NULL	b'\x00\x01j\x08...	31/03/2014	2014	Maison	240000	100	2400	35024	Betton	240000
7	NULL	b'\x00\x01j\x08...	10/06/2015	2015	Maison	205000	86	2384	35024	Betton	205000
8	NULL	b'\x00\x01j\x08...	22/06/2015	2015	Maison	169008	86	1965	35024	Betton	169008
9	NULL	b'\x00\x01j\x08...	07/01/2015	2015	Maison	347500	165	2106	35024	Betton	347500
10	NULL	b'\x00\x01j\x08...	15/07/2014	2014	Maison	365000	120	3042	35024	Betton	365000
11	NULL	b'\x00\x01j\x08...	28/08/2015	2015	Maison	285000	110	2591	35024	Betton	285000

Requêtes imbriquées

```
select * from DVF where codecommun in (select codeinseeco from bus
where equip_mobil='Abri double' ) and type ='Maison'
```

Tri

Tri et agrégation

Tri

Il est possible de classer le résultat d'une requête en ajoutant le mot clef **ORDER BY** suivi d'une liste de champs.

Ex : `SELECT * FROM commune ORDER BY nom_comm` pour classer le résultat par nom de commune.

Un tri décroissant peut-être obtenu en ajoutant **DESC**.

Ex :

```
SELECT * FROM commune ORDER BY nom_comm DESC
```

```
SELECT nom_comm, round(cast(population as float)/superficie,2) AS densite FROM commune ORDER BY densite
```

retourne la densité de population par ordre croissant de densité.

Résultat :

	NOM_COMM	densite
1	VAULANDRY	0.11
2	COURCELLES-LA-FORET	0.2
3	THOREE-LES-PINS	0.25
4	SAINT-JEAN-DE-LA-MOTTE	0.28
5	BOUSSE	0.33
6	CLEFS	0.35

Densite de population triée

Tri

- Trier les transactions DVF de la plus chère à la moins chère

```
select * from DVF order by prix desc
```

	id	geom	date	annee	type	prix	surface	prixm2	codecommun	commune
1	NULL	b'\x00\x01j\x08...	08/07/2015	2015	Maison	9e+05	200	4500	35238	Rennes
2	NULL	b'\x00\x01j\x08...	10/05/2019	2019	Maison	9e+05	205	4390	35238	Rennes
3	NULL	b'\x00\x01j\x08...	19/09/2017	2017	Maison	9e+05	223	4036	35238	Rennes
4	NULL	b'\x00\x01j\x08...	28/03/2019	2019	Maison	9e+05	206	4369	35238	Rennes
5	NULL	b'\x00\x01j\x08...	30/03/2018	2018	Appartement	99999	45	2222	35055	Chantepie

Champs

#	Nom	Type	Null	Défaut
0	id	TEXT	Y	
1	geom	MULTIPOINT	Y	
2	date	TEXT	Y	
3	annee	TEXT	Y	
4	type	TEXT	Y	
5	prix	TEXT	Y	
6	surface	TEXT	Y	
7	prixm2	TEXT	Y	
8	codecommun	TEXT	Y	
9	commune	TEXT	Y	
10	colum	PrixOK integer	Y	
11	PrixOK	integer	Y	
12	PrixM2OK	integer	Y	

Problème = les colonnes sont en texte !

Tri

- **Créer une nouvelle colonne avec le prix en *integer* et faite le tri à nouveau**

```
alter table DVF add column PrixOk integer
```

```
update dvf set Prixok = prix
```

```
select * from DVF order by Prixok desc
```

	id	geom	date	annee	type	prix	surface	prixm2	codecommun	commune	PrixOk
1	NULL	b'\x00\x01j\x08...	26/01/2015	2015	Maison	1555000	307	5065	35238	Rennes	1555000
2	NULL	b'\x00\x01j\x08...	10/09/2015	2015	Maison	1422000	273	5209	35238	Rennes	1422000
3	NULL	b'\x00\x01j\x08...	30/06/2014	2014	Maison	1420000	380	3737	35238	Rennes	1420000
4	NULL	b'\x00\x01j\x08...	04/07/2014	2014	Maison	1285000	245	5245	35238	Rennes	1285000
5	NULL	b'\x00\x01j\x08...	21/07/2015	2015	Maison	1270000	359	3538	35238	Rennes	1270000

Tri

- Créer aussi une colonne prixM2 ok et surfaceOK

Info | Table | Aperçu | Requête (M2Base.sqlite)

	id	geom	date	annee	type	prix	surface	prixm2	codecommun	commune	PrixOk	Prixm2Ok	SurfaceOK
1	NULL	MULTIPOINT	09/03/2018	2018	Maison	105000	78	1346	35001	Acigné	105000	1346	78
2	NULL	MULTIPOINT	04/08/2017	2017	Maison	265000	125	2120	35001	Acigné	265000	2120	125
3	NULL	MULTIPOINT	18/03/2016	2016	Maison	14000	14	1000	35001	Acigné	14000	1000	14
4	NULL	MULTIPOINT	03/04/2017	2017	Maison	189000	84	2250	35001	Acigné	189000	2250	84
5	NULL	MULTIPOINT	29/12/2017	2017	Appartement	133107	64	2080	35001	Acigné	133107	2080	64
6	NULL	MULTIPOINT	06/10/2017	2017	Appartement	108953	52	2095	35001	Acigné	108953	2095	52
7	NULL	MULTIPOINT	30/03/2016	2016	Appartement	190000	71	2676	35001	Acigné	190000	2676	71
8	NULL	MULTIPOINT	14/09/2016	2016	Maison	192000	107	1794	35001	Acigné	192000	1794	107
9	NULL	MULTIPOINT	03/01/2014	2014	Maison	210000	70	3000	35001	Acigné	210000	3000	70
10	NULL	MULTIPOINT	10/09/2018	2018	Maison	296000	135	2193	35001	Acigné	296000	2193	135
11	NULL	MULTIPOINT	31/05/2018	2018	Maison	42479	93	457	35001	Acigné	42479	457	93
12	NULL	MULTIPOINT	30/01/2018	2018	Appartement	50000	27	1852	35001	Acigné	50000	1852	27
13	NULL	MULTIPOINT	27/07/2016	2016	Maison	333000	163	2043	35001	Acigné	333000	2043	163
14	NULL	MULTIPOINT	13/07/2017	2017	Appartement	151000	63	2397	35001	Acigné	151000	2397	63

Tri

- **Utiliser la clause LIMIT pour restreindre la sélection à un certain nombre d'entités**
  - Sélectionner les dix stations de vélos avec le plus d'emplacements

```
select * from velos
order by nbvelos asc
limit 10
```

	id	geom	nom	ionmetrocorrespon	code	adressenumero	adressevoie	nomcommune	idstationproche2	eemplacementsthe	codeinseecommune
1	NULL		Place de Bretag...	NULL	5524	3	Place de Bretag...	Rennes	5521	40	35238
2	NULL		Place Hoche	NULL	5504	12	Place Hoche	Rennes	5506	39	35238
3	NULL		Gares - Solférino	15008	5515	4	Boulevard Solfé...	Rennes	5517	35	35238
4	NULL		Beaulieu Chimie	NULL	5547	NULL	Allée de Beaulieu	Rennes	5546	30	35238
5	NULL		Auberge de Jeu...	NULL	5537	83	Rue de Saint-M...	Rennes	5533	30	35238
6	NULL		Beaulieu Restau...	NULL	5548	41	Avenue des But...	Rennes	5581	30	35238
7	NULL		République	15006	5501	19	Quai Lamartine	Rennes	5510	30	35238
8	NULL		TNB	NULL	5512	1	Square de Kergus	Rennes	5516	28	35238
9	NULL		La Poterie	15015	5582	211	Rue de Vern	Rennes	5567	28	35238
10	NULL		Chèques Postaux	NULL	5528	66	Mail François ...	Rennes	5523	28	35238

Tri

- **Sélectionner les 20 transactions DVF les plus chères en 2019**

```
select * from DVF where annee = '2019' order by Prixok desc limit 20
```

Exécuter | 20 enregistrements, 0.0 secondes | Créer une vue | Effacer

	id	geom	date	annee	type	prix	surface	prixm2	codecommun	commune	PrixOk
1	*NULL*	b'\x00\x01j\x08...	10/04/2019	2019	Maison	1250000	240	5208	35238	Rennes	1250000
2	*NULL*	b'\x00\x01j\x08...	25/10/2019	2019	Maison	1104000	357	3092	35047	Bruz	1104000
3	*NULL*	b'\x00\x01j\x08...	07/11/2019	2019	Maison	979080	285	3435	35278	Saint-Grégoire	979080
4	*NULL*	b'\x00\x01j\x08...	29/08/2019	2019	Maison	964000	300	3213	35352	Vern-sur-Seiche	964000
5	*NULL*	b'\x00\x01j\x08...	21/06/2019	2019	Maison	935000	206	4539	35238	Rennes	935000
6	*NULL*	b'\x00\x01j\x08...	15/05/2019	2019	Maison	915000	221	4140	35238	Rennes	915000
7	*NULL*	b'\x00\x01j\x08...	10/05/2019	2019	Maison	9e+05	205	4390	35238	Rennes	900000
8	*NULL*	b'\x00\x01j\x08...	28/03/2019	2019	Maison	9e+05	206	4369	35238	Rennes	900000
9	*NULL*	b'\x00\x01j\x08...	24/07/2019	2019	Maison	890000	210	4238	35238	Rennes	890000
10	*NULL*	b'\x00\x01j\x08...	09/07/2019	2019	Maison	880000	190	4632	35238	Rennes	880000

Agrégation

Agrégation

Agrégations

Une agrégation est une opération qui permet de regrouper les enregistrements de la table en sortie selon des critères et d'obtenir des informations statistiques sur ces regroupements. Il faut utiliser l'expression **GROUP BY** suivi du critère de regroupement.

La clause **GROUP BY** fonctionne de concert avec les fonctions d'agrégation (ici sum()). Les principales fonctions d'agrégation sont :

- **count()** : renvoie le nombre d'enregistrements
- **sum()** : renvoie la somme
- **max()** : maximum
- **min()** : minimum
- **avg()** : moyenne

Statistiques basiques / agrégation

- **Compter des entités (nb de stations de vélos)**

```
select count (*) from velos
```

- **Sortie le min/max(nb de stations de vélos)**

```
select Min(nb_socles), Max(nb_socles) from velos
```

- **Calcul de moyenne et de somme**

```
select avg(nb_socles), sum(nb_socles) from velos
```

Agrégation

- Afficher le nombre de chaque type d'arrêts de bus

```
select equip_mobilier as type, count(*) as NB from bus
group by equip_mobilier
```

	type	NB
1	*NULL*	17
2	Abri auvent	46
3	Abri bâti	4
4	Abri double	120
5	Abri platine	4
6	Abri provisoire	10
7	Abri simple	864
8	Poteau	450
9	Poteau provisoire	7
10	Station métro	29

Sélections attributaires

- Compter le nombre d'arrêts de bus dans les communes de Rennes, Betton, Bruz Chantepie et Cesson

```
select nomcommune as commune, count(*) from bus where
nomcommune IN ('Rennes', 'Bruz', 'Chantepie', 'Betton', 'Cesson-
Sévigné') and access_pmr = 'OUI' group by nomcommune
```

	commune	count(*)
1	Betton	36
2	Bruz	37
3	Cesson-Sévigné	93
4	Chantepie	26
5	Rennes	521

Agrégation

- **Afficher le nombre d'arrêts de bus par commune pour toutes les communes possédant plus de 50 arrêts de bus**
  - Utiliser ici la condition HAVING

```
select nomcommune as Commune, count(*) as NBbus
from bus
group by nomcommune having NBbus>50
order by  NBbus desc
```

	Commune	NBbus
1	Rennes	577
2	Cesson-Sévigné	98
3	Saint-Jacques-...	66
4	*NULL*	59
5	Bruz	54
6	Saint-Grégoire	51

Agrégation

- Afficher le nombre d'arrêts de bus en fonction du type d'équipement et de la commune

```
select nomcommune, equip_mobilier,  count(*) from bus
group by nomcommune, equip_mobilier
```

	nomcommune	equip_mobilier	count(*)
1	*NULL*	*NULL*	10
2	*NULL*	Abri auvent	1
3	*NULL*	Abri platine	1
4	*NULL*	Abri simple	10
5	*NULL*	Poteau	8
6	*NULL*	Station métro	29
7	Acigné	Abri simple	8
8	Acigné	Poteau	6
9	Betton	Abri double	2
10	Betton	Abri simple	25
11	Betton	Poteau	19

Agrégation

- **Compter le nombre de zones en fonction du zonage**

	Zonage	NbZones
1	U	1937
2	N	1626
3	Nh	835
4	A	528
5	AUc	325
6	AUs	155
7	*NULL*	3

Agrégation

- Compter le nombre de zones en fonction du zonage et du libellé, le tout trié par libellé et fréquence

	Zonage	libelle	NbZones
1	*NULL*	non connu	2
2	*NULL*	PSMV	1
3	A	Ah	279
4	A	A	231
5	A	AUL	18
6	AUc	1AUO	23
7	AUc	1AUG	19
8	AUc	1AUD2	17
9	AUc	1AUDo	16
10	AUc	1AUE1	16
11	AUc	1AUE	13

```
select typezone as Zonage, libelle, count(*) as NbZones from PLU
group by typezone, libelle order by Zonage asc, NbZones desc
```

Agrégation

- **Afficher pour chaque commune le nombre d'arrêts de bus par type**

```
select codeinseecommune as Commune, mobilier as Type,
count(*) as NB_bus,
count(case when mobilier ='Abri bâti' then 1 end) as NB_Abri_Bati,
count(case when mobilier ='Abri simple' then 1 end) as
NB_Abri_Simple,
count(case when mobilier ='Poteau' then 1 end) as NB_Poteau
from bus group by codeinseecommune order by commune asc
```

	Commune	Type	NB_bus	NB_Abri_Bati	NB_Abri_Simple	NB_Poteau
1	35001	Abri simple	14	1	8	5
2	35022	Abri simple	6	0	1	4
3	35024	Abri simple	46	0	25	19
4	35032	Abri simple	29	1	9	14
5	35039	Abri simple	13	1	5	7
6	35047	Abri simple	56	0	28	17
7	35051	Abri simple	106	0	62	37
8	35055	Abri simple	37	0	14	18
9	35058	Poteau	2	0	1	1
10	35059	Abri simple	18	2	5	7

Agrégation

- **Afficher pour chaque commune :**
  - Le nombre de transactions au-dessus de 1 millions d'euros
  - Le nombre de transactions au dessus de 5000e le m²
  - Le nombre de transactions au dessus de 300m²

	Commune	NB_transac	NB_Transac1millior	NB_TransacM2chei	Transacgrandesurf
1	Acigné	401	0	0	0
2	Betton	721	0	0	0
3	Bourgbarré	248	0	0	0
4	Bruz	1166	1	0	4
5	Brécé	124	0	0	0
6	Bécherel	48	0	0	0
7	Cesson-Sévigné	866	1	5	1
8	Chantepie	651	0	0	0
9	Chartres-de-Br...	582	0	0	0
10	Chavagne	271	0	0	0
11	Chevaigné	108	0	0	0

Agrégation

```
select commune as Commune,
count(*) as NB_transac,
count(case when PrixOK>1000000 then 1 end) as NB_Transac1million,
count(case when PrixM2OK>5000 then 1 end) as NB_TransacM2cher,
count(case when surfaceOK >300 then 1 end) as NB_Transacgrandesurface
from DVF group by commune order by commune asc
```

Agrégation

- Compter le nombre bâtiments par commune

	code_insee	NbBatiments
1	35238	41738
2	35051	6049
3	35047	5479
4	35024	4877
5	35210	4368
6	35352	3796
7	35278	3569
8	35196	3422
9	35206	3253
10	35001	3171
11	35334	3154

Agrégation

- **Afficher le nombre d'arrêts de bus de type *Poteau* et *Abri simple***

```
select equip_mobilier as type, count(*) as NB
from bus
where equip_mobilier='Poteau' or equip_mobilier ='Abri simple'
group by equip_mobilier
```

	type	NB
1	Abri simple	864
2	Poteau	450

Exercices

- **Faire une table de synthèse par commune de variables issues des données carroyées**

	Commune	NbCarreaux	Population	NbMenage	MedianeNiveauVie
1	AcignÃ©	224	6794.0	2632.4	25393.672091838038
2	Betton	323	11515.5	4666.700000000...	26794.793951147174
3	BourgbarrÃ©	153	3957.5	1394.200000000...	23927.628813793013
4	Bruz	349	16647.5	6959.899999999...	25012.60372416257
5	BrÃ©cÃ©	80	2228.5	786.3999999999...	23654.277489364184
6	Cesson-SÃ©vi...	362	16878.5	7468.499999999...	27991.96739254267
7	Chantepie	153	9927.0	4507.700000000...	25750.327451432673
8	Chartres-de-Br...	95	7263.5	3260.9	24826.010574168642
9	Chavagne	117	3928.5	1566.299999999...	23721.140955759693
10	ChevaignÃ©	94	2225.0	829.0	23848.68770718933

Exercices

- **Calculer le nombre d'arrêts de bus de type Abri double» et Abri simple» par communes**

	Commune	Nbbus	Abri_simple	Abri_double	total_criteres
1	Rennes	593	418	58	476
2	Cesson-Sévigné	99	61	5	66
3	Saint-Jacques-de-la-Lande	68	38	4	42
4	Bruz	55	31	4	35
5	Saint-Grégoire	50	30	4	34
6	Betton	46	25	2	27
7	Pacé	45	21	1	22
8	Noyal-Châtillon-sur-Seiche	36	16	4	20
9	Vern-sur-Seiche	36	20	1	21
10	*NULL*	34	0	0	0
11	Chantepie	34	14	3	17
12	Le Rheu	34	16	4	20

Exercices

- Compter le nombre de transactions par commune
- Compter le nombre de transactions par communes et par type
- Compter le nombre de transactions par communes, par type et par année

	commune	nb
1	Rennes	16924
2	Bruz	1166
3	Saint-Jacques-...	1085
4	Cesson-Sévigné	866
5	Le Rheu	768
6	Betton	721
7	Chantepie	651
8	Pacé	606

	commune	type	nb
1	Acigné	Appartement	167
2	Acigné	Maison	234
3	Betton	Appartement	275
4	Betton	Maison	446
5	Bourgbarré	Appartement	71
6	Bourgbarré	Maison	177
7	Bruz	Appartement	548
8	Bruz	Maison	618

	commune	annee	type	nb
1	Acigné	2014	Appartement	35
2	Acigné	2014	Maison	28
3	Acigné	2015	Appartement	21
4	Acigné	2015	Maison	39
5	Acigné	2016	Appartement	28
6	Acigné	2016	Maison	60
7	Acigné	2017	Appartement	49
8	Acigné	2017	Maison	53
9	Acigné	2018	Appartement	19
10	Acigné	2018	Maison	34

Exercices

```
select  commune, type, annee, count(*) as nb from DVF
group by commune, type
```

```
select  commune, count(*) as nb from DVF
group by commune
```

```
select  commune, annee, type, count(*) as nb from DVF
group by commune,annee, type
```

Exercices

- Réaliser ce tableau récapitulatif

	commune	type	nb	prixmoyen	prixM2Moyen
1	Rennes	Maison	2177	349657.8566835...	3413.733578318...
2	Cesson-Sévigné	Maison	474	365541.2405063...	2915.107594936...
3	Saint-Grégoire	Maison	342	389161.2982456...	2856.602339181...
4	Cesson-Sévigné	Appartement	392	175118.1198979...	2781.834183673...
5	Saint-Grégoire	Appartement	228	175327.9517543...	2678.052631578...
6	Saint-Jacques-...	Maison	322	231757.0559006...	2565.968944099...
7	Rennes	Appartement	14747	143471.1832237...	2526.458398318...
8	Chantepie	Maison	254	262600.8582677...	2474.740157480...
9	Pacé	Maison	397	315872.0302267...	2460.72040302267
10	Thorigné-Fouill...	Maison	347	281021.1268011...	2447.489913544...
11	Vezin-le-Coquet	Maison	154	252033.6558441...	2411.805194805...

Exercices

```
select  commune, type, count(*) as nb, avg(PrixOk) as prixmoyen,
avg(prixm2ok) as prixM2Moyen
from DVF
group by commune, type
order by prixM2Moyen desc
```

Exercices

- **Réaliser ce tableau récapitulatif**

	Commune	type	Nb_Vente	Prix_Min	Prix_Max	Prix_Moyen
1	Acigné	Appartement	167	50000	359000	145073.5868263...
2	Acigné	Maison	234	14000	720000	238015.3290598...
3	Betton	Appartement	275	37000	383850	145468.8909090...
4	Betton	Maison	446	45000	770000	276076.7443946...
5	Bourgbarré	Appartement	71	52000	192000	127685.0985915...
6	Bourgbarré	Maison	177	55000	730000	208114.3785310...
7	Bruz	Appartement	548	34000	347750	119493.8978102...
8	Bruz	Maison	618	28000	1104000	269385.5647249...
9	Brécé	Appartement	12	39000	158350	117854.75
10	Brécé	Maison	112	40000	426000	202845.3303571...
11	Bécherel	Appartement	5	38000	100000	72800.0

Exercices

- Réaliser ce tableau récapitulatif

```
select  commune, type, count(*) as Nb_Vente, min(PrixOK)
as Prix_Min, max(PrixOK) as Prix_Max, avg(PrixOK) as
Prix_Moyen
from DVF
group by commune, type
```

Exercices

- **Réaliser ce tableau récapitulatif**

	type	Nb_Vente	Prix_Min	Prix_Max	Prix_Moyen	PrixM2_Min	PrixM2_Max	PrixM2_Moyen	Surface_moyenne
1	Appartement	20484	10000	887170	141320.4353641...	312	5400	2445.476713532...	59.81834602616...
2	Maison	10925	10500	1555000	271195.4518993...	302	5400	2478.627276887...	111.6040274599...

Exercices

- Réaliser ce tableau récapitulatif

```
select type, count(*) as Nb_Vente, min(PrixOK) as Prix_Min,
max(PrixOK) as Prix_Max, avg(PrixOK) as Prix_Moyen,
min(PrixM2OK) as PrixM2_Min, max(PrixM2OK) as
PrixM2_Max, avg(PrixM2OK) as PrixM2_Moyen,
avg(surfaceok) as Surface_moyenne
from DVF
group by type
```

Exercices

- Réaliser ce tableau récapitulatif

	commune	Diff_PrixM2Moyen
1	Cesson-Sévigné	397.78060046189375
2	Saint-Grégoire	328.1824561403509
3	Rennes	183.59205861498464
4	Pacé	-35.12541254125413
5	Chantepie	-51.39324116743472
6	Betton	-70.47434119278779
7	Thorigné-Fouill...	-73.17281879194631
8	Montgermont	-92.49230769230769
9	Bruz	-127.02744425385934
10	Vezin-le-Coquet	-127.12056737588652
11	Acigné	-203.73316708229427

Exercices

- Réaliser ce tableau récapitulatif

```
select avg(prixm2) from DVF
```

```
select commune, avg(prixm2-2457) as Diff_PrixM2Moyen
from DVF
group by commune
order by Diff_PrixM2Moyen desc
```

Agrégation

- **Afficher pour chaque commune le nombre d'arrêts de bus total, le nombre de bus équipés d'un banc et le nombre de bus non-équipés d'un banc**

```
SELECT nomcommune as Commune, Count(*) as Nbbus, count(case when equip_banc ='OUI' then 1 end) as bancOK, count(case when equip_banc ='NON' then 1 end) as NoBancs
FROM bus
GROUP BY nomcommune
ORDER BY Nbbus DESC
```

	Commune	Nbbus	bancOK	NoBancs
1	Rennes	577	468	103
2	Cesson-Sévigné	98	66	32
3	Saint-Jacques-...	66	43	23
4	*NULL*	59	12	37
5	Bruz	54	35	19
6	Saint-Grégoire	51	34	17
7	Betton	46	27	19
8	Pacé	45	22	23
9	Chantepie	40	17	23
10	Noyal-Châtillo...	36	23	13
11	Vern-sur-Seiche	35	21	14
12	Le Rheu	34	19	15

Exercices

- **Afficher pour chaque bureau de vote le candidat arrivé en tête au 1er tour des élections présidentielles**

	bureau	candidat	pourcentage
1	Centre Social des Longs Prés	MACRON Emm...	38.25
2	Groupe Scolaire Jules Isaac	MACRON Emm...	37.47
3	Groupe Scolaire Joseph Lotte	MACRON Emm...	35.7
4	Groupe Scolaire Jean Rostand	MACRON Emm...	34.96
5	Groupe Scolaire Liberté	MACRON Emm...	34.79
6	Groupe Scolaire Châteaugiron-Landry	MACRON Emm...	34.49
7	Groupe Scolaire Louise Michel	MACRON Emm...	34.16
8	Groupe Scolaire de la Poterie	MACRON Emm...	33.65

- **Quel jeu de données privilégier ?**
  - Dataset1
  - Dataset2

Elections

- Calculer le nombre total de bureaux de vote à Rennes

	Nb_BV
1	133

- Calculer le nombre personnes inscrites à Rennes

	NB_Inscrits
1	351321.0

- Calculer le nombre de personne ayant votées à Rennes

	NB_EMARGEMENTS
1	278403.0

- Calculer le taux de participation à Rennes

	NB_Inscrits	NB_EMARGEMENTS	TxParticipation
1	352317.0	278403.0	79.02059792743466

Elections

```
select count(distinct(NUMERO_LIEU)) from elections
```

```
select sum(NB_INSCRITS) as NB_Inscrits
from (select distinct(NOM_LIEU), NB_INSCRITS from elections)
```

```
select sum(NB_EMARGEMENTS) as NB_EMARGEMENTS
from (select distinct(NOM_LIEU), NB_EMARGEMENTS from elections)
```

```
select sum(NB_INSCRITS) as NB_Inscrits,
sum(NB_EMARGEMENTS) as NB_EMARGEMENTS,
sum(NB_EMARGEMENTS)/ sum(NB_INSCRITS) *100  as TxParticipation
from (select distinct(NOM_LIEU), NB_INSCRITS, NB_EMARGEMENTS from
elections)
```

Elections

- **Produire ce tableau récapitulatif**

	NOM_LIEU	NB_Inscrits	NB_EMARGEMENTS	TxParticipation
1	Centre Social des Longs Prés	4156.0	3600.0	86.62175168431...
2	Groupe Scolaire Duchesse Anne	4250.0	3564.0	83.85882352941...
3	Groupe Scolaire Albert de Mun	11512.0	9626.0	83.61709520500...
4	Groupe Scolaire Sonia Delaunay	4986.0	4156.0	83.3533894905736
5	Cité Internationale Paul Ricoeur	9750.0	8066.0	82.72820512820...
6	Gymnase Commandant Bougouin	4580.0	3760.0	82.09606986899...
7	Groupe Scolaire Joseph Lotte	7500.0	6148.0	81.97333333333...
8	Groupe Scolaire Echange	8578.0	7030.0	81.95383539286...
9	Groupe Scolaire Liberté	8324.0	6792.0	81.59538683325...
10	Groupe Scolaire de l'Ille	14120.0	11512.0	81.52974504249...
11	Groupe Scolaire Louise Michel	10824.0	8806.0	81.35624538063...

Elections

- Produire ce tableau récapitulatif

	NUMERO_LIEU	NB_Inscrits	NB_EMARGEMENT	TxParticipation
1	141.0	996.0	870.0	87.34939759036...
2	452.0	1093.0	953.0	87.19121683440...
3	14.0	2078.0	1800.0	86.62175168431...
4	632.0	1343.0	1161.0	86.4482501861504
5	142.0	1082.0	930.0	85.95194085027...
6	152.0	1244.0	1066.0	85.69131832797...
7	135.0	1092.0	932.0	85.34798534798...
8	151.0	1249.0	1065.0	85.26821457165...
9	455.0	1171.0	995.0	84.97011101622...
10	261.0	1190.0	1010.0	84.87394957983...
11	344.0	945.0	802.0	84.86772486772...

Elections

- Produire ce tableau récapitulatif

	CANDIDAT	nb_voix	Tx_candidat
1	MACRON Emm...	87270.0	31.34664497149...
2	MÉLENCHON J...	70827.0	25.44045861574...
3	FILLON François	45294.0	16.26922123684...
4	HAMON Benoît	37869.0	13.60222411396...
5	LE PEN Marine	18363.0	6.595834096615...
6	DUPONT-AIGN...	6501.0	2.335104147584...
7	POUTOU Philippe	2649.0	0.951498367474...
8	ASSELINEAU Fr...	1860.0	0.668096248962...
9	ARTHAUD Nath...	1503.0	0.539864872145...
10	LASSALLE Jean	1353.0	0.485986142390...
11	CHEMINADE Ja...	447.0	0.160558614670...

Elections

```
select CANDIDAT,
sum(NB_VOIX) as nb_voix,
sum(NB_VOIX) / 278403 *100 as Tx_candidat
from elections
group by candidat
order by nb_voix desc
```

Elections

- Afficher le candidat arrivé en tête dans chaque lieu de vote
- Afficher le candidat arrivé en tête dans chaque bureau de vote
- Afficher le candidat arrivé en dernier dans chaque lieu de vote
- Afficher le candidat arrivé en dernier dans chaque bureau de vote

Elections

```
select  NOM_LIEU, CANDIDAT, max(NB_VOIX) from elections
group by NOM_LIEU
```

```
select  NOM_LIEU, NUMERO_LIEU, CANDIDAT, max(NB_VOIX)
from elections group by NUMERO_LIEU
```

```
select  NOM_LIEU, CANDIDAT, min(NB_VOIX) from elections
group by NOM_LIEU
```

```
select  NOM_LIEU, NUMERO_LIEU, CANDIDAT, min(NB_VOIX)
from elections group by NUMERO_LIEU
```

Elections

- **Produire ce tableau final**

	NUMERO_LIEU	Gagnant	Nb_Voix	Pourcentage	TxParticipation
1	151.0	FILLON François	477.0	45.26	85.26821457165...
2	113.0	FILLON François	307.0	41.94	83.97727272727...
3	152.0	FILLON François	425.0	40.25	85.69131832797...
4	142.0	MACRON Emm...	352.0	38.43	85.95194085027...
5	14.0	MACRON Emm...	679.0	38.25	86.62175168431...
6	141.0	MACRON Emm...	327.0	38.07	87.34939759036...
7	111.0	MÉLENCHON J...	304.0	37.67	76.36195752539...
8	131.0	MACRON Emm...	290.0	37.47	67.58620689655...
9	314.0	MACRON Emm...	342.0	37.38	82.48239436619...
10	414.0	MACRON Emm...	300.0	37.31	80.19801980198...
11	616.0	MACRON Emm...	319.0	37.14	83.34946757018...

Elections

- Produire ce tableau final

```
select  NUMERO_LIEU, CANDIDAT as Gagnant,
max(NB_VOIX) as Nb_Voix,
POURCENTAGE as Pourcentage,
sum(NB_EMARGEMENTS)/ sum(NB_INSCRITS) *100  as TxParticipation
from elections
group by NUMERO_LIEU
order by POURCENTAGE desc
```

Jointure attributaire

Jointure attributaire

Effectuer des jointures attributaires

- Nous voulons ajouter à la table des IRIS, la population (issues d'un tableur)
- Dans le gestionnaire de bases de données, importer le fichier contenant les informations (IRISRM.csv)

	id	code_iris	nmiris	population2016
1	1	350660104	Chartres de Bre...	671.90593035408
2	2	350470102	Bruz centre sud...	2135.38771704236
3	3	352381108	Portugal	1654.87321917705
4	4	350760000	Chavagne	3909
5	5	350660102	Chartres de Bre...	4281.00681868078
6	6	352040000	Nouvoitou	3016
7	7	350660103	Chartres de Bre...	109.214764555056
8	8	353520102	Vern sur seiche ...	3342.21191227929
9	9	353520101	Vern sur seiche ...	2138.70982153183
10	10	352381004	Villejean Nord-...	2253.81988434717
11	11	350550103	Chantepie sud ...	1643.69756364803

Jointure attributaire

- Le problème est que la colonne Population est formatée comme du texte (issue d'un CSV)
- Créer une nouvelle colonne numérique pour cette variable

	id	code_iris	nmiris	population2016	Pop2016
1	1	350660104	Chartres de Bre...	671.90593035408	671.90593035408
2	2	350470102	Bruz centre sud...	2135.38771704236	2135.38771704236
3	3	352381108	Portugal	1654.87321917705	1654.87321917705
4	4	350760000	Chavagne	3909	3909
5	5	350660102	Chartres de Bre...	4281.00681868078	4281.00681868078
6	6	352040000	Nouvoitou	3016	3016
7	7	350660103	Chartres de Bre...	109.214764555056	109.214764555056
8	8	353520102	Vern sur seiche ...	3342.21191227929	3342.21191227929

Informations générales

Type de relation : Table
Lignes : 172

Champs

#	Nom	Type	Null	Défaut
0	id	INTEGER	Y	
1	code_iris	TEXT	Y	
2	nmiris	TEXT	Y	
3	population2016	TEXT	Y	
4	Pop2016	integer	Y	

Jointure attributaire

- Retourner dans Qspatialite pour effectuer la jointure attributaire entre la table *IRIS* et le tableur *IRISdata*
- Exécuter la requête suivante

```
SELECT *
FROM IRIS, IRISDATA
WHERE IRIS.code_iris== IRISDATA.code_iris
```

	nusectmorp	nmsectmorp	nmiris	niris	st_area_sha	st_length_s	st_area_sha_1	st_length_s_1	nbDVF	NbBus	id	code_iris	nmiris	population2016	Pop2016
1	*NULL*	*NULL*	Chartres de Bre...	CHARTRES DE B...	3566339.70019531	0	3566339.70019531	0	*NULL*	52	1	350660104	Chartres de Bre...	671.90593035408	671.90593035408
2	*NULL*	*NULL*	Bruz centre sud...	BRUZ CENTRE S...	2814091.15234375	0	2814091.15234375	0	*NULL*	54	2	350470102	Bruz centre sud...	2135.38771704236	2135.38771704236
3	11-2	Italie	Portugal	PORTUGAL	785003.67675781	0	785003.67675781	0	*NULL*	47	3	352381108	Portugal	1654.87321917705	1654.87321917705
4	*NULL*	*NULL*	Chavagne	CHAVAGNE	12568637.0927734	0	12568637.0927734	0	*NULL*	58	4	350760000	Chavagne	3909	3909
5	*NULL*	*NULL*	Chartres de Bre...	CHARTRES DE B...	1380925.89355469	0	1380925.89355469	0	*NULL*	60	5	350660102	Chartres de Bre...	4281.00681868078	4281.00681868078
6	*NULL*	*NULL*	Nouvoitou	NOUVOITOU	19417922.6445313	0	19417922.6445313	0	*NULL*	60	6	352040000	Nouvoitou	3016	3016
7	*NULL*	*NULL*	Chartres de Bre...	CHARTRES DE B...	3944962.81152344	0	3944962.81152344	0	*NULL*	50	7	350660103	Chartres de Bre...	109.214764555056	109.214764555056
8	*NULL*	*NULL*	Vern sur seiche ...	VERN SUR SEIC...	7847638.67773438	0	7847638.67773438	0	*NULL*	62	8	353520102	Vern sur seiche ...	3342.21191227929	3342.21191227929
9	*NULL*	*NULL*	Vern sur seiche ...	VERN SUR SEIC...	6820675.23828125	0	6820675.23828125	0	*NULL*	58	9	353520101	Vern sur seiche ...	2138.70982153183	2138.70982153183
10	10-1	Villejean	Villejean Nord-	VILLEJEAN NOR...	216930.71679688	0	216930.71679688	0	*NULL*	48	10	352381004	Villejean Nord-	2253.81988434717	2253.81988434717

Jointure attributaire

- Autre possibilité ne garder que ce qui nous intéresse

```
SELECT IRIS.nmiris, IRIS.nom_groupe as Commune , IRISDATA.Pop2016 as Population
FROM IRIS
JOIN IRISDATA ON IRIS.code_iris = IRISDATA.code_iris
```

	nmiris	Commune	Population
1	Chartres de Bret. rural est	CHARTRES DE BRETAGNE	671.90593035408
2	Bruz centre sud est	BRUZ	2135.387717042...
3	Portugal	RENNES	1654.873219177...
4	Chavagne	CHAVAGNE	3909
5	Chartres de Bret. centre ouest	CHARTRES DE BRETAGNE	4281.006818680...
6	Nouvoitou	NOUVOITOU	3016
7	Chartres de Bret. rural ouest/ la ja...	CHARTRES DE BRETAGNE	109.2147645550...
8	Vern sur seiche ouest	VERN SUR SEICHE	3342.211912279...
9	Vern sur seiche nord	VERN SUR SEICHE	2138.709821531...

Jointure attributaire

- **Créer ce tableau récapitulatif**
  - Il faut ajouter les fonctions de regroupement statistiques adéquates
  - et un group by au bon endroit

	Commune	Nb_IRIS	PopulationCommune
1	RENNES	92	216267.9999999999
2	BRUZ	6	18094.0
3	CESSON-SEVIGNE	9	17371.0
4	SAINT JACQUES DE LA LANDE	3	12917.0
5	PACE	4	11764.0
6	BETTON	4	11222.0
7	CHANTEPIE	3	10378.999999999989
8	SAINT GREGOIRE	3	9521.00000000002
9	LE RHEU	3	8571
10	THORIGNE-FOUILLARD	3	8425.0

Jointure attributaire

```
SELECT IRIS.nom_groupe as Commune , count(IRIS.nmiris) as
Nb_IRIS, sum(IRISDATA.Pop2016) as PopulationCommune
FROM IRIS
JOIN IRISDATA ON IRIS.code_iris = IRISDATA.code_iris
group by  IRIS.nom_groupe
order by PopulationCommune desc
```

Jointure attributaire

- **Ajouter à la table IRIS la population légale de 2016 dans une nouvelle colonne**

nmiris	niris	st_area_sha	st_length_s	st_area_sha_1	st_length_s_1	nbDVF	NbBus	Population
Henri Fréville S...	HENRI FREVILL...	161779.06152344	0	161779.06152344	0	*NULL*	49	1862.57890931588
Emmanuel\Mo...	EMMANUEL M...	157444.81640625	0	157444.81640625	0	*NULL*	51	2207.49993040689
Le Gallet-\Les L...	LE GALLET LES ...	692089.35644531	0	692089.35644531	0	*NULL*	57	3390.95841578855
Le Gast Ouest	LE GAST OUEST	207279.72167969	0	207279.72167969	0	*NULL*	49	1507.00723707811
Le Gast Est	LE GAST EST	302973.26269531	0	302973.26269531	0	*NULL*	55	3077.75043570067
Vezin-le-Coquet	VEZIN-LE-COQ...	7920703.24902344	0	7920703.24902344	0	*NULL*	69	5650
Cesson-Sévign...	CESSON-SEVIG...	4615547.23339844	0	4615547.23339844	0	*NULL*	49	218.476249762976
Morbihan Est	MORBIHAN EST	345993.00195313	0	345993.00195313	0	*NULL*	54	2049.18423630737
Saint-Armel	SAINT-ARMEL	7864090.02734375	0	7864090.02734375	0	*NULL*	54	1869
Pont-Péan	PONT-PEAN	8773513.51855469	0	8773513.51855469	0	*NULL*	63	4225.00000000001
Bruz ouest	BRUZ OUEST	6075184.76269531	0	6075184.76269531	0	*NULL*	64	6809.93128677768

Jointure attributaire

- Ajouter à la table IRIS la population légaléde 2016 dans une nouvelle colonne

```
alter table IRIS add column Population integer
```

```
update IRIS set Population = (SELECT   Pop2016 FROM IRISDATA
WHERE IRIS.code_iris== IRISDATA.code_iris)
```

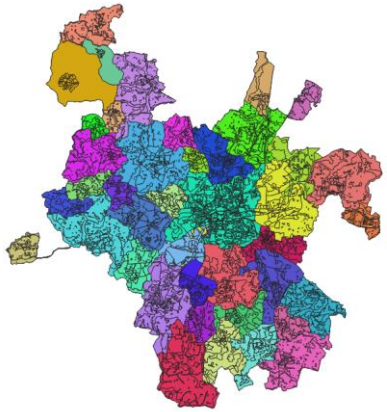
Jointure attributaire

- **Ajouter à la table PLU le nom des communes**

  - Ajouter une colonne codecommune à la table (PLU)

	id	geom	objectid	id_docurba	libelle	libelong	typezone	destdomi	codecommune
1	1	MULTIPOLYGON	3662.0	0000353152015...	UE1	Urbanisé	U	*NULL*	35315
2	2	MULTIPOLYGON	3668.0	0000353152015...	UE2	Urbanisé	U	*NULL*	35315
3	3	MULTIPOLYGON	3727.0	0000353512015...	Ah	Agricole	A	*NULL*	35351
4	4	MULTIPOLYGON	3743.0	0000353512015...	Nh	Naturel constru...	Nh	*NULL*	35351
5	5	MULTIPOLYGON	3776.0	0000353522015...	Nh1	Naturel constru...	Nh	*NULL*	35352
6	6	MULTIPOLYGON	3900.0	0000353522015...	Nh	Naturel constru...	Nh	*NULL*	35352
7	7	MULTIPOLYGON	3873.0	0000353522015...	UE2	Urbanisé	U	*NULL*	35352
8	8	MULTIPOLYGON	3903.0	0000353522015...	Nh	Naturel constru...	Nh	*NULL*	35352
9	9	MULTIPOLYGON	3888.0	0000353522015...	A	Agricole	A	*NULL*	35352
10	10	MULTIPOLYGON	3965.0	0000352382015...	NEh	Naturel constru...	Nh	*NULL*	35238
11	11	MULTIPOLYGON	1619.0	0000350242015...	UI1	Urbanisé	U	*NULL*	35024

Jointure attributaire



- **Ajouter à la table PLU le nom des communes**

  - Faire la jointure avec la table des communes

	id	geom	objectid	id_docurba	libelle	libelong	typezone	destdomi	codecommune	nomcommune
1	1	MULTIPOLYGON	3662.0	0000353152015...	UE1	Urbanisé	U	*NULL*	35315	Saint-Sulpice-la...
2	2	MULTIPOLYGON	3668.0	0000353152015...	UE2	Urbanisé	U	*NULL*	35315	Saint-Sulpice-la...
3	3	MULTIPOLYGON	3727.0	0000353512015...	Ah	Agricole	A	*NULL*	35351	le Verger
4	4	MULTIPOLYGON	3743.0	0000353512015...	Nh	Naturel constru...	Nh	*NULL*	35351	le Verger
5	5	MULTIPOLYGON	3776.0	0000353522015...	Nh1	Naturel constru...	Nh	*NULL*	35352	Vern-sur-Seiche
6	6	MULTIPOLYGON	3900.0	0000353522015...	Nh	Naturel constru...	Nh	*NULL*	35352	Vern-sur-Seiche
7	7	MULTIPOLYGON	3873.0	0000353522015...	UE2	Urbanisé	U	*NULL*	35352	Vern-sur-Seiche
8	8	MULTIPOLYGON	3903.0	0000353522015...	Nh	Naturel constru...	Nh	*NULL*	35352	Vern-sur-Seiche
9	9	MULTIPOLYGON	3888.0	0000353522015...	A	Agricole	A	*NULL*	35352	Vern-sur-Seiche
10	10	MULTIPOLYGON	3965.0	0000352382015...	NEh	Naturel constru...	Nh	*NULL*	35238	Rennes
11	11	MULTIPOLYGON	1619.0	0000350242015...	UI1	Urbanisé	U	*NULL*	35024	Betton
12	12	MULTIPOLYGON	1880.0	0000350882015...	N	Naturel	N	*NULL*	35088	Corps-Nuds
13	13	MULTIPOLYGON	1924.0	0000350882015...	N	Naturel	N	*NULL*	35088	Corps-Nuds
14	14	MULTIPOLYGON	1932.0	0000353532015...	Ah	Agricole	A	*NULL*	35353	Vezin-le-Coquet

Jointure attributaire

```
alter table PLU add column codecommune text
```

```
update PLU set nomcommune = (select nom from communes where
communes.code_insee == PLU.codecommune)
```

Jointure attributaire

- Ajouter à la couche BUS la variable PMR récente (issue de la table Bus2020)
- D'abord en mode vue (select)

	ap_timeo	Mobilier	Banc	eclairage	poubelle	Commune	CodeINSEE	PMR
1	1004	Abri simple	OUI	OUI	NON	Rennes	35238	true
2	1020	Abri simple	OUI	OUI	NON	Rennes	35238	true
3	1024	Abri bâti	NON	NON	NON	Rennes	35238	true
4	1025	Abri simple	OUI	OUI	NON	Rennes	35238	false
5	1026	Abri auvent	NON	NON	NON	Rennes	35238	true
6	1032	Abri simple	OUI	OUI	NON	Rennes	35238	true
7	1035	Abri simple	OUI	OUI	NON	Rennes	35238	true
8	1038	Abri simple	OUI	OUI	NON	Rennes	35238	true
9	1039	Abri simple	OUI	OUI	NON	Rennes	35238	true
10	1047	Abri simple	OUI	OUI	NON	Rennes	35238	true

Jointure attributaire

```
SELECT bus.ap_timeo,
bus.equip_mobil as Mobilier,
bus.equip_banc as Banc,
bus.equip_eclai as eclairage,
bus.equip_poube as poubelle,
bus.nomcommune as Commune,
bus.codeinseeco as CodeINSEE,
bus2020.estaccessiblepmr as PMR
FROM bus
JOIN bus2020 ON bus.ap_timeo = bus2020.code
```

Jointure attributaire

- Ajouter à la couche BUS la variable PMR récente (issue de la table Bus2020)
- Ensuite avec une nouvelle colonne dans la table bus

	id	geom	ap_timeo	ap_datemaj	equip_mobil	equip_banc	equip_eclai	equip_poube	access_pmr	access_li_o	access_li_n	nomcommune	codeinseeco	PMR
1	*NULL*	MULTIPOINT	1004	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	1 50	Rennes	35238	true
2	*NULL*	MULTIPOINT	1020	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	57	Rennes	35238	true
3	*NULL*	MULTIPOINT	1024	2011-08-24Z	Abri bâti	NON	NON	NON	NON	*NULL*	1 5 9	Rennes	35238	true
4	*NULL*	MULTIPOINT	1025	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	5 1 9 8	Rennes	35238	false
5	*NULL*	MULTIPOINT	1026	2011-08-24Z	Abri auvent	NON	NON	NON	NON	*NULL*	8 1 9 5	Rennes	35238	true
6	*NULL*	MULTIPOINT	1032	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	1 30	Rennes	35238	true
7	*NULL*	MULTIPOINT	1035	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	1 50	Rennes	35238	true
8	*NULL*	MULTIPOINT	1038	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	1	Rennes	35238	true
9	*NULL*	MULTIPOINT	1039	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	1	Rennes	35238	true
10	*NULL*	MULTIPOINT	1047	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	31 2	Rennes	35238	true
11	*NULL*	MULTIPOINT	1050	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	2	Rennes	35238	true
12	*NULL*	MULTIPOINT	1054	2018-09-04Z	Abri simple	OUI	NON	NON	NON	*NULL*	11 2 1	Rennes	35238	false
13	*NULL*	MULTIPOINT	1062	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	*NULL*	2	Rennes	35238	false

Jointure attributaire

```
alter table bus add column PMR text
```

```
update bus set PMR = (select estaccessiblepmr from bus2020 where
bus.ap_timeo == bus2020.code)
```

Gestion des données spatiales

Bases du spatial

- Afficher le SCR des géométries de la table

```
select srid(geom) from busrm
```

	srid(geom)
1	2154
2	2154
3	2154
4	2154

- Afficher les coordonnées géographique d'une couche

```
select code as code, x(geom) as Long, y(geom) as lat from busrm
```

	code	Long	lat
1	4007	361091.475829	6791866.63547
2	4502	344566.321169	6801962.94216
3	4530	344720.32956	6803351.39947
4	4531	343551.248981	6802111.62565
5	4509	343721.7598	6802421.69389
6	4304	341900.768778	6791502.4654

Changer le CRS

- Créer deux colonnes (X;Y) dans la table Bus pour afficher les coordonnées en WGS84

```
UPDATE busrm SET X= (x(Transform(geom, 4326)))
```

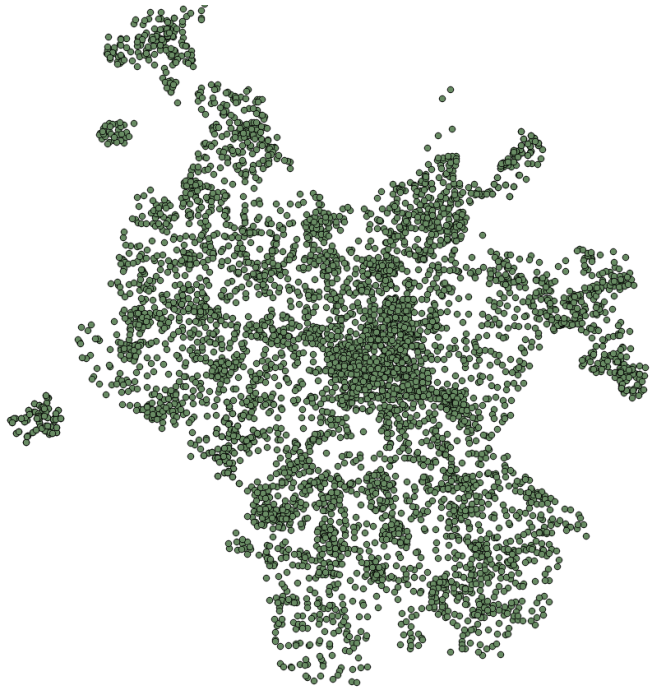
	nom	code	X	Y
1	Boulais	4007	-1.558744	48.139094
2	Gévezé Mairie	4502	-1.788723	48.220986
3	Lande de Villé	4530	-1.787786	48.23354
4	Cheval Blanc	4531	-1.802487	48.221767
5	Coualeuc	4509	-1.800449	48.224645
6	L'Hermitage Ce...	4304	-1.815962	48.125578
7	L'Hermitage Gare	4310	-1.817992	48.12375
8	Noë Biche	4315	-1.802546	48.122997
9	Rouesnais Sud	4750	-1.742873	47.962967
10	Gripay	4758	-1.726359	47.94704

Centroïdes de polygones

- Charger la couche des centroïdes de la table PLU

```
select id, typezone, centroid(geom) as geom from PLU
```

	id	typezone	geom
1	1	U	
2	2	Nh	
3	3	Nh	
4	4	Nh	
5	5	Nh	
6	6	U	
7	7	AUc	
8	8	N	



Opérateurs topographiques

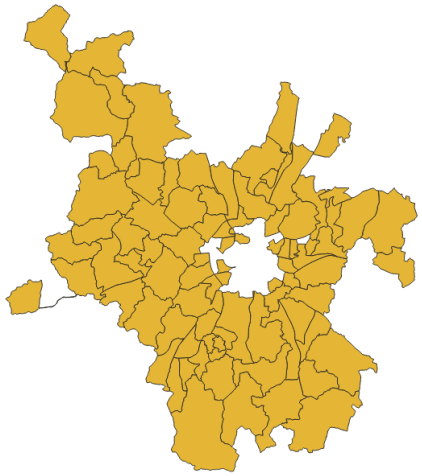
Opérateurs topographiques

- **ST_Area()** retourne la surface d'un objet ;
- **ST_Buffer()** retourne un nouvel objet tampon construit autour d'un objet ;
- **ST_Length()** : retourne la longueur d'un objet de type ligne ou multi-ligne (attention à ne pas utiliser length() qui retourne la longueur du champ, spatialite autorise aussi Glength()) ;
- **ST_Perimeter()** : retourne le périmètre d'un objet polygone ou multi-polygone.

Requêtes spatiales topographiques

- **Sélectionner des entités selon leurs surfaces**
  - Sélectionner les IRIS de RM de plus d'1 km2 (88 IRIS)

```
select *
from IRISPopulation
where (Area('IRISPopulation'.'Geometry') / 1000000) > 1
```



Requêtes spatiales topographiques

- **Sélectionner des entités selon leurs surfaces**
  - Sélectionner les IRIS de RM de plus d'10 km2 (22 IRIS)
  - Sélectionner les IRIS de RM de moins de 10 hectares (2 IRIS)
  - Sélectionner les IRIS de RM entre 1km² et 5 km² (38 IRIS)

Calculer des surfaces

- **Afficher les surfaces des quartier (en km2) par ordre décroissant**

```
SELECT nmquart, (Area(geom) / 1000000) as surface
FROM quartiers order by surface desc
```

	nmquart	surface
1	VILLEJEAN - BEAUREGARD	8.452159231031...
2	CLEUNAY - ARSENAL - REDON	7.26520915365062
3	MAUREPAS - PATTON	6.511026862774...
4	FRANCISCO-FERRER - VERN - PO...	4.859243407629...
5	BOURG L'EVESQUE - LA TOUCHE ...	4.210140800686...
6	JEANNE D'ARC - LONGS CHAMP...	4.077159278340...
7	THABOR - SAINT-HELIER - ALPH...	3.590117947501...
8	LE BLOSNE	2.687483291492...
9	BREQUIGNY	2.632401108304...

Calculer des surfaces

- **Afficher les surfaces des zones du PLU (en hectare) par ordre décroissant**

	objectid	typezone	surface
1	5414.0	N	2658.106807600...
2	1889.0	A	1449.124421807...
3	4021.0	A	1139.197074569...
4	1271.0	A	1054.451755794...
5	2235.0	A	928.3143645258...
6	651.0	A	891.6754122306...
7	2920.0	A	831.7632936974...
8	4414.0	A	817.9189515099...

Calculer des surfaces

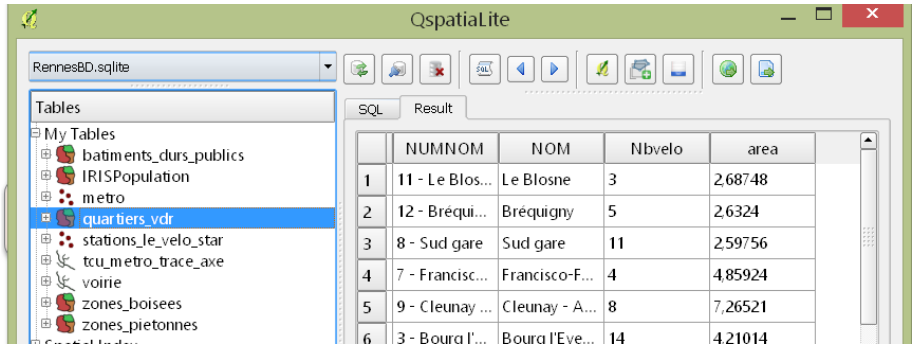
- Afficher les surfaces des zones du PLU (en hectare) par ordre décroissant

```
SELECT objectid, typezone, (Area(geom) / 10000) as surface
FROM PLU order by surface desc
```

Calculer des surfaces

- **Ajouter les surfaces des quartiers (en km2) dans la table des quartiers**
  - Ajouter une colonne area» en flottant dans la table
  - Exécuter la requête SQL

```
update 'quartiers_vdr' set 'area' = (select (area( 'quartiers_vdr'.'Geometry')/1000000))
```



Requêtes spatiales topographiques

- Sélectionner les zones du PLU ayant une superficie supérieure à 1km² (136 zones)
- Sélectionner les zones du PLU ayant une superficie entre 2 km² et 3 km² (21 zones)
- Ajouter à la table PLU une colonne superficie (en Km²)

	id	geom	libelong	typezone	objectid	libelle	id_docurba	geo_point_2d	surface
1	1	MULTIPOLYGON	Urbanisé	U	3189	UA1	3506620151231	(2:48.033920286...	0.0601542225713
2	2	MULTIPOLYGON	Naturel constru...	Nh	3220	Nh	3521620151231	(2:48.186998727...	0.00193044981049
3	3	MULTIPOLYGON	Naturel constru...	Nh	3222	Nh	3521620151231	(2:48.185684193...	0.0175504237171
4	4	MULTIPOLYGON	Naturel constru...	Nh	3236	Nh	3521620151231	(2:48.178167121...	0.00573093683271
5	5	MULTIPOLYGON	Naturel constru...	Nh	3239	Nh	3521620151231	(2:48.185847053...	0.00772459608686
6	6	MULTIPOLYGON	Urbanisé	U	3285	UE1	3527520151231	(2:48.153384253...	0.03016960287

Calculer des distance

- **Sélectionner les stations de vélos à moins de 200m d'un métro (24 entités)**
  - Opérateur **Distance**

```
SELECT * FROM velos, metro
WHERE distance( velos.geom, metro.geom) <200
group by velos.id
```

OU

  - Opérateur **Buffer**

```
SELECT * FROM velos, metro
WHERE within( velos.'Geom', buffer(metro.'Geom',200))
group by velos.'id'
```

Interroger des distance

- Arrêts de bus à moins de 200m d'un métro (115)
- Arrêts de bus à moins de 500m d'un métro (294)
- Arrêts de bus à moins de 200m d'une station de vélo(229)
- Arrêts de bus entre 300m et 500m d'un métro (194)
- Arrêts de bus à moins de 200m d'un métro et à moins de 200m d'une station de vélo (91)

Calculer des indicateurs de distance

- Afficher les distance min/max et moyenne des arrêts de bus aux stations de métro

```
SELECT min(distance( bus.geom, metro.geom ))  as DistanceMin,
max(distance( bus.geom, metro.geom ))  as DistanceMax ,
avg(distance( bus.geom, metro.geom ))  as DistanceMoyenne
FROM bus, metro
```

	DistanceMin	DistanceMax	DistanceMoyenne
1	16.542430529923212	32216.044983371125	6732.47970364617

Calculer des indicateurs de distance

- **Afficher les distance min/max et moyenne des stations de vélos aux stations de métro**

	DistanceMin	DistanceMax	DistanceMoyenne
1	23.26398903933...	7532.088415017...	2608.394665171...

Créer une matrice de distances

- Créer une matrice de distance entre les stations de métro et les stations de vélos

```
SELECT metro.nom as Nom_Metro,  velos.nom as Velo_Nom,
Distance(metro.geom, velos.geom) as distance
FROM metro, velos
order by Nom_metro asc
```

	Nom_Metro	Velo_Nom	distance
56	Atalante	Mairie	4150.62219152
57	Atalante	TNB	4025.67216904
58	Atalante	Brest - Verdun	5102.50932181
59	Atalante	Paul Bert	3654.77068781
60	Atalante	Beaulieu Chimie	1265.07143923
61	Atalante	Villejean-Univer...	5702.18362371
62	Atalante	Clemenceau	5091.37861496
63	Atalante	Gros-Chêne	2758.91453748
64	Atalante	La Poterie	4589.2501458
65	Atalante	Armorique	3540.25472416

Calculer des longueurs

- **Pour calculer des longueurs > opérateur st_length**
  - Je cherche à calculer les longueurs des tronçons du schémas directeur de vélo

```
select id, type, domanialite, st_length(geom) as longeur from schemavelo
```

	id	type	domanialite	longeur
1	1	Liaison alternative	Communal	113.8429433247...
2	2	Liaison secondaire	Rennes Métropole	16.97724401200...
3	3	Liaison secondaire	Rennes Métropole	844.8801842724...
4	4	Liaison secondaire	Rennes Métropole	16.55421918414...
5	5	Liaison secondaire	Rennes Métropole	81.70719746178...
6	6	Liaison principale	Rennes Métropole	63.46556203288...
7	7	Liaison principale	Rennes Métropole	938.7156273798...
8	8	Liaison secondaire	Rennes Métropole	583.8756490118...

Calculer des longueurs

- Calculer la longueur totale du réseau du schéma directeur vélo (en Km)

	Longueur
1	666.0482310443...

- Calculer le nombre de tronçons et la somme des longueurs des tronçons du schéma directeur vélo par type

	type	Nb_Troncons	Longueur
1	*NULL*	1	0.1453246125064...
2	Liaison alternative	128	111.1414979588...
3	Liaison non retenue	66	52.26708069512...
4	Liaison principale	300	105.5149515484...
5	Liaison secondaire	698	396.9793762293...

Calculer des longueurs

```
select sum((st_length(geom)/1000)) as Longueur from schemavelo
```

```
select type, count(id) as Nb_Troncons, sum((st_length(geom)/1000)) as
Longueur from schemavelo
group by type
```

Calculer des longueurs

- Ajouter à la table *schemadirecteurvelo* une nouvelle colonne avec la longueur des tronçons (en m)

	id	geom	num_iti	nom_iti	etat	type	domanialite	voie_verte	observation	objectid	longueur
1	1	MULTILINESTRING	NULL	NULL	Réalisé	Liaison alternative	Communal	Oui	NULL	99.0	113.8429433247...
2	2	MULTILINESTRING	NULL	NULL	Réalisé	Liaison secondaire	Rennes Métropole	NULL	NULL	8.0	16.97724401200...
3	3	MULTILINESTRING	NULL	NULL	Réalisé	Liaison secondaire	Rennes Métropole	Non	NULL	18.0	844.8801842724...
4	4	MULTILINESTRING	NULL	NULL	Réalisé	Liaison secondaire	Rennes Métropole	Non	NULL	24.0	16.55421918414...
5	5	MULTILINESTRING	NULL	NULL	Réalisé	Liaison secondaire	Rennes Métropole	Oui	NULL	28.0	81.70719746178...
6	6	MULTILINESTRING	NULL	NULL	A créer	Liaison principale	Rennes Métropole	Non	partiellement programmée (ZA...	129.0	63.46556203288...
7	7	MULTILINESTRING	NULL	NULL	A améliorer	Liaison principale	Rennes Métropole	Non	bandes cyclables à sécurisées par ilo...	42.0	938.7156273798...
8	8	MULTILINESTRING	NULL	NULL	Réalisé	Liaison secondaire	Rennes Métropole	Oui	NULL	52.0	583.8756490118...
9	9	MULTILINESTRING	NULL	NULL	A améliorer	Liaison secondaire	Rennes Métropole	Non	projet piste bidir rive Nord	53.0	144.3763219320...
10	10	MULTILINESTRING	NULL	NULL	Réalisé	Liaison secondaire	Rennes Métropole	Non	NULL	61.0	1030.767591538...

Calculer des longueurs

```
alter table schemavelo add column longueur interger
```

```
update schemavelo set longueur= st_length(geom)
```

Exercice - Zonage

- Créer une table avec le type de zonage, le nombre de parcelles par type de zonage et la superficie totale des types de zonages

```
select typezone, count(typezone) as nbtype, sum(area(geom)) as Surface
from PLU
group by typezone
order by Surface desc
```

	typezone	nbtype	Surface
1	A	528	276492184.3064...
2	N	1626	258769721.3196...
3	U	1937	121875264.1483...
4	Nh	835	24784556.51391...
5	AUc	325	20921429.78512...
6	AUs	155	17919005.30655...
7	*NULL*	0	343070.2392731...

Exercice - Zonage

- **Calcul de la proportion des superficies des libellés de zonages**
  - 1. Calcul de la superficie totale (MAJ d'un champ existant)

```
update PLU set shape_area= (select sum(surface) from PLU)
```

  - 2. Calcul des proportions (création d'un nouveau champ)

```
Alter table PLU add column propzonage integer
```

  - Calcul de la variable

```
update PLU set propzonage = ((surface/shape_area)*100)
```

Exercice - Zonage

- 3. Calcul de l'emprise spatiale des zonages

```
select type, count(type) as nbtype,
sum(propzonage) as Prop
from PLU
group by type
order by Prop desc
```

	type	nbtype	Prop
1	UG2	122	14.3314601995
2	NE	31	12.9361965345
3	UM	10	7.29062602454
4	UI1	17	5.30711045889
5	UE1	85	5.13944673574
6	A	9	4.91497840512
7	UE2a	59	4.75344647407
8	NP	5	4.34066539929
9	AUL	18	4.3173826833
10	UC	72	3.98768579533
11	UD	68	3.24891555019
12	UB2	20	2.05227223524

Exercice - Zonage

- Calculer la surface et la proportion par type de zonage

	typezone	surface	prop
1	A	276492184.3064...	38.34283433023...
2	N	258769721.3196...	35.88515378519...
3	U	121875264.1483...	16.90117597325...
4	Nh	24784556.51390...	3.437023533757...
5	AUc	20921429.78511...	2.901300513122...
6	AUs	17919005.30655...	2.484936250748...
7	*NULL*	343070.2392737...	0.047575613687...

Exercice zonage

- **Calculer par commune, la surface de chaque type de zonage**

	nomcommune	typezone	surface
1	Acigné	A	14991545.89852...
2	Acigné	AUc	145651.7653310...
3	Acigné	AUs	650323.8206478...
4	Acigné	N	11586683.23228...
5	Acigné	Nh	741434.559233148
6	Acigné	U	2063878.646241...
7	Betton	A	10799468.73452...
8	Betton	AUc	778482.6397504...
9	Betton	AUs	803422.861951926
10	Betton	N	8411668.7972916
11	Betton	Nh	1792035.195937...

Exercice zonage

	typezone	nomcommune	surface
1	*NULL*	Chantepie	9885.488057002...
2	*NULL*	Rennes	333184.7512167...
3	A	Acigné	14991545.89852...
4	A	Betton	10799468.73452...
5	A	Bourgbarré	8881901.152187...
6	A	Bruz	4867164.285015...
7	A	Brécé	3682483.300287...
8	A	Cesson-Sévigné	10648247.38915...
9	A	Chantepie	2667995.447546...
10	A	Chartres-de-Br...	2819758.661058...
11	A	Chavagne	6194034.078406...

Exercice zonage

- Calculer le superficie et la proportion des zonages par communes

	nomcommune	typezone	Proportion	Surface_totale	Nb_zones
1	Acigné	A	49.67457047241...	14991545.89852...	118
2	Acigné	N	38.39253914567...	11586683.23228...	20
3	Acigné	U	6.838673339837...	2063878.646241...	53
4	Acigné	Nh	2.456747523744...	741434.559233148	44
5	Acigné	AUs	2.154851586175...	650323.8206478...	6
6	Acigné	AUc	0.482617932155...	145651.7653310...	3
7	Betton	A	40.34827467735...	10799468.73452...	14
8	Betton	N	31.42713141463...	8411668.7972916	16
9	Betton	U	15.6190954932294	4180548.853438...	69
10	Betton	Nh	6.695285675122...	1792035.195937...	184
11	Betton	AUs	3.001696390164...	803422.861951926	8

Exercice zonage

> Vue de la table finale PLU

	id	geom	objectid	id_docurba	libelle	libelong	typezone	destdomi	codecommune	nomcommune	surface	surfacecommune	propcommune
1	1	MULTIPOLYGON	3662.0	0000353152015...	UE1	Urbanisé	U	*NULL*	35315	Saint-Sulpice-la...	6997.090782256...	13263225.40819...	0.052755574657...
2	2	MULTIPOLYGON	3668.0	0000353152015...	UE2	Urbanisé	U	*NULL*	35315	Saint-Sulpice-la...	53393.32052307...	13263225.40819...	0.402566637298...
3	3	MULTIPOLYGON	3727.0	0000353512015...	Ah	Agricole	A	*NULL*	35351	le Verger	3932.662547070...	13979781.92294...	0.028131072206...
4	4	MULTIPOLYGON	3743.0	0000353512015...	Nh	Naturel constru...	Nh	*NULL*	35351	le Verger	15357.10620336...	13979781.92294...	0.109852258697...
5	5	MULTIPOLYGON	3776.0	0000353522015...	Nh1	Naturel constru...	Nh	*NULL*	35352	Vern-sur-Seiche	45753.30277741...	19897390.36518...	0.229946248918...
6	6	MULTIPOLYGON	3900.0	0000353522015...	Nh	Naturel constru...	Nh	*NULL*	35352	Vern-sur-Seiche	1540.638641045...	19897390.36518...	0.007742918105...
7	7	MULTIPOLYGON	3873.0	0000353522015...	UE2	Urbanisé	U	*NULL*	35352	Vern-sur-Seiche	77598.29173055...	19897390.36518...	0.389992307063...
8	8	MULTIPOLYGON	3903.0	0000353522015...	Nh	Naturel constru...	Nh	*NULL*	35352	Vern-sur-Seiche	1648.892856571...	19897390.36518...	0.008286980484...
9	9	MULTIPOLYGON	3888.0	0000353522015...	A	Agricole	A	*NULL*	35352	Vern-sur-Seiche	260882.6690037...	19897390.36518...	1.311140125491...
10	10	MULTIPOLYGON	3965.0	0000352382015...	NEh	Naturel constru...	Nh	*NULL*	35238	Rennes	26398.84427758...	50306033.45212...	0.052476497282...
11	11	MULTIPOLYGON	1619.0	0000350242015...	UI1	Urbanisé	U	*NULL*	35024	Betton	27877.93935623...	26765627.08289...	0.104155748975...
12	12	MULTIPOLYGON	1880.0	0000350882015...	N	Naturel	N	*NULL*	35088	Corps-Nuds	43259.07371587...	22873988.43455...	0.189119067886...
13	13	MULTIPOLYGON	1924.0	0000350882015...	N	Naturel	N	*NULL*	35088	Corps-Nuds	167803.4635594...	22873988.43455...	0.733599494637...
14	14	MULTIPOLYGON	1932.0	0000353532015...	Ah	Agricole	A	*NULL*	35353	Vezin-le-Coquet	2106.726872933...	7923950.138030...	0.026586826472...
15	15	MULTIPOLYGON	4549.0	0000352382015...	1AUO	A Urbaniser alte...	AUc	*NULL*	35238	Rennes	97040.70910807...	50306033.45212...	0.192900736649...

Exercice zonage

- **Calculer le superficie et la proportion des zonages par communes**

	nomcommune	surfacecommune	sum(surface)	sum(propcommune)
1	Acigné	30179517.92226...	30179517.92226166	100.00000000000013
2	Betton	26765627.08289...	26765627.08289044	100.0
3	Bourgbarré	14778881.87226...	14778881.872261617	100.00000000000006
4	Bruz	29925737.24123...	29925737.241236206	99.99999999999996
5	Brécé	7230039.220598...	7230039.220598982	99.99999999999997
6	Cesson-Sévigné	32900802.70533...	32900802.705334682	100.00000000000006
7	Chantepie	11952774.82579...	11952774.825795433	99.99999999999997
8	Chartres-de-Br...	9942400.799930...	9942400.799930943	99.99999999999994
9	Chavagne	23602370.76409...	23602370.76409219	99.99999999999999
10	Chevaigné	10375713.50527...	10375713.505275898	100.0

Vérification de rigueur

Exercice zonage

```
select nomcommune, typezone, sum(surface) as surface from PLU group by nomcommune, typezone
```

```
select  typezone, nomcommune, sum(surface) as surface from PLU group by typezone, nomcommune
```

```
select nomcommune, codecommune,  sum(surface) as surface from PLU group by nomcommune
```

```
update PLU set surfacecommune = (select surface from etape2 where etape2.codecommune == PLU.codecommune)
```

```
update PLU set propcommune = ((surface / surfacecommune)*100)
```

```
select nomcommune, typezone, sum(propcommune) as Proportion, sum(surface) as Surface_totale, count(*) as Nb_zones from PLU group by nomcommune, typezone order by nomcommune asc, Proportion desc
```

Opérateurs topologiques

Requêtes spatiales topologiques

Prédicat	Types d'objets (P point, L polyligne, S polygone)	Conditions
Equals	**Tous**	A **Equals** B si les objets sont géométriquement identiques : relation topologique (le nombre de sommets des 2 objets peut être différent)
Disjoint	**Tous**	A **Disjoint** B si les objets n'ont aucun point commun (intérieur et limite). (Inverse de Intersects)
Touches	**S/S, L/S, L/L, P/S, P/L**	A **Touches** B si les limites des objets ont au moins un point commun et si les intérieurs n'ont pas de point commun (non applicable à P/P)
Crosses	**P/S, P/L, L/S, L/L**	A **Crosses** B si les intérieurs ont au moins un point commun mais pas tous et si la dimension de l'intersection des intérieurs est inférieure à la dimension maximale des objets A et B (non applicable à P/P, S/S)
Within	**Tous**	A **Within** B si tout point de A est un point de B et si les intérieurs ont au moins un point commun (aucun point de A n'est à l'extérieur de B). (Inverse de Contains)
Contains	**Tous**	A **Contains** B si tout point de B est un point de A et si les intérieurs ont au moins un point commun (aucun point de B n'est à l'extérieur de A). (Inverse de Within)
Overlaps	**S/S, L/L, P/P**	A **Overlaps** B si à la fois : - A et B ont la même dimension (non applicable à P/L, P/S, L/S) - A et B ont des points en commun, mais pas tous - L'intersection des intérieurs de A et de B a la même dimension que A et B
Intersects	**Tous**	A **Intersects** B si A et B ont au moins un point commun (intérieur ou limite) (Inverse de Disjoint)
Covers (*)	**Tous**	A **Covers** B si aucun point de B n'est à l'extérieur de A (tout point de B est un point de A) (à comparer à Contains)
CoveredBy (*)	**Tous**	A **CoveredBy** B si aucun point de A n'est à l'extérieur de B (tout point de A est un point de B) (à comparer à Within)
Relate (A,B, DE-9IM Pattern Matrix)	**Tous**	Exprime la relation spatiale de A et de B à l'aide de la matrice modèle DE-9IM Permet la généralisation des prédicats spatiaux aux 98 relations topologiques. Ex : **Relate** (A, B, « 0F1FF0102 ») <=> A Intersects B

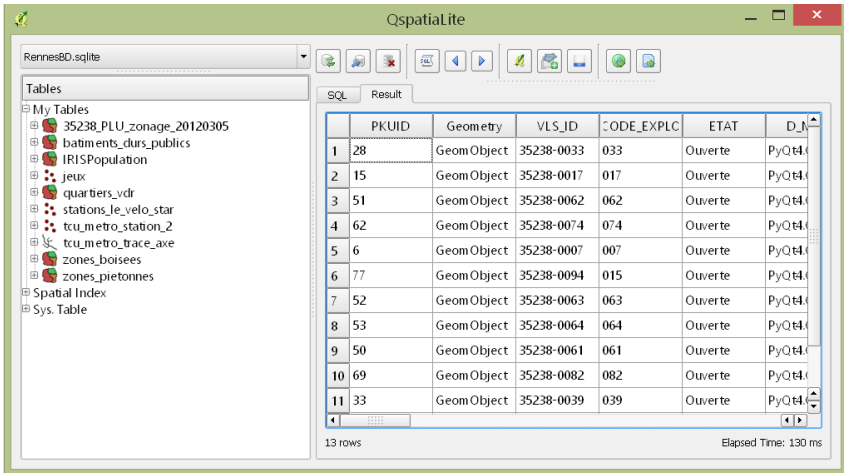
(*) : Prédicats non définis par la norme OGC, présents dans Oracle spatial, JTS, GEOS, PostGis

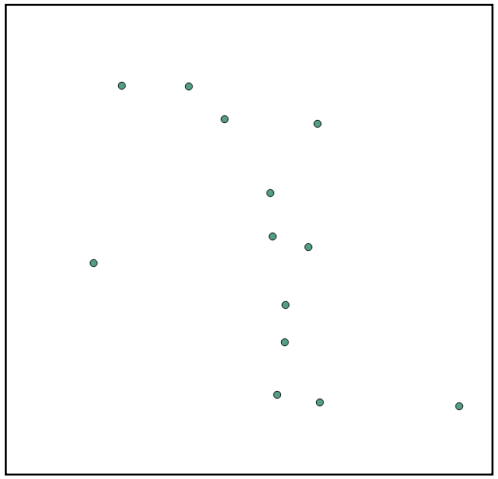
Prédicats de l'OGC

Requêtes spatiales

- **Sélectionner des entités selon leur distance (par le biais de buffers)**
  - Sélectionner les stations de vélos situées à moins de 100 mètres d'une station de métro (13 stations)

```
select * from stations_le_velo_star, tcu_metro_station_2
where within( stations_le_velo_star.'Geometry', buffer(tcu_metro_station_2.'Geometry',100))
group by stations_le_velo_star.'NOM'
```





Requêtes spatiales topographiques

- Compter le nombre d'arrêts de bus à moins de 200m d'une station de métro (27)

```
SELECT COUNT(*)
FROM
(SELECT *FROM bus, metro where ST_Distance(bus.geom,metro.geom) <200
group by metro.nom ) as Bus200m
```

- Compter le nombre d'arrêts de bus à moins de 200m d'un métro et à moins de 300m d'une station de vélos (19)

```
SELECT COUNT(*)
FROM
(SELECT *FROM bus, metro, velos where ST_Distance(bus.geom,metro.geom)
<200 and ST_Distance(bus.geom,velos.geom) <300
group by metro.nom ) as Bus200m
```

Requêtes spatiales topologiques

- Sélectionner les IRIS possédant un arrêt de métro (30 IRIS)

```
SELECT *
FROM referent_iris,metro
WHERE Contains(referent_iris.'Geometry',metro.'geom')
```

- Sélectionner les stations de métro situées dans des zones piétonnes (4 stations)

```
SELECT *
FROM metro, zones_pietonnes
WHERE Within(metro.'geom',zones_pietonnes.'Geometry')
```

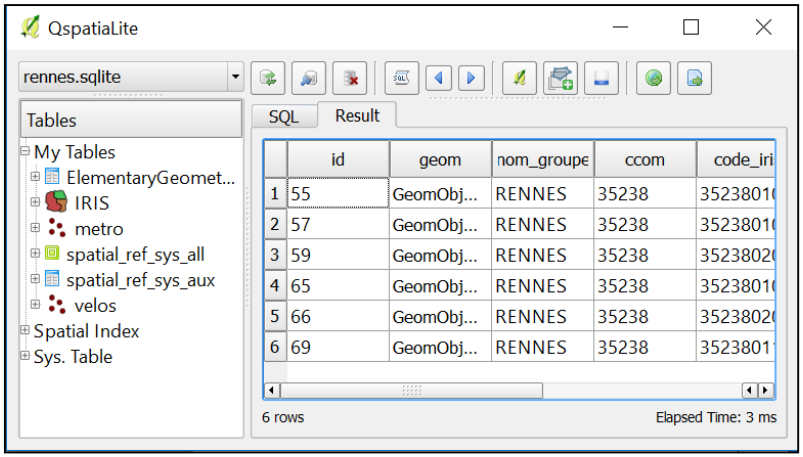
- Sélectionner les bâtiments public du Blosne (132 batiments)

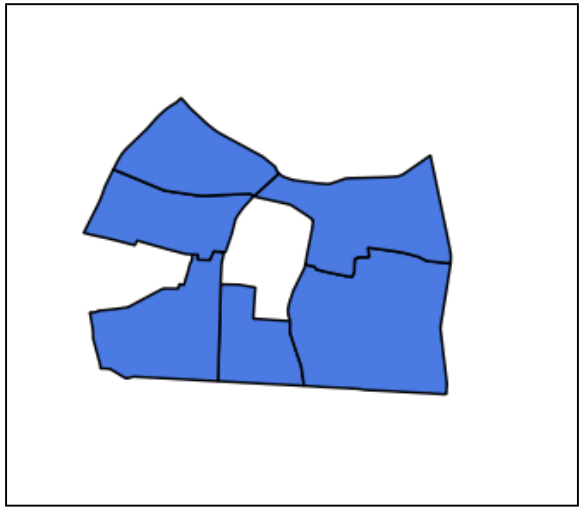
```
SELECT *
FROM batiments_durs_publics, quartiers_vdr
WHERE intersects(batiments_durs_publics.'Geometry',quartiers_vdr.'Geometry') and
quartiers_vdr.'NUQUART' = 11
```

Requêtes spatiales topologiques

- **Sélectionner les IRIS contiguës à l'IRIS Hoche»**
  - On mobilise ici l'opérateur *Touches*

```
SELECT *
FROM IRIS
WHERE (Touches ((select IRIS.'geom' from IRIS
where IRIS.'niris' =HOCHE), IRIS.'geom' ))
```





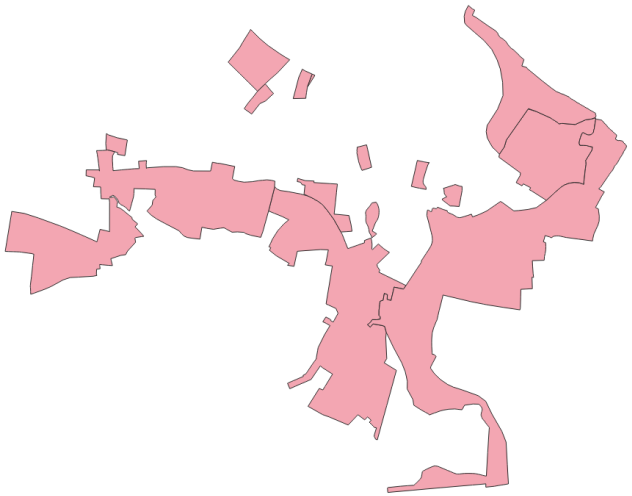
Requêtes spatiales topologiques

- **Sélectionner les zones du PLU connexes à la zone '577'**

```
SELECT * FROM PLU
WHERE(Touches ((select PLU.'geom' from PLU wherePLU.'id'=1644'),PLU.'geom'))
```

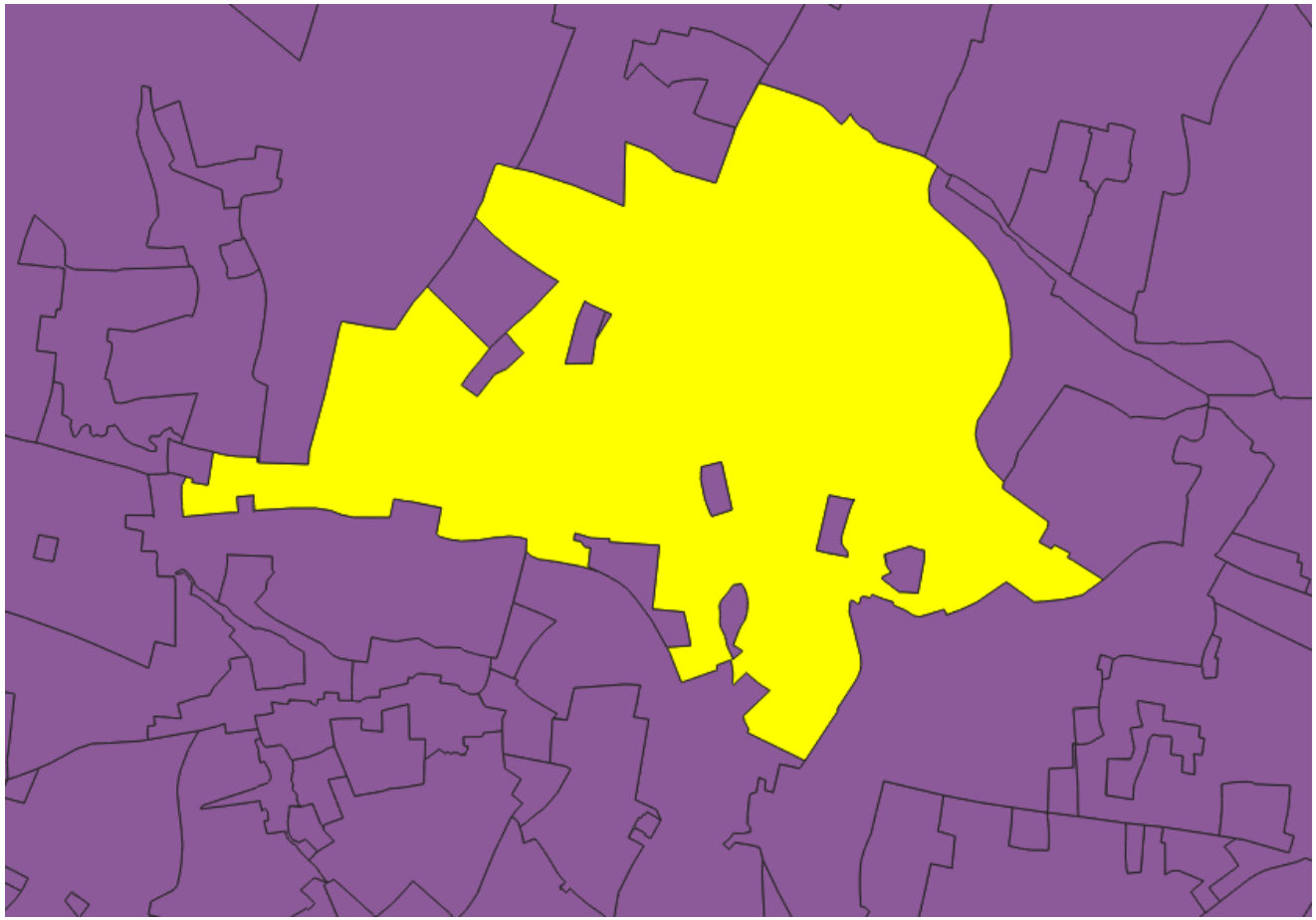
Exécuter | 15 lignes, 0.112 secondes | Créer une vue | Effacer

	id	geom	objectid	id_docurba	libelle	libelong	typezone	destdomi
1	4	b'\x00\x01j\x08\...	1940.0	0000353532015...	Ah	Agricole	A	*NULL*
2	96	b'\x00\x01j\x08\...	1935.0	0000353532015...	N	Naturel	N	*NULL*
3	97	b"\x00\x01j\x08\...	1942.0	0000353532015...	Ah	Agricole	A	*NULL*
4	102	b'\x00\x01j\x08\...	1963.0	0000353532015...	1AUO	A Urbaniser alternatif	AUc	*NULL*
5	1018	b'\x00\x01j\x08\...	1998.0	0000353532015...	NP	Naturel	N	*NULL*
6	1649	b'\x00\x01j\x08\...	1955.0	0000353532015...	Ah	Agricole	A	*NULL*
7	1656	b'\x00\x01j\x08\...	1978.0	0000353532015...	UE2	UrbanisÃ©	U	*NULL*
8	2229	b"\x00\x01j\x08\...	1941.0	0000353532015...	N	Naturel	N	*NULL*
9	2302	b'\x00\x01j\x08\...	1939.0	0000353532015...	Ah	Agricole	A	*NULL*
10	2304	b'\x00\x01j\x08\...	1936.0	0000353532015...	Ah	Agricole	A	*NULL*



Requêtes spatiales topologiques

- **Problème ???**



Requêtes spatiales

- **Sélectionner les zones du PLU à moins de 300m d'une station de métro**

```
SELECT * FROM PLU, metro
WHERE intersects(PLU.'geom',Buffer(metro.'geom',300)) group by PLU.'id'
```

- **Sélectionner les zones du PLU de type U à moins de 300m d'un arrêt de bus**

```
SELECT * FROM PLU, metro
WHERE intersects(PLU.'geom',Buffer(metro.'geom',300)) and PLU.'typezone'='U'
group by PLU.'id'
```

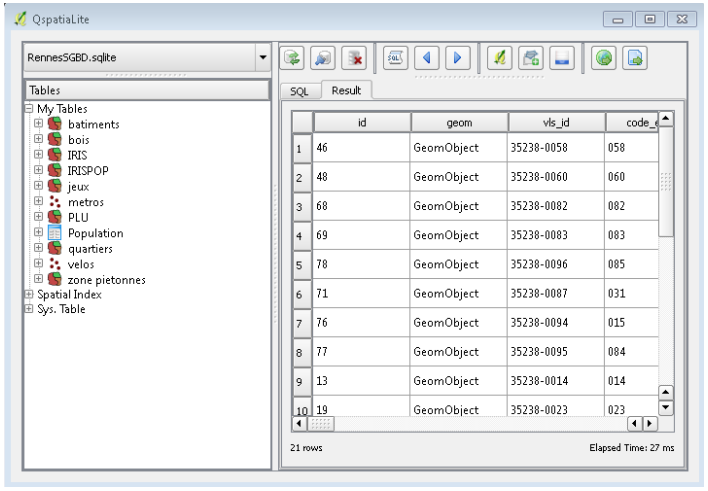
- **Sélectionner les zones du PLU de type U à moins de 500m d'un arrêt de bus particulier**

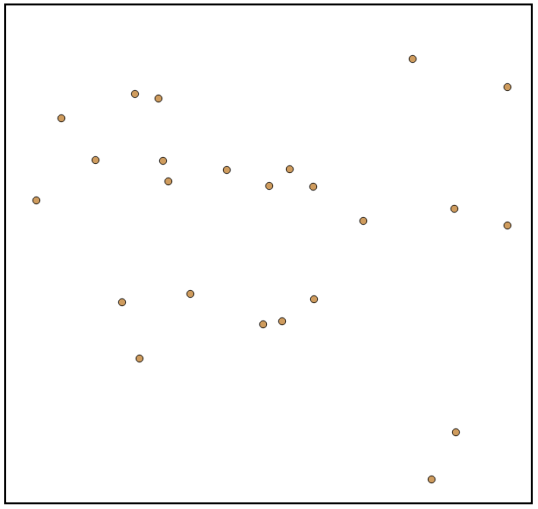
```
SELECT * FROM PLU, bus
WHERE (Within(PLU.'geom',Buffer(bus.'geom',500)) and bus.'id'='1')
and PLU.'typezone'='U'
```

Exercices

- **Sélectionner les stations de vélos situées dans des IRIS de plus de 3000 habitants (21 entités)**

```
SELECT *
FROM stations_le_velo_star, IRISPopulation
WHERE within(stations_le_velo_star.'Geometry',IRISPopulation.'Geometry') and
IRISPopulation.'Pop' > 3000
```

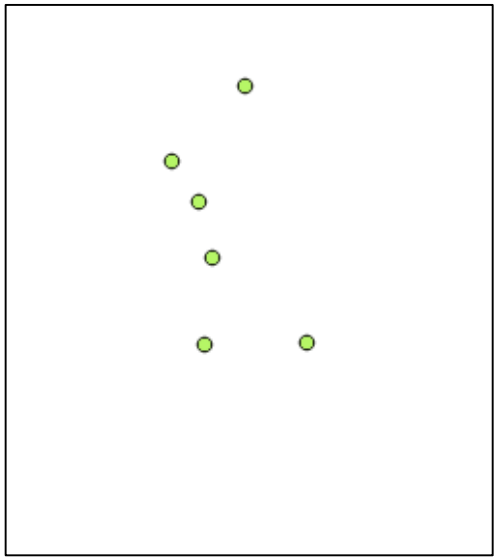




Exercices

- **Sélectionner les stations de vélos du quartier centre avec plus de 20 socles (6 entités)**

```
select  from velos, IRIS
where within(velos.'geom', iris.'geom')
and IRIS.'nmsectmorp' = 'Centre'
and velos.'nb_socles' >20
```



SQL | Result

	geom	nom	nb_socles
1	GeomObject	LibertÃ©	24
2	GeomObject	TNB	28
3	GeomObject	RÃ©publique	30
4	GeomObject	Mairie	24
5	GeomObject	Champ Jacquet	24
6	GeomObject	Place Hoche	24

6 rows Elapsed Time: 5 ms

Exercices

- **Sélectionner les IRIS remplissant les critères suivants : (3 entités)**
  - Stations Vélos star
  - Arrêt de métro
  - Zones piétonnes
  - Population supérieure à 3000 habitants

```
SELECT *
FROM IRISPopulation, stations_le_velo_star, zones_pietonnes, metro
WHERE IRISPopulation.'Pop' > 3000
and Contains(IRISPopulation.'Geometry',stations_le_velo_star.'Geometry')
and Contains(IRISPopulation.'Geometry',zones_pietonnes.'Geometry')
and Contains(IRISPopulation.'Geometry',metro.'geom')
GROUP BY IRISPopulation.'NMIRIS'
```

Exercices

- Sélectionner les arrêts de bus dans des zones agricole

```
select * from bus, PLU
where intersects(PLU.geom, bus.geom) and PLU.typezone like 'A%'
group by bus.id
```

- Sélectionner les zones du PLU connexes à des zones agricoles

Exercices

- **Compter pour chaque arrêt de bus, le nombre de parking à vélos situés à moins de 200m**

```
SELECT Arrets_bus.ap_timeo, count(*) as nbvelos
FROM Arrets_bus, amenity_bicycle_parking
WHERE
ST_Intersects(amenity_bicycle_parking.geom,ST_Buffer(Arrets_bus.geom,200))
GROUP BY Arrets_bus.ap_timeo
ORDER BY nbvelos DESC
```

-> Et si on inverse les critères spatiaux que se passet-il ?

Exercices

- **Calculer les superficies des types de zonage par quartier pour la ville de Rennes**

	Quartier	libelong	surface
4	BOURG L'EVESQUE - LA TOUCHE - M...	Agricole	239869.5517976...
5	BOURG L'EVESQUE - LA TOUCHE - M...	Naturel	313.5985015152...
6	BOURG L'EVESQUE - LA TOUCHE - M...	Naturel constructible (Art...	408933.1078407...
7	BOURG L'EVESQUE - LA TOUCHE - M...	UrbanisÃ©	3561014.776767...
8	BREQUIGNY	*NULL*	*NULL*
9	BREQUIGNY	A Urbaniser alternatif	*NULL*
10	BREQUIGNY	A urbaniser bloquÃ©	0. 0247041340921...
11	BREQUIGNY	Agricole	107244.2654943...
12	BREQUIGNY	Naturel	1. 291285021731104
13	BREQUIGNY	Naturel constructible (Art...	502369.6912812...

Exercices

- **Calculer les superficies des types de zonage par quartier pour la ville de Rennes**

```
select quartiers.nmquart as Quartier, PLU.libelong,
sum(area(st_intersection(PLU.geom, quartiers.geom))) as surface
from quartiers, PLU
group by quartiers.nmquart, PLU.libelong
```

Exercices

- **Calculer la proportion des zones urbanisées par quartier**

	id	Quartier	surfacetot	surfacePLU	prop
1	1	LE BLOSNE	2687483.29151	1918898.84702	71.4013312413
2	2	CLEUNAY - ARSENAL - REDON	7265209.15369	2367362.18029	32.5849143529
3	3	SAINT MARTIN	1800757.81623	1414101.81761	78.5281510298
4	4	VILLEJEAN - BEAUREGARD	8452159.23108	3404956.19568	40.2850455438
5	5	BREQUIGNY	2632401.10832	2022949.68147	76.8480789297
6	6	JEANNE D'ARC - LONGS CHAMPS - BEAULIEU	4077159.27836	3814087.60891	93.547672497
7	7	SUD GARE	2597560.99824	2597480.76865	99.9969113492
8	8	BOURG L'EVESQUE - LA TOUCHE - MOULIN D...	4210140.80071	3517272.81214	83.5428784603

Exercices

- **Calculer la proportion des zonages par quartiers de Rennes**

```
select quartiers.id, quartiers.nmquart as Quartier, area(quartiers.geom) as
surfacetot, sum(area(st_intersection(PLU.geom, quartiers.geom))) as
surfacePLU, (sum(area(st_intersection(PLU.geom,
quartiers.geom)))/area(quartiers.geom)*100) as prop
from quartiers, PLU where st_intersects(PLU.geom, quartiers.geom) and
PLU.libelong like 'Urba%' group by quartiers.id
```

Exercices

- **Calculer la longueur totale des tronçons du schéma directeur vélo par quartier**

	Quartier	longueur
2	CLEUNAY - ARSENAL - REDON	11815.23471334...
3	FRANCISCO-FERRER - VERN -...	11814.42117447...
4	VILLEJEAN - BEAUREGARD	10058.64684652...
5	THABOR - SAINT-HELIER - ALPHO...	8016.420096447...
6	JEANNE D'ARC - LONGS CHAMPS ...	7503.497621088...
7	CENTRE	6402.537215206...
8	BOURG L'EVESQUE - LA TOUCHE - M...	6284.060535711...
9	SUD GARE	5197.023095142...
10	NORD - SAINT-MARTIN	4023.849136066...
11	BREQUIGNY	3269.895541884...
12	LE BLOSNE	2692.766851428...

Exercices

- Calculer la longueur totale des tronçons du schéma directeur vélo par quartier

```
select quartiers.nmquart as Quartier,
sum(st_length(st_intersection(schemavelo.geom, quartiers.geom))) as longueur
from quartiers, schemavelo
group by quartiers.nmquart
order by longueur desc
```

Exercices

- **Calculer la longueur totale des tronçons du schéma directeur vélo par IRIS**

	IRIS	longueur
1	Noyal-Châtillon sur Seiche ouest	18456.13912605...
2	le Rheu est et sud	16452.06170132...
3	Chavagne	14663.27009482...
4	Saint-Erblon	13975.42777137...
5	Noyal-Châtillon sur Seiche est	13847.46707642...
6	Vern sur seiche nord	13416.06980429...
7	Vezin-le-Coquet	13158.98439809...
8	Betton est	12513.27625606...
9	Pont-Péan	12110.84586102...
10	Nouvoitou	11867.17916558...

Exercices

- **Calculer la longueur totale des tronçons du schéma directeur vélo par Commune (en utilisant la couche IRIS)**

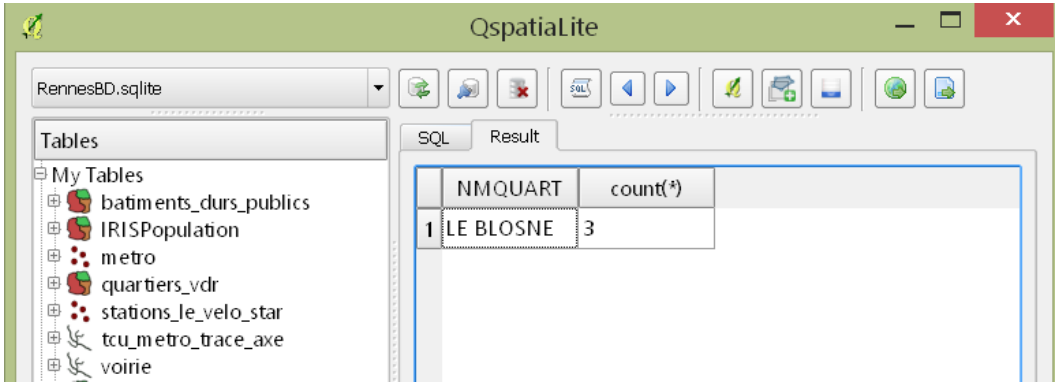
	IRIS	longueur
1	RENNES	89161.01015435...
2	CESSON-SEVIGNE	45899.77563383...
3	BRUZ	37406.28478352...
4	NOYAL-CHATILLON SUR SEICHE	32303.606202477
5	LE RHEU	24088.88251949...
6	SAINT GREGOIRE	22370.11958367...
7	VERN SUR SEICHE	21518.25104222...
8	BETTON	20230.04231941...
9	PACE	18537.68657986...
10	MORDELLES	18399.67271120...

Comptage et Jointure spatiale

Comptage

- **Afficher le nombre de points dans un polygone**
  - Connaître le nombre de stations vélos stars dans le quartier du Blosne
    - Exécuter cette requête SQL

```
SELECT quartiers_vdr.'NMQUART', count(*)
FROM 'stations_le_velo_star', 'quartiers_vdr'
WHERE 'quartiers_vdr'.'NUQUART' = '11' and
Intersects('stations_le_velo_star'.'Geometry','quartiers_vdr'.'Geometry')
```



Comptage

- Calculer le nombre d'arrêts de stations de vélos par IRIS

```
select nmiris as IRIS, count(*)  as NB_Velo FROM IRIS, velos WHERE
intersects(IRIS.'geom', velos.'geom') group by nmiris
```

	IRIS	NB_Velo
1	Albert de Mun	1
2	Alphonse Guérin	1
3	Arsenal	1
4	Beauregard	2
5	Campus de Bea...	2
6	Canada	1
7	Cathédrale	2
8	Champeaux	1
9	Cimetière de l'Est	1
10	Cimetière du N...	2

Comptage

- **Ajouter un champ avec le nombre de point par polygones**
  - Ajouter le nombre de stations vélos stars par quartier dans la table des quartiers
    - Créer une colonne nbstations» dans la table des quartiers
    - Exécuter cette requête SQL

```
update quartier set 'nbstations'= (select count(*) FROM velos
WHERE Intersects(quartier.'geom',velos.'geom'))
```

	id	geom	nmquart	code_insee	numquar	nbsport	nbculture	nbvelos	nbstations
1	1	MULTIPOLYGON	LE BLOSNE	352388.0	11	11	8	*NULL*	0
2	2	MULTIPOLYGON	CLEUNAY - AR...	352388.0	9	11	4	56	3
3	3	MULTIPOLYGON	SAINT MARTIN	352388.0	4	2	1	30	1
4	4	MULTIPOLYGON	VILLEJEAN - BE...	352388.0	10	13	11	120	5
5	5	MULTIPOLYGON	BREQUIGNY	352388.0	12	11	3	18	1
6	6	MULTIPOLYGON	JEANNE D'ARC ...	352388.0	6	7	7	100	4
7	7	MULTIPOLYGON	SUD GARE	352388.0	8	2	4	94	5
8	8	MULTIPOLYGON	BOURG L'EVES...	352388.0	3	7	2	124	5
9	9	MULTIPOLYGON	MAUREPAS - B...	352388.0	5	20	6	60	3
10	10	MULTIPOLYGON	THABOR - SAI...	352388.0	2	4	9	263	12
11	11	MULTIPOLYGON	FRANCISCO-FE...	352388.0	7	6	10	28	1
12	12	MULTIPOLYGON	CENTRE	352388.0	1	3	20	386	15

Comptage

- **Compter le nombre de zones de PLU par quartier en mobilisant le centroïde des zones**

```
update quartier set 'nbzones'= (select count(*) FROM PLU
WHERE contains(quartier.'geom', centroid(PLU.'geom')))
```

Vérifier le compte

Nb de zones comptabilisées dans la table des quartiers

```
select sum(nbzones) from quartier
```

Nb de zones à l'intérieur de la couche des quartiers

```
select count(*) FROM PLU,quartier
WHERE intersects(quartier.'geom', centroid(PLU.'geom'))
```

Comptage

- **Compter le nombre de bâtiments par IRIS (en utilisant les centroïdes)**

	nmiris	NB_batiments
1	La Poterie Nord	1112
2	Les Mottais	1093
3	Margueritte	1018
4	Cimetière du Nord	938
5	La Madeleine-Mauconseil	929
6	Gaëtan Hervé	883
7	Cleunay Est	830
8	Le Landry	787
9	Francisco Ferrer	784
10	Croix Saint-Hélier	773

Comptage

- Compter le nombre de bâtiments par IRIS (en utilisant les centroïdes)

```
select IRIS. nmiris, count(*) as
NB_batiments FROM IRIS, batirennes
WHERE contains(IRIS.geom,
centroid(batirennes.geom))
group by IRIS. nmiris
order by NB_batiments desc
```

Jointure spatiale

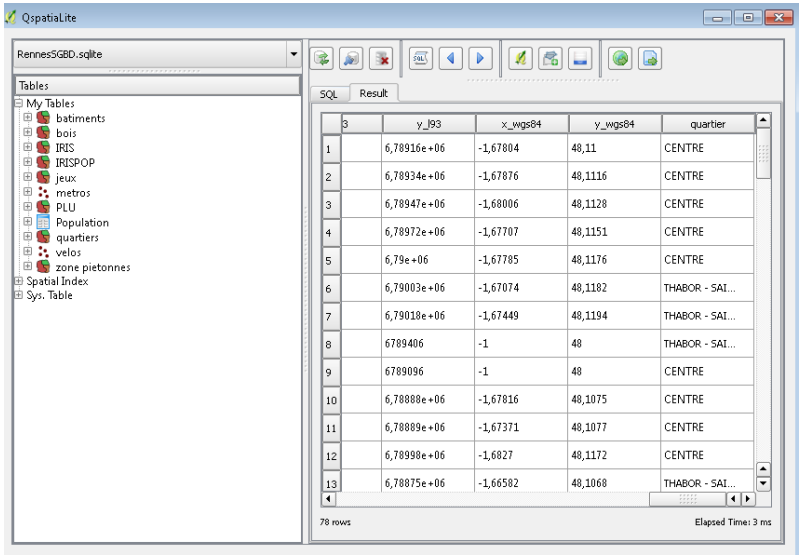
- **Attribuer des informations d'une couche à une autre selon un référent spatial**
  - Ici, nous voulons ajouter le nom du quartier à chacune des stations dans la table des stations vélos stars
  - Avant de faire la requête SQL il faut créer une nouvelle colonne
    - Dans le gestionnaire de base de données → sélectionner la table
    - Dans le menu Table → éditer la table
    - Ajouter une colonne → définir le nom de la colonne, son format (ici textuel) et sa longueur
  - Ou alors passer par une requête SQL pour créer une nouvelle colonne

```
Alter table velos add column quartier text
```

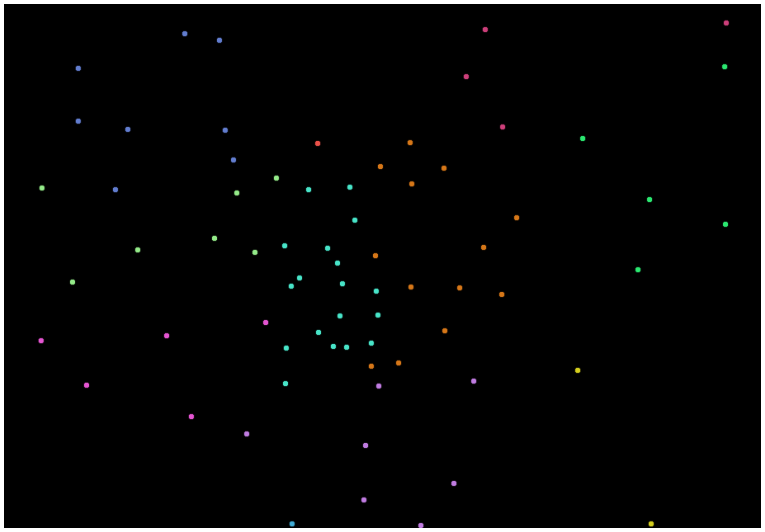
Jointure spatiale (texte)

- Retourner dans Spatialite
- Renseigner la requête SQL pour effectuer cette opération

```
update stations_le_velo_star set 'nomquartier'= (select quartiers_vdr.'NMQUART'
FROM quartiers_vdr
WHERE Intersects(quartiers_vdr.'Geometry',stations_le_velo_star.'Geometry'))
```



	3	y_l93	x_wgs84	y_wgs84	quartier
1		6,78916e+06	-1,67804	48,11	CENTRE
2		6,78934e+06	-1,67876	48,1116	CENTRE
3		6,78947e+06	-1,68006	48,1128	CENTRE
4		6,78972e+06	-1,67707	48,1151	CENTRE
5		6,79e+06	-1,67785	48,1176	CENTRE
6		6,79003e+06	-1,67074	48,1182	THABOR - SAI...
7		6,79018e+06	-1,67449	48,1194	THABOR - SAI...
8		6789406	-1	48	THABOR - SAI...
9		6789096	-1	48	CENTRE
10		6,78888e+06	-1,67816	48,1075	CENTRE
11		6,78889e+06	-1,67371	48,1077	CENTRE
12		6,78998e+06	-1,6827	48,1172	CENTRE
13		6,78875e+06	-1,66582	48,1068	THABOR - SAI...



Jointure spatiale (texte)

- Enrichir la couche *bus* du nom de l'IRIS

	id	geom	ap_timeo	ap_datemaj	equip_mobil	equip_banc	equip_eclai	equip_poube	access_pmr	access_li_o	access_li_n	nomcommune	codeinseeco	IIRIS
1	NULL	POINT	1004	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	1 50	Rennes	35238	Le Gallet-\Les L...
2	NULL	POINT	1020	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	57	Rennes	35238	La Madeleine-...
3	NULL	POINT	1024	2011-08-24Z	Abri bâti	NON	NON	NON	NON	NULL	1 5 9	Rennes	35238	Cathédrale
4	NULL	POINT	1025	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	5 1 9 8	Rennes	35238	Cathédrale
5	NULL	POINT	1026	2011-08-24Z	Abri auvent	NON	NON	NON	NON	NULL	8 1 9 5	Rennes	35238	Hoche
6	NULL	POINT	1032	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	1 30	Rennes	35238	Parc de Maurep...
7	NULL	POINT	1035	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	1 50	Rennes	35238	Campus de Bea...
8	NULL	POINT	1038	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	1	Rennes	35238	Les Longs Cha...
9	NULL	POINT	1039	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	1	Rennes	35238	Les Longs Cha...
10	NULL	POINT	1047	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	31 2	Rennes	35238	La Touche
11	NULL	POINT	1050	2011-08-24Z	Abri simple	OUI	OUI	NON	NON	NULL	2	Rennes	35238	Mail

- Compter le nombre d'arrêts de bus par IRIS

	IRIS	NB_Bus
1	Chartres de Bre...	52
2	Cesson-Sévign...	31
3	Orgères	27
4	St Jacques aéro...	26
5	St Jacques cent...	25
6	Beauregard	24
7	Pacé est	24

Jointure spatiale (texte)

- Enrichir la couche *bus* du nom de l'IRIS

```
alter table bus add column IRIS text
```

```
update bus set IRIS= (select IRIS.nmiris
FROM IRIS
WHERE Intersects(IRIS.geom, bus.geom))
```

```
Select  IRIS, count(*) as NB_Bus from bus
group by IRIS order by NB_Bus desc
```

Jointure spatiale (texte)

- **Enrichir la couche des bâtiments de Rennes du nom de l'IRIS**

	id	geom	objectid	texture	type_3d	territoire	IRIS
1	1	MULTIPOLYGON Z	64831	0	régie	intra-rocade	Arsenal
2	2	MULTIPOLYGON Z	64824	1	NULL	intra-rocade	Beauregard
3	3	MULTIPOLYGON Z	64823	1	NULL	intra-rocade	Beauregard
4	4	MULTIPOLYGON Z	64822	1	NULL	intra-rocade	Beauregard
5	5	MULTIPOLYGON Z	64821	1	NULL	intra-rocade	Beauregard
6	6	MULTIPOLYGON Z	64820	1	NULL	intra-rocade	Beauregard
7	7	MULTIPOLYGON Z	64819	1	NULL	intra-rocade	Beauregard
8	8	MULTIPOLYGON Z	64818	1	NULL	intra-rocade	Le Gast Ouest
9	9	MULTIPOLYGON Z	64817	1	NULL	intra-rocade	Le Gast Ouest
10	10	MULTIPOLYGON Z	64816	1	NULL	intra-rocade	Beauregard
11	11	MULTIPOLYGON Z	64815	1	NULL	intra-rocade	Morbihan Est
12	12	MULTIPOLYGON Z	64814	1	NULL	intra-rocade	Beauregard
13	13	MULTIPOLYGON Z	64813	1	NULL	intra-rocade	Beauregard
14	14	MULTIPOLYGON Z	64812	1	NULL	intra-rocade	Beauregard
15	15	MULTIPOLYGON Z	64811	1	NULL	intra-rocade	Beauregard

- **Compter le nombre bâtiment par IRIS en utilisant un group_by**

	IRIS	nb_batiments
1	La Poterie Nord	1112
2	Les Mottais	1093
3	Margueritte	1018
4	Cimetière du Nord	938
5	La Madeleine-Mauconseil	929
6	Gaëtan Hervé	883

Jointure spatiale (texte)

- **Enrichir la couche des bâtiments de Rennes du nom de l'IRIS**

```
alter table batirennes add column IRIS text
```

```
update batirennes set IRIS = (select IRIS.nmiris FROM IRIS WHERE Intersects(IRIS.geom, batirennes.geom))
```

```
select IRIS, count(*) as nb_batiments from batirennes group by IRIS order by nb_batiments desc
```

!! Les opérations d'enrichissement sémantique sur des volumes importants de données peuvent prendre plusieurs minutes !!

Jointure spatiale (résumé statistique)

- **Agréger pour chaque quartier le nombre de vélos en libre en service**
  - Il faut d'abord créer dans la table *quartier* un champ *nbvelos*

```
update quartier set 'nbvelos'= (select sum(velos.'nbvelos') FROM velos
WHERE Intersects(quartier.'geom',velos.'geom'))
```

	id	geom	nmquart	code_insee	numquar	nbsport	nbculture	nbvelos
1	1	MULTIPOLYGON	LE BLOSNE	352388.0	11	11	8	*NULL*
2	2	MULTIPOLYGON	CLEUNAY - AR...	352388.0	9	11	4	56
3	3	MULTIPOLYGON	SAINT MARTIN	352388.0	4	2	1	30
4	4	MULTIPOLYGON	VILLEJEAN - BE...	352388.0	10	13	11	120
5	5	MULTIPOLYGON	BREQUIGNY	352388.0	12	11	3	18
6	6	MULTIPOLYGON	JEANNE D'ARC ...	352388.0	6	7	7	100
7	7	MULTIPOLYGON	SUD GARE	352388.0	8	2	4	94
8	8	MULTIPOLYGON	BOURG L'EVES...	352388.0	3	7	2	124
9	9	MULTIPOLYGON	MAUREPAS - B...	352388.0	5	20	6	60
10	10	MULTIPOLYGON	THABOR - SAI...	352388.0	2	4	9	263
11	11	MULTIPOLYGON	FRANCISCO-FE...	352388.0	7	6	10	28
12	12	MULTIPOLYGON	CENTRE	352388.0	1	3	20	386

Jointure spatiale (résumé statistique)

- **Calculer la surface moyenne des maisons et des appartements pour chaque IRIS de Rennes Métropole**

	IRIS	Type	surface_Moyenne
1	Acigné est	Appartement	65.5
2	Acigné est	Maison	106.5373134328...
3	Acigné ouest	Appartement	64.03448275862...
4	Acigné ouest	Maison	104.2
5	Albert de Mun	Appartement	58.39455782312...
6	Albert de Mun	Maison	99.83636363636...
7	Alphonse Guérin	Appartement	59.21635883905...
8	Alphonse Guérin	Maison	101.6896551724...
9	Arsenal	Appartement	50.94779116465...
10	Arsenal	Maison	111.6

Jointure spatiale (résumé statistique)

- Calculer la surface moyenne des maisons et des appartements pour chaque IRIS de Rennes

```
select IRIS.nmiris as IRIS, DVF.type as Type, avg(DVF.surface) as surface_Moyenne
from IRIS, DVF where st_contains(IRIS.geom, DVF.geom)
group by IRIS.nmiris, DVF.type
```

Jointure spatiale (résumé statistique)

- **Calculer pour l'année 2018 et 2019 le prix moyen au m² et le nombre de vente par IRIS dans la ville de Rennes**
  - Il faut en amont créer des tables pour les ventes de Rennes en 2018 et en 2019 (deux tables différentes avec géométries)

2018

	IRIS	prix_moyen	NB_vente
1	Parcheminerie ...	3789.838709677...	31
2	Haut-Quineleu	3555.125	16
3	Vieux St.Etienne	3463.043478260...	23
4	Parlement	3443.291666666...	24
5	Les Longs Cha...	3429.933333333...	15

2019

	IRIS	prix_moyen	NB_vente
1	Dalle du Colom...	4196.4	5
2	La Mabilais	3813.538461538...	39
3	St-Hélier	3809.677419354...	31
4	Les Mottais	3773.541666666...	24
5	Jean Macé	3770.2	15

Jointure spatiale (résumé statistique)

```
select nmiris as IRIS, avg(2018.'prixm2') as prix_moyen,
count(*) as NB_vente FROM 2018 , IRIS
WHERE Intersects(IRIS.'geom',2018.'geom') group by
nmiris order by prix_moyen desc
```

```
select nmiris as IRIS, avg(2019.'prixm2') as prix_moyen,
count(*) as NB_vente FROM 2019, IRIS
WHERE Intersects(IRIS.'geom',2019.'geom') group by
nmiris order by prix_moyen desc
```

Jointure spatiale (résumé statistique)

- **Compter le nombre de carreaux par commune et calculer la population des communes à partir des données carroyées de l'INSEE**

	id	geom	gml_id	objectid	code_insee	nom	commune_agg	x_centrbrg	y_centrbrg	populationcarreaux	Nbcarreaux
1	1	MULTIPOLYGON	commune_em...	24.0	35278	Saint-Grégoire	1.0	1351561	7227380	9432	223
2	2	MULTIPOLYGON	commune_em...	90.0	35131	l'Hermitage	1.0	1341750	7225171	4079.5	93
3	3	MULTIPOLYGON	commune_em...	20.0	35189	Montgermont	1.0	1349369	7228155	3218.5	73
4	4	MULTIPOLYGON	commune_em...	75.0	35363	Pont-Péan	1.0	1349248	7212127	4219	101
5	5	MULTIPOLYGON	commune_em...	140.0	35080	Cintré	1.0	1337426	7223193	2332.5	94
6	6	MULTIPOLYGON	commune_em...	85.0	35144	Langan	1.0	1339783	7238521	*NULL*	0
7	7	MULTIPOLYGON	commune_em...	118.0	35001	Acigné	1.0	1362553	7224845	6791	223
8	8	MULTIPOLYGON	commune_em...	29.0	35051	Cesson-Sévigné	1.0	1357481	7223662	16889.5	363
9	9	MULTIPOLYGON	commune_em...	30.0	35059	la Chapelle-des...	1.0	1348281	7230494	4862.5	108
10	10	MULTIPOLYGON	commune_em...	41.0	35022	Bécherel	1.0	1333361	7244588	*NULL*	0
11	11	MULTIPOLYGON	commune_em...	28.0	35238	Rennes	1.0	1351714	7223283	174411.5	822
12	12	MULTIPOLYGON	commune_em...	139.0	35196	Mordelles	1.0	1339160	7219654	7523	307

Jointure spatiale (numérique)

- Compter le nombre de carreaux par commune et calculer la population des communes à partir des données carroyées de l'INSEE

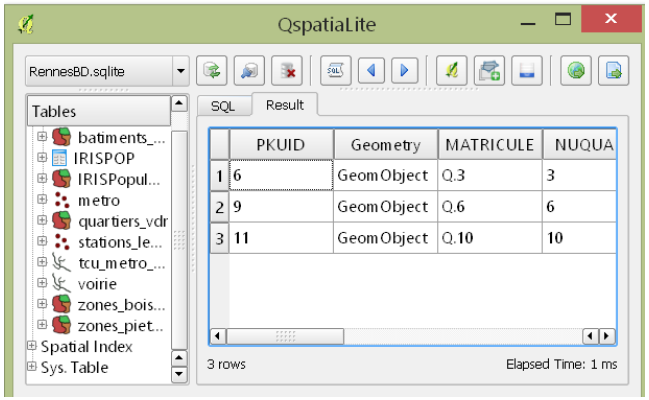
```
update communes set Nbcarreaux = (select count(*) from carreaux
where contains(communes.geom, centroid(carreaux .geom)))
```

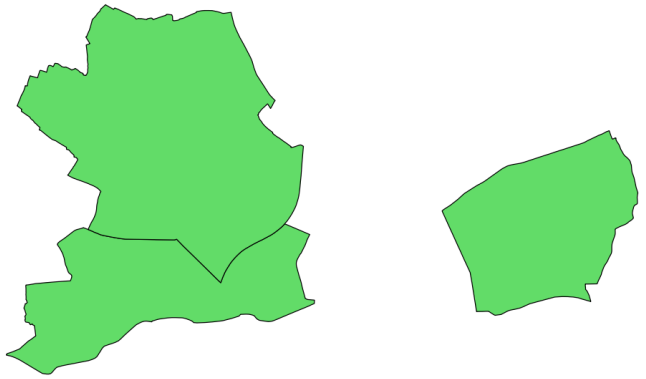
```
update communes set Populationcarreaux = (select sum(ind) from carreaux
where contains(communes.geom, centroid(carreaux .geom)))
```

Exercices

Aller plus loin

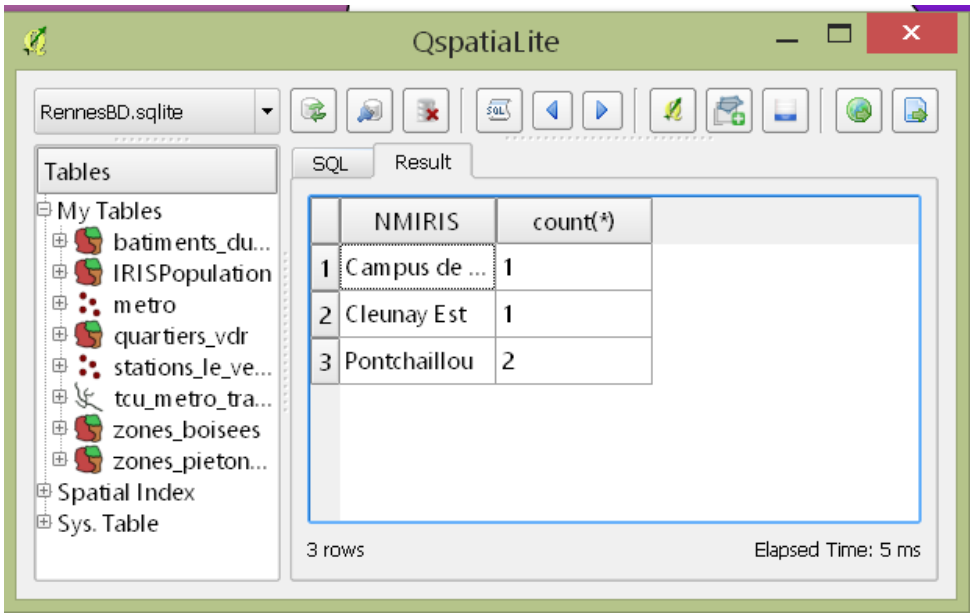
- **Exercice 1.**
  - Sélectionner les quartiers de plus de 4 km2, qui possèdent au moins 5 stations de vélos stars (5 IRIS)





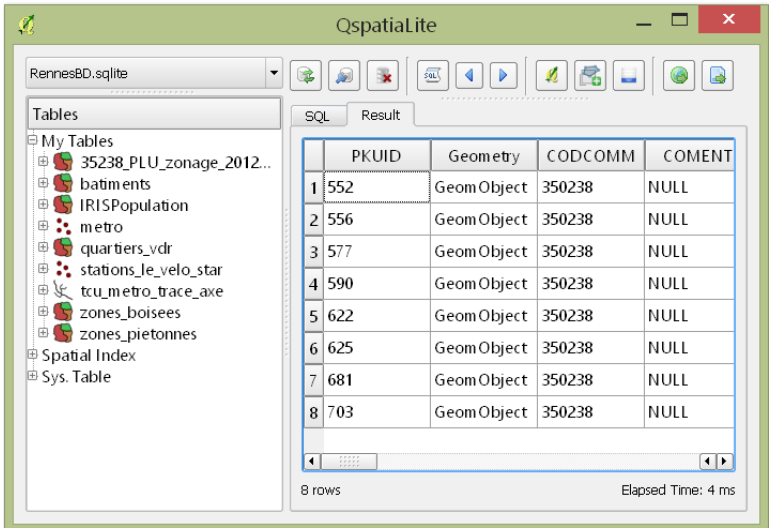
Aller plus loin

- **Exercice 2.**
  - Calculer le nombre d'IRIS de moins d'1 km2 et possédant une station de métro (3 IRIS)



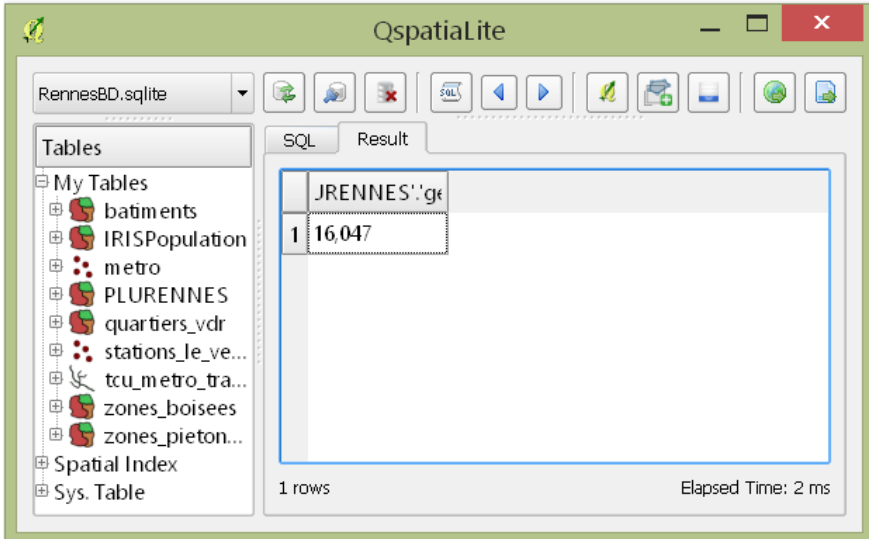
Aller plus loin

- **Exercice 3.**
  - Afficher les parcelles zonées UD» du quartier Gare Sud» (8 parcelles)



Aller plus loin

- **Exercice 4.**
  - Calculer la surface totale des parcelles zonées UD du quartier Gare Sud»

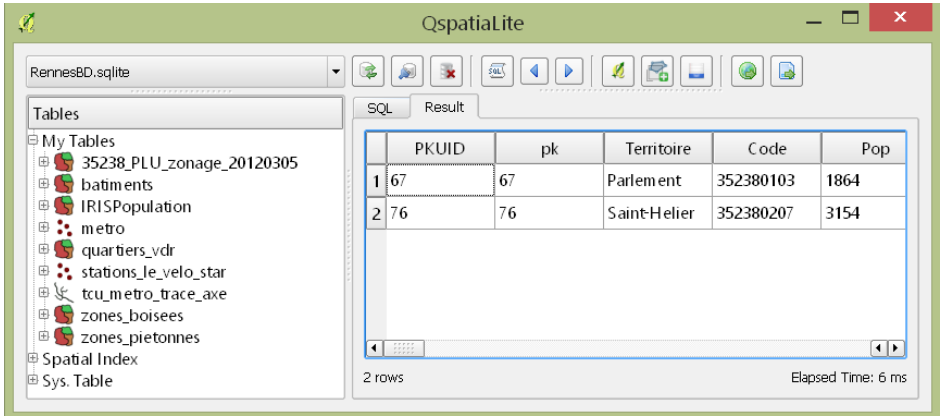


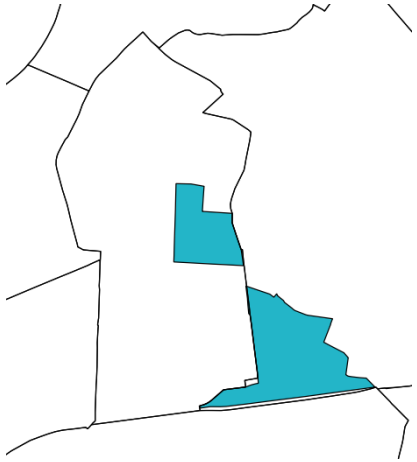
Aller plus loin

- **Exercice 5.**
  - Sélectionner les IRIS du quartier centre» possédant des stations de vélos équipées d'un terminal de paiement et connectée à un métro

→ Un seul IRIS dans la réalité

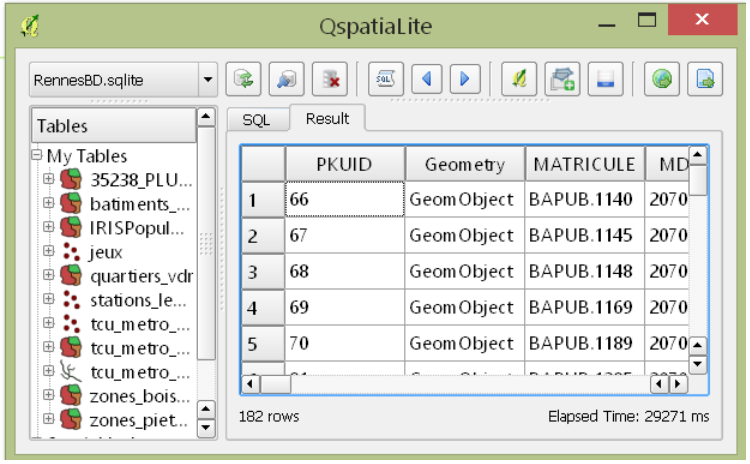
- on a un problème topologique entre les limites de quartiers et celles des IRIS, ce qui induit l'erreur

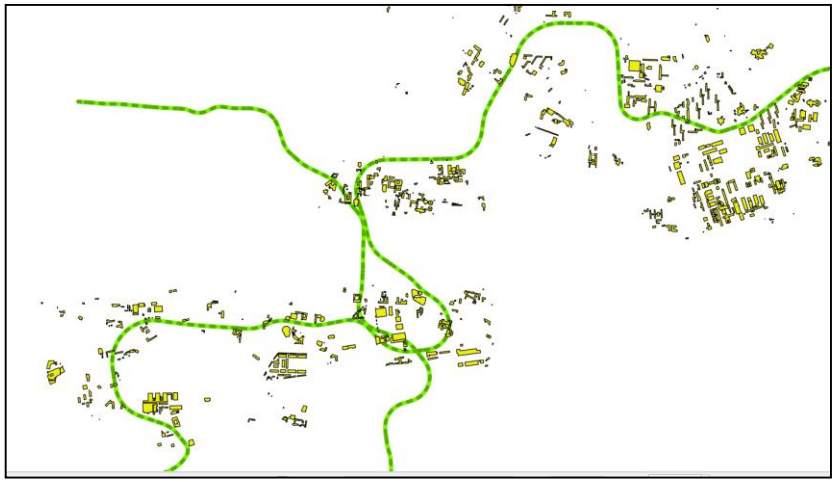




Aller plus loin

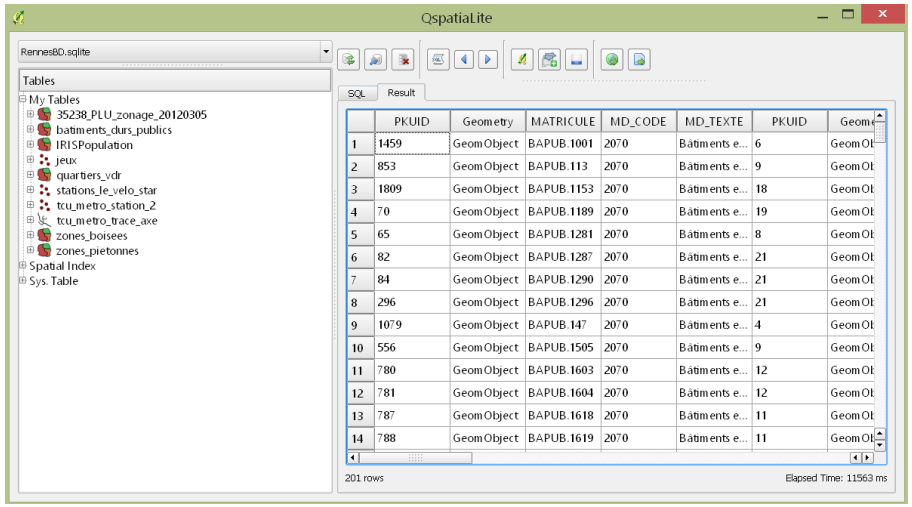
- **Exercice 6.**
  - Sélectionner les bâtiments des IRIS qui contiennent des stations de métro de la ligne B, qui possèdent une station de vélos et une aire de jeux (182 bâtiments)

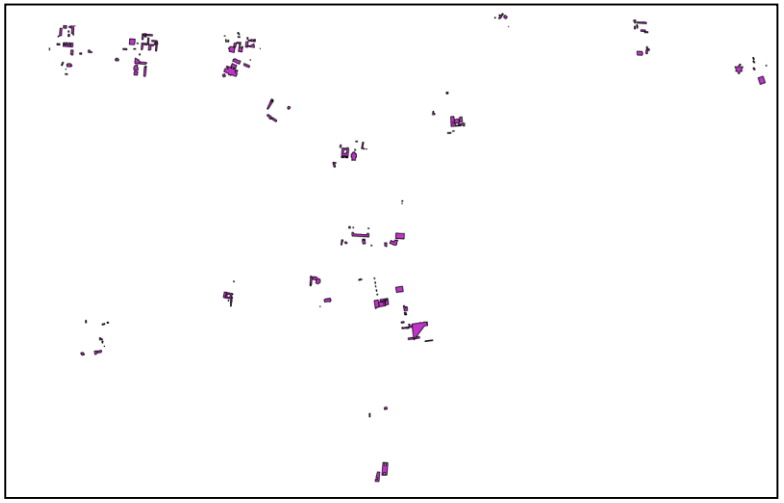




Aller plus loin

- **Exercice 7.**
  - Sélectionner les bâtiments publics situés dans une zone U» (utiliser la condition like valeur%» ici like U%») du PLU et se trouvant à moins de 200 mètres d'une station de métro (201 bâtiments)





Opérations de recouvrement

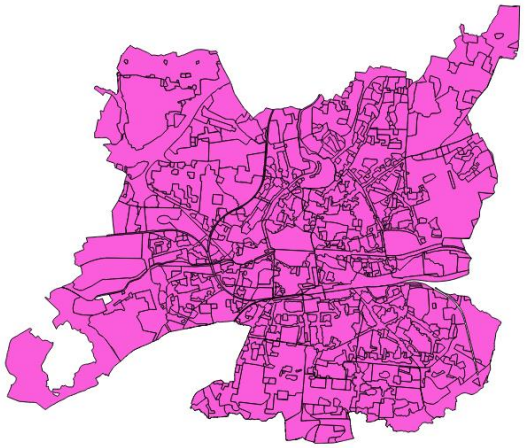
Opérateurs de recouvrement

Function	Syntax
Intersection	Intersection(geom1 *Geometry* , geom2 *Geometry*) : *Geometry* ST_Intersection(geom1 *Geometry* , geom2 *Geometry*) : *Geometry*
Difference	Difference(geom1 *Geometry* , geom2 *Geometry*) : *Geometry* ST_Difference(geom1 *Geometry* , geom2 *Geometry*) : *Geometry*
GUnion	GUnion(geom1 *Geometry* , geom2 *Geometry*) : *Geometry* OpenGis name for this function is **Union()**, but it conflicts with an SQLite reserved keyword ST_Union(geom1 *Geometry* , geom2 *Geometry*) : *Geometry*
GUnion	GUnion(geom *Geometry*) : *Geometry* ST_Union(geom *Geometry*) : *Geometry*
SymDifference	SymDifference(geom1 *Geometry* , geom2 *Geometry*) : *Geometry* ST_SymDifference(geom1 *Geometry* , geom2 *Geometry*) : *Geometry*
Buffer	Buffer(geom *Geometry* , dist *Double precision*) : *Geometry* ST_Buffer(geom *Geometry* , dist *Double precision*) : *Geometry*
ConvexHull	ConvexHull(geom *Geometry*) : *Geometry* ST_ConvexHull(geom *Geometry*) : *Geometry*

Découpage (recouvrement)

- Découper la couche du PLU par rapport à celle des quartiers

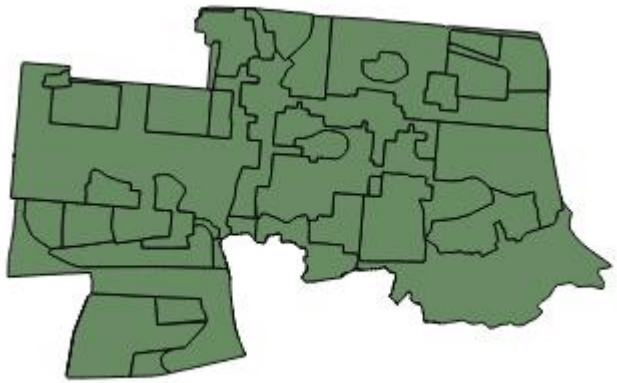
```
select PLU.id, intersection(PLU.geom, quartier.geom) from PLU, quartier
where intersects(PLU.geom, quartier.geom)
```



Découpage (recouvrement)

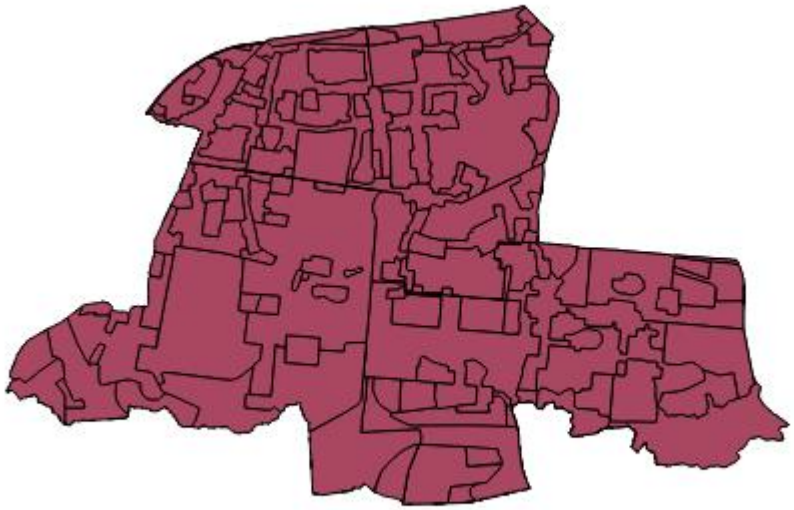
- Créer une couche du PLU pour l'emprise du quartier du Blosne

```
select PLU.id, PLU.libelong as Type, intersection(PLU.geom, quartier.geom) as geom
from PLU, quartier
where intersects(PLU.geom, quartier.geom) and quartier.nmquart='LE BLOSNE'
```



Découpage (recouvrement)

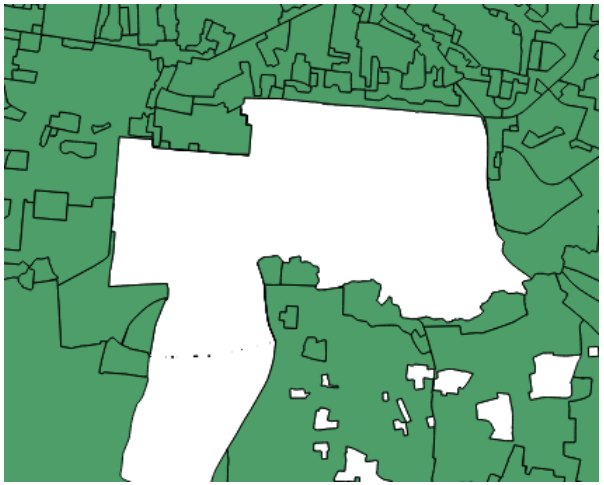
- Créer une couche du PLU pour l'emprise du quartier du Blosne, de Brequigny et de Sud Gare



Différence

- Supprimer de la couche PLU l'emprise du quartier du Blosne

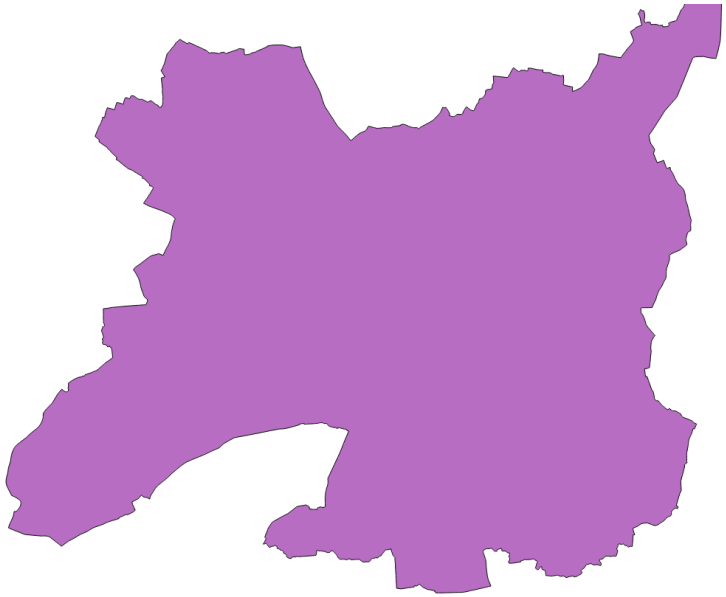
```
select PLU.id, PLU.libelong as Type, difference(PLU.geom, quartier.geom) as
geom
from PLU, quartier
 where quartier.nmquart='LE BLOSNE'
```



Dissolve

- Fusionner la couche des quartiers pour obtenir une couche de l'emprise de la ville de Rennes

```
Select id, st_unaryunion(st_collect(quartiers.geom)) as geom
From quartiers
Group by champdissolve
```



Exercices

- Tester UNION

- Tester Symdifference

Aller plus loin / Solutions

```
select *
from quartiers_vdr
where (Area( quartiers_vdr.'Geometry' ) / 1000000) > 4 and
quartiers_vdr.'Nbvelo' >10
```

```
SELECT IRISPopulation.'NMIRIS', count(*)
FROM 'IRISPopulation', 'metro'
WHERE (Area('IRISPopulation'.'Geometry') / 1000000) > 1 and
contains('IRISPopulation'.'Geometry','metro'.'geom')
group by IRISPopulation.'NMIRIS'
```

```
SELECT *
FROM 35238_PLU_zonage_20120305, quartiers_vdr
WHERE 35238_PLU_zonage_20120305.'TYPE'=UD and within(
35238_PLU_zonage_20120305.'Geometry', quartiers_vdr.'Geometry') and
quartiers_vdr.'NUQUART' = 8
```

```
SELECT sum(area(35238_PLU_zonage_20120305.'Geometry' )/1000000)
FROM 35238_PLU_zonage_20120305 , quartiers_vdr
WHERE 35238_PLU_zonage_20120305 .'type' =UD and within(
35238_PLU_zonage_20120305.'Geometry', quartiers_vdr.'Geometry') and quartiers_vdr.'NUQUART'
= 8
```

Aller plus loin / Solutions

```
SELECT *
FROM IRISPopulation, quartiers_vdr, stations_le_velo_star
WHERE quartiers_vdr.'NUQUART'=1 and contains(quartiers_vdr.'Geometry',
centroid( IRISPopulation.'Geometry')) and contains( IRISPopulation.'Geometry',
stations_le_velo_star.'Geometry') and stations_le_velo_star.'METRO' = oui and
stations_le_velo_star.'TPE' =oui
```

```
SELECT *
FROM batiments_durs_publics, IRISPopulation, tcu_metro_station_2,
stations_le_velo_star, jeux
WHERE within(batiments_durs_publics.'Geometry',IRISPopulation.'Geometry') and
contains(IRISPopulation.'Geometry',tcu_metro_station_2.'Geometry') and
tcu_metro_station_2.'LIGNE' =b and
contains(IRISPopulation.'Geometry',stations_le_velo_star.'Geometry') and contains(
IRISPopulation.'Geometry',jeux.'Geometry')
group by batiments_durs_publics.'PKUID'
```

Aller plus loin

```
select *
from batiments_durs_publics, tcu_metro_station_2,
35238_PLU_zonage_20120305
where within( batiments_durs_publics.'Geometry',
35238_PLU_zonage_20120305.'Geometry') and
35238_PLU_zonage_20120305.'TYPE' LIKE 'U%' and within(
batiments_durs_publics.'Geometry',buffer(
tcu_metro_station_2.'Geometry',200))
group by batiments_durs_publics.'MATRICULE'
```

Solutions

- Exercices présidentielles

```
create view FN as select NUMERO_LIEU, NOM_LIEU,POURCENTAGE
from elections where CANDIDAT='LE PEN Marine'
```

```
create view Macron as select NUMERO_LIEU, NOM_LIEU,POURCENTAGE from
elections where CANDIDAT='MACRON Emmanuel'
```

```
select FN.NUMERO_LIEU as Solutions, FN.NOM_LIEU, 'LE PEN',
FN.POURCENTAGE as ScoreFN,
'MACRON', Marcon.POURCENTAGE as ScoreLREM
from FN, Marcon WHERE FN.NUMERO_LIEU==Marcon.NUMERO_LIEU
order by Solutions asc
```

- **Boris.mericskay@uhb.fr**