

## **UE 702-2 : Théorie et ingénierie des bases de données**

**Sujet :** Système de Gestion et Analyse des Données Urbaines et Environnementales de la métropole de Rennes

### **I. Description de la base de données**

Pour ce projet de SGBD, nous avons choisi de concevoir une base de données géographique reflétant la gestion des données pour Rennes métropole, incluant divers aspects liés aux infrastructures, espaces naturels, services publics, et aménagements. Cette base de données pourrait être utilisée dans des scénarios réels par des collectivités locales, des urbanistes ou des organismes environnementaux.

Nous avons structuré notre projet autour de **12 tables principales**, dont plusieurs possèdent un champ géométrique pour intégrer des données spatiales en exploitant le système PostGIS. Ces tables sont conçues pour répondre à des besoins variés allant de la gestion des écoles primaires et des parkings vélo à l'analyse des bassins versants et de l'occupation des sols.

#### **1. Structure et choix des tables**

Les tables suivantes constituent le cœur de notre base de données :

1. **limite\_communale** \*: Représente les communes, incluant leurs limites géographiques, leur nom, leur code postal et leur code INSEE.
2. **ecole\_primaire** \*: Contient les informations sur les écoles primaires, leur localisation géographique, leur nombre d'élèves et leur commune de rattachement.
3. **parcking\_velo** \*: Gère les données sur les parkings vélo, notamment leur position et leur capacité.
4. **toilette\_publique** \*: Détaille les toilettes publiques, avec des informations sur leur emplacement, leur type et leur commune.
5. **amphibien** \*: Représente les données sur les amphibiens recensés, intégrant leur localisation géographique.
6. **types\_wc** : Regroupe les différents types de toilettes publiques existants.

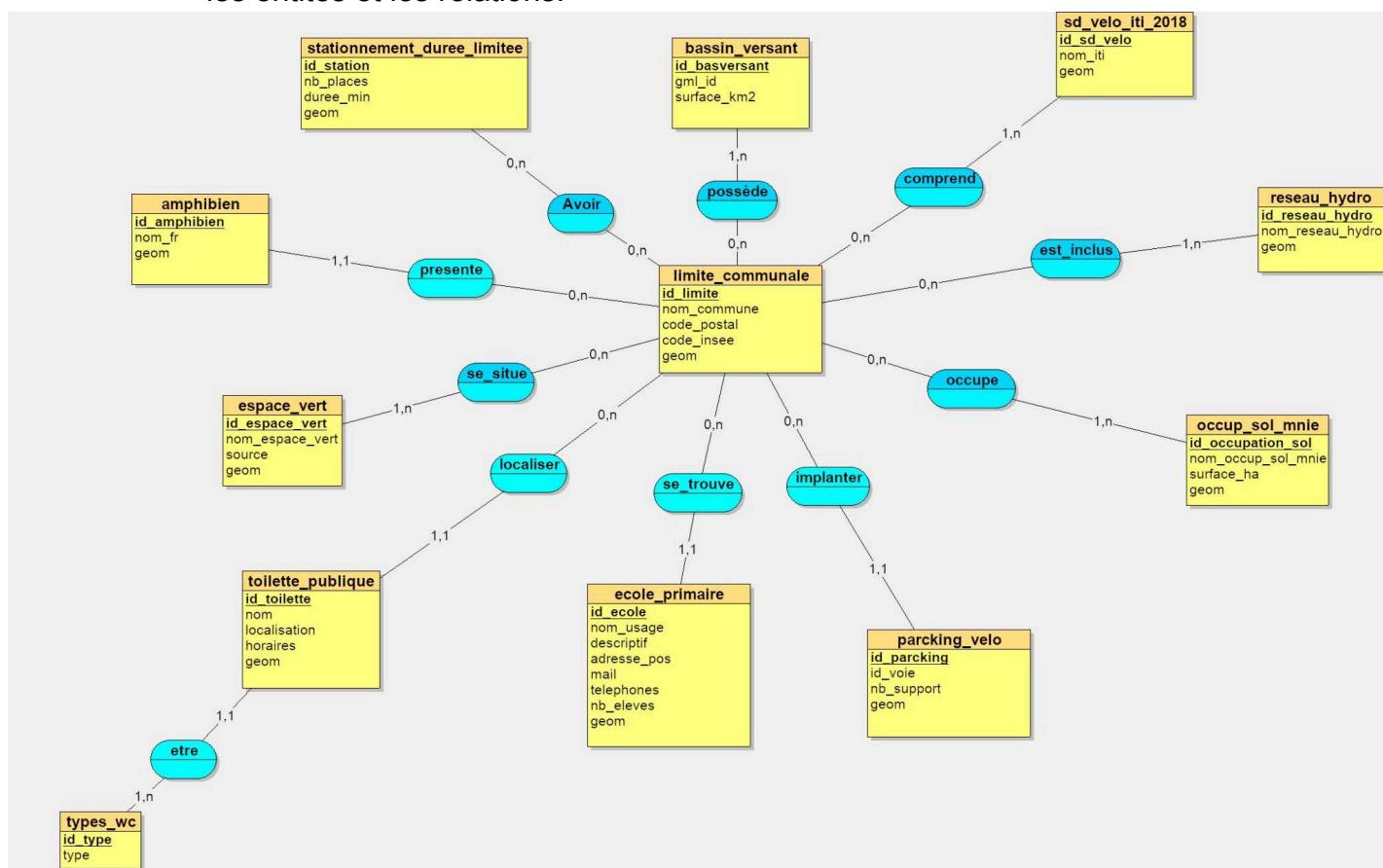
7. **espace\_vert** \*: Cartographie les espaces verts, en tenant compte de leurs surfaces et de leur source.
8. **occup\_sol\_mnie** \*: Décrit l'occupation des sols, classifiée par type et surface.
9. **reseau\_hydro** \*: Représente les réseaux hydrologiques sous forme de géométries linéaires.
10. **bassin\_verseant** : Gère des informations textuelles sur les bassins versants.
11. **sd\_velo\_itl\_2018** \*: Décrit les itinéraires vélo sous forme de géométries linéaires.
12. **stationnement\_duree\_limitee** \*: Gère les données des zones de stationnement limité en durée, incluant leur capacité et localisation.

(\* : tables contenant des champs géométriques)

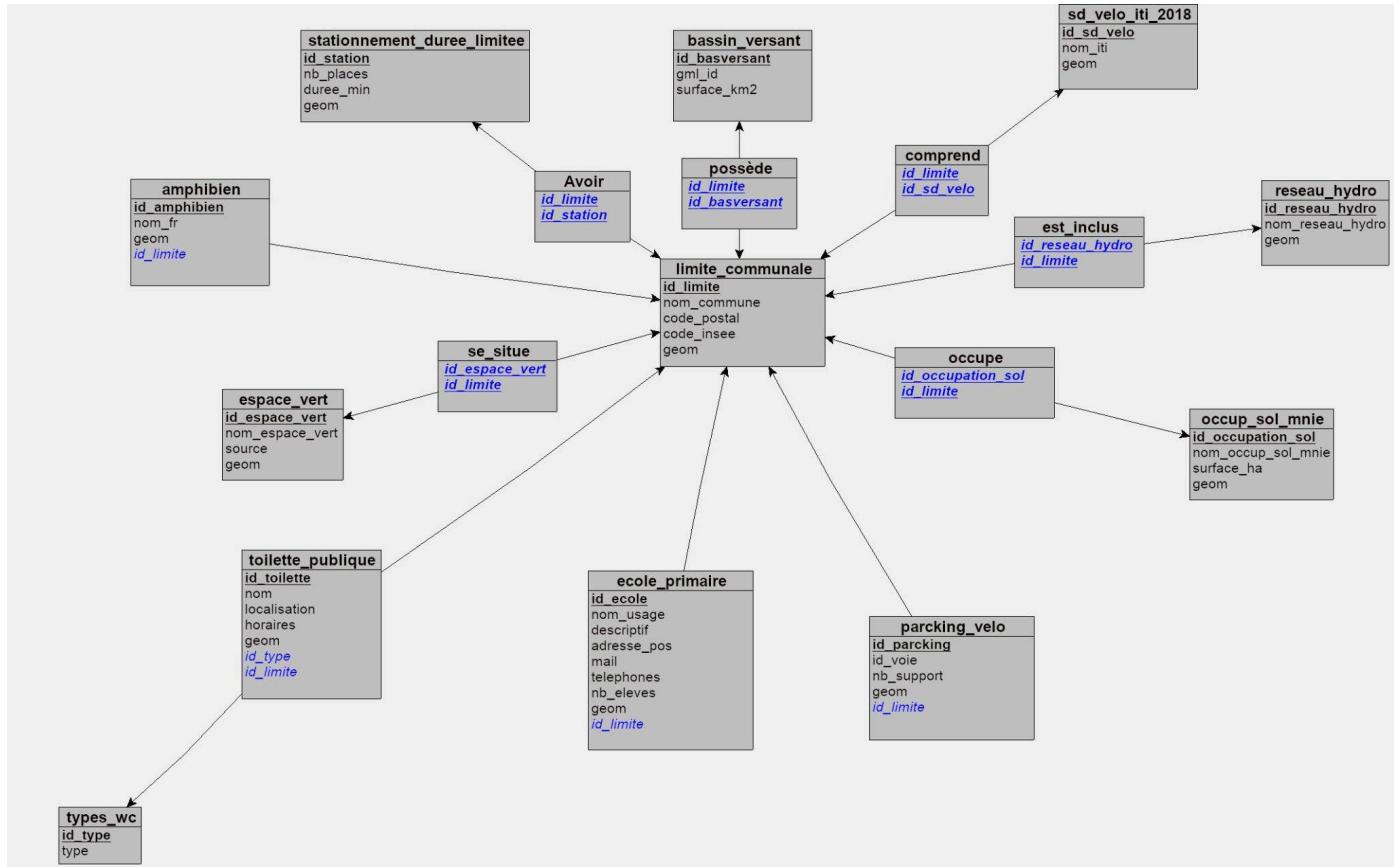
## 2. Méthodologie

Pour la création de notre base, nous avons procédé en plusieurs étapes :

- **Collecte et préparation des données géographiques :**
  - Les limites communales, espaces verts, réseaux hydrologiques et emprises des infrastructures ont été obtenus à partir de l'open data de Rennes métropole.
  - Les géométries ont été manipulées dans QGIS, puis exportées au format compatible avec PostGIS.
- **Modélisation des relations :**
  - Nous avons utilisé un **Modèle Conceptuel de Données (MCD)** pour définir les entités et les relations.



- Ces relations ont ensuite été transformées en un **Modèle Logique de Données (MLD)**, en s'assurant de la cohérence des cardinalités et des contraintes d'intégrité.



## Table reseau\_hydro

- **Champs** : `id_reseau_hydro` (clé primaire), `nom_reseau_hydro`, `geom` (type `MultiPolygon`).
- **Justification de la cardinalité :**
  - Cette table est liée à la limite communale.

## Table espace\_vert

- **Champs** : `id_espace_vert` (clé primaire), `nom_espace_vert`, `source`, `geom` (type `MultiPolygon`).
- **Justification de la cardinalité :**
  - Comme **reseau\_hydro**, cette table est liée à la table **limite\_communale**. Une commune peut avoir un réseau ou plusieurs, et un réseau est dans une ou plusieurs communes (à cheval entre plusieurs communes).

## Table occup\_sol\_mnie

- **Champs** : `id_occupation_sol` (clé primaire), `nom_occup_sol_mnie`, `surface_ha`, `geom` (type `MultiPolygon`).
- **Justification de la cardinalité** :
  - Cette table est aussi liée à la limite communale.

## Table limite\_communale

- **Champs** : `id_limite` (clé primaire), `nom_commune`, `code_postal`, `code_insee`, `geom` (type `MultiPolygon`).
- **Justification de la cardinalité** :
  - **1 à n avec la table `ecole_primaire`, 1 à n avec la table `parcking_velo`, 1 à n avec la table `amphibien`, et 1 à n avec la table `toilette_publique`, 1 à n avec la table `bassin_versant`, 1 à n avec la table `école_primaire`, 1 à n avec la table `espace_vert`, 1 à n avec la table `occupation_sol_mnie`, 1 à n avec la table `réseau_hydro`, 1 à n avec la table `sd_vélo_it_2018`, 1 à n avec la table `stationnement_durée_limitée` :**
    - Chaque limite communale peut avoir plusieurs écoles, parkings, espaces pour amphibiens et toilettes publiques.

## Table bassin\_versant

- **Champs** : `id_basversant` (clé primaire), `gml_id`, `surface_km2`.
- **Justification de la cardinalité** :
  - Cette table est liée à la limite communale.

## Table ecole\_primaire

- **Champs** : `id_ecole` (clé primaire), `nom_usage`, `descriptif`, `adresse_pos`, `mail`, `telephones`, `nb_eleves`, `geom` (type `MultiPoint`), `id_limite` (clé étrangère).
- **Justification de la cardinalité** :
  - **1 à 1 avec la table `limite_communale`** : Une école est située dans une commune spécifique, représentée par une limite communale.

## Table parcking\_velo

- **Champs** : `id_parcaking` (clé primaire), `id_voie`, `nb_support`, `geom` (type `Point`), `id_limite` (clé étrangère).
- **Justification de la cardinalité** :
  - **1 à n avec la table limite\_communale** : Un parking vélo est lié à une commune spécifique, représentée par une limite communale.

## Table sd\_velo\_iti\_2018

- **Champs** : `id_sd_velo` (clé primaire), `nom_iti`, `geom` (type `MultiLineString`).
- **Justification de la cardinalité** :
  - **1 à n avec la table limite\_communale** : Une voie cyclable est liée à une commune spécifique, représentée par une limite communale.

## Table stationnement\_duree\_limitee

- **Champs** : `id_station` (clé primaire), `nb_places`, `duree_min`, `geom` (type `LineString`).
- **Justification de la cardinalité** :
  - **1 à 1 avec la table limite\_communale** : Une place de stationnement est liée à une commune spécifique, représentée par une limite communale.

## Table types\_wc

- **Champs** : `id_type` (clé primaire), `type`.
- **Justification de la cardinalité** :
  - **1 à n avec la table toilette\_publique** : Un type de toilettes peut être utilisé par plusieurs toilettes publiques.

## Table amphibien

- **Champs** : `id_amphibien` (clé primaire), `nom_fr`, `geom` (type `MultiPoint`), `id_limite` (clé étrangère).
- **Justification de la cardinalité** :
  - **1 à 1 avec la table limite\_communale** : Un amphibien est situé dans une commune spécifique, représentée par une limite communale.

## Table toilette\_publique

- **Champs** : `id_toilette` (clé primaire), `nom`, `localisation`, `horaires`, `geom` (type `Point`), `id_type` (clé étrangère), `id_limite` (clé étrangère).

- **Justification de la cardinalité :**
- **1 à 1 avec la table limite\_communale** : Une toilette publique est située dans une commune spécifique, représentée par une limite communale.
- **1 à 1 avec la table types\_wc** : Chaque toilette publique est liée à un type spécifique de toilettes.

### **3. Choix des types de données :**

Les champs géométriques utilisent des types adaptés tels que `MultiPolygon`, `MultiPoint`, ou `LineString`, selon les spécificités des entités.

Les champs textuels et numériques ont été optimisés pour garantir une utilisation efficace de l'espace et des performances, avec des types comme `CHAR` ou `VARCHAR` pour les identifiants et `SMALLINT` ou `DECIMAL` pour les valeurs numériques.

## **II . Exploitation de la base de données :**

### **Requêtes de modification de tables :**

#### **1.1. Ajouter une occupation du sol dans une commune**

```
INSERT INTO occup_sol_mnie (id_occupation_sol, nom_occup_sol_mnie, surface_ha, geom)
VALUES ( 601, 'Zone agricole',15.75,
ST_SetSRID(ST_Multi(ST_MakePolygon(ST_GeomFromText("LINESTRING(498000
6498000, 498100 6498000, 498100 6498100, 498000 6498100, 498000
6498000)'))),2154));
```

#### **1.2. Suppression correspondante**

```
DELETE FROM occup_sol_mnie
WHERE id_occupation_sol = 601 ;
```

#### **2.1. Ajouter un amphibiens géoréférencé proche d'un cours d'eau**

```
INSERT INTO amphibien (id_amphibien, nom_fr, geom, id_limite)
VALUES (501, 'Grenouille verte', ST_SetSRID(ST_Multi(ST_MakePoint(500500, 6500500)), 2154), 10);
```

## **2.2. Suppression correspondante**

```
DELETE FROM amphibien  
WHERE id_amphibien = 501;
```

## **3.1. Ajouter un parking vélo près d'une école**

```
INSERT INTO parcking_velo (id_parcaking, id_voie, nb_support, geom, id_limite)  
VALUES (401, 5, 10, ST_SetSRID(ST_MakePoint(500050, 6500050), 2154), 10);
```

### **3.2 Suppression correspondante :**

```
DELETE FROM parcking_velo  
WHERE id_parcaking = 401;
```

## **4.1. Ajouter une toilette publique dans la commune de Rennes (id\_limite=33)**

```
INSERT INTO toilette_publique (id_toilette, nom, localisation, horaires, geom,  
id_type, id_limite)  
VALUES (201, 'Toilettes du Parc', 'Parc Central', '08:00 - 20:00',  
ST_SetSRID(ST_MakePoint(499000, 6499000), 2154), 1, 33);
```

### **4.2. Suppression correspondante**

```
DELETE FROM toilette_publique  
WHERE id_toilette = 201;
```

# Requêtes SELECT

## 1. Liste des écoles primaires

```
SELECT id_ecole, nom_usage, nb_eleves  
FROM ecole_primaire;
```

## 2. Liste des toilettes publiques dans une commune spécifique

```
SELECT nom, localisation, horaires  
FROM toilette_publique  
WHERE id_limite = 33;
```

## 3. Identifier les amphibiens à proximité des cours d'eau

```
SELECT a.nom_fr AS amphibien, rh.nom_reseau_hydro AS cours_d_eau  
FROM amphibien a  
JOIN reseau_hydro rh ON ST_DWithin(a.geom, rh.geom, 300);
```

## 4. Surface totale des espaces verts

```
SELECT SUM(ST_Area(geom)) AS surface_totale_m2  
FROM espace_vert;
```

## 5. Nombre d'écoles primaires par commune

```
SELECT l.nom_commune, COUNT(e.id_ecole) AS nb_ecoles  
FROM limite_communale l  
LEFT JOIN ecole_primaire e ON l.id_limite = e.id_limite  
GROUP BY l.nom_commune  
ORDER BY nb_ecoles DESC;
```

## 6. Vérifier si une école est dans une commune

```
SELECT e.nom_usage, l.nom_commune  
FROM ecole_primaire e  
JOIN limite_communale l  
ON ST_Contains(l.geom, e.geom);
```

## **7. Nombre de toilettes publiques par type**

```
SELECT t.type, COUNT(tp.id_toilette) AS nb_toilettes
FROM types_wc t
LEFT JOIN toilette_publique tp ON t.id_type = tp.id_type
GROUP BY t.type
ORDER BY nb_toilettes DESC;
```

## **8. les 5 communes avec les plus grandes surfaces occupées par des zones d'occupation du sol**

```
SELECT lc.nom_commune AS commune,
       SUM(ST_Area(os.geom)) / 10000 AS surface_totale_ha
  FROM limite_communale lc
JOIN occup_sol_mnie os ON ST_Intersects(lc.geom, os.geom)
 GROUP BY lc.nom_commune
 ORDER BY surface_totale_ha DESC
LIMIT 5;
```

## **9. Lister les occupations des sols dans la commune de Rennes**

```
SELECT os.id_occupation_sol, os.nom_occup_sol_mnie, os.surface_ha
  FROM occup_sol_mnie os
JOIN limite_communale lc
  ON ST_Contains(lc.geom, os.geom)
 WHERE lc.nom_commune = 'Rennes';
```

## **10. Moyenne des élèves par commune avec école**

```
SELECT l.nom_commune, AVG(e.nb_eleves) AS moyenne_eleves
  FROM limite_communale l
JOIN ecole_primaire e ON l.id_limite = e.id_limite
 GROUP BY l.nom_commune
 HAVING AVG(e.nb_eleves) > 50
 ORDER BY moyenne_eleves DESC;
```

6. Occupations des sols partagées entre plusieurs communes

## **11. Occupations des sols partagées entre plusieurs communes**

```
SELECT os.nom_occup_sol_mnie, COUNT(DISTINCT lc.nom_commune) AS nb_communes
FROM occup_sol_mnie os
JOIN limite_communale lc
ON ST_Intersects(os.geom, lc.geom)
GROUP BY os.nom_occup_sol_mnie
HAVING COUNT(DISTINCT lc.nom_commune) > 1;
```

## **12. Visualiser les 5 premières limites communales avec les plus grands nombre de réseaux hydrologiques traversants**

```
SELECT lc.nom_commune, COUNT(rh.id_reseau_hydro) AS nb_reseaux
FROM limite_communale lc
LEFT JOIN reseau_hydro rh
ON ST_Intersects(lc.geom, rh.geom)
GROUP BY lc.nom_commune
ORDER BY nb_reseaux DESC
LIMIT 5;
```

## **13. Amphibiens observés dans plusieurs communes**

```
SELECT a.nom_fr, COUNT(DISTINCT lc.nom_commune) AS nb_communes
FROM amphibien a
JOIN limite_communale lc
ON ST_Intersects(a.geom, lc.geom)
GROUP BY a.nom_fr
HAVING COUNT(DISTINCT lc.nom_commune) > 1
ORDER BY nb_communes DESC;
```

## **14. Statistiques des distances entre écoles et toilettes publiques**

```
SELECT MIN(ST_Distance(ep.geom, tp.geom)) AS min_distance,
       AVG(ST_Distance(ep.geom, tp.geom)) AS avg_distance,
       MAX(ST_Distance(ep.geom, tp.geom)) AS max_distance
FROM ecole_primaire ep
CROSS JOIN toilette_publique tp;
```