

---

# RAPPORT MINI PROJET BDD

---

```
/-----\
|          Bienvenue !          |
| Connexion à la base de données réussite ! |
|          Que voulez-vous faire ?          |
|-----/

MENU PRINCIPAL

1 - Afficher la liste des régions
2 - Afficher la liste des départements
3 - Afficher les données d'une région au choix
4 - Afficher les données d'un département et un thème :
   social ou environnemental
5 - Afficher la liste des départements dont type d'énergie
   (éolien, photovoltaïque, autre) a augmenté entre
   les 2 années de référence, dans l'ordre décroissant
```

Réalisé par Julie **PRATX**

M1 Bioinformatique Biocomp

Le 19 Avril 2020

# INTRODUCTION

Il existe plusieurs Systèmes de Gestion de Base de Données (SGBD) gratuits qui ont chacun leurs avantages et leurs inconvénients [1].

## **SQLite :**

- + module natif de Python
- + vise à remplacer l'utilisation des fichiers .txt
- + très utilisé sur les téléphones Android
- + application de traitement de base de données légère et transportable (sur une clé USB, par exemple) : DB Browser
- ne peut pas gérer énormément de données (ne peut pas dépasser le gigaoctet de données)
- ne gère pas la connexion assez importante (plusieurs personnes en même temps effectuant la même recherche)

## **MySQL :**

- + solution la plus répandue (correspond, pratiquement, à tous les projets utilisés dans le monde)
- + gère la connexion simultanée (exemple, les forums, les news,...)
- + performant pour gérer une base de plusieurs gigaoctets de données

## **PostgreSQL** semblable à *MySQL* mais :

- + optimisé pour des bases de données très très importante (de l'ordre du téraoctet de données ou plus) : exemple, une base de données d'un CHU.

En conclusion, il n'existe pas un meilleur système de base de données, mais on préférera un système plutôt qu'un autre suivant le projet que l'on voudra réaliser.

Ma base de données fait quelques kilooctets de données, mon choix, pour ce mini projet, s'est donc porté sur SQLite.

# CONCEPTION DU PROGRAMME

**Partie\_1.py** réalisé à l'aide d'un cours en ligne [2] et correspond à :

- la création de la base de données nommée : base.db (l'utilisateur peut changer le nom dans le programme)
- la création des tables DEPARTEMENTS et REGIONS
- l'ajout des données issues des fichiers csv aux tables [3]
- la lecture des données d'un fichier xls [4]
- l'ajout des tables SOCIALREG, SOCIALDEP et ENVIRONNEMENTDEP
- la conversion xls → csv
- l'ajout des données dans les nouvelles tables [3]

Pour l'utilisation des données présentent dans le fichier xls, j'ai préféré le modifier « esthétiquement » pour le convertir en fichier csv exploitable par la base de données.

**Partie\_2.py** est le programme en lui-même. Il contient :

- un menu avec interaction directe de l'utilisateur
- la connexion automatique à la base de données
- les requêtes SQL (détaillées dans la partie suivante du rapport) [5]
- la déconnexion automatique de la base de données à la fermeture du programme

# SCHÉMA DE LA BASE DE DONNÉES

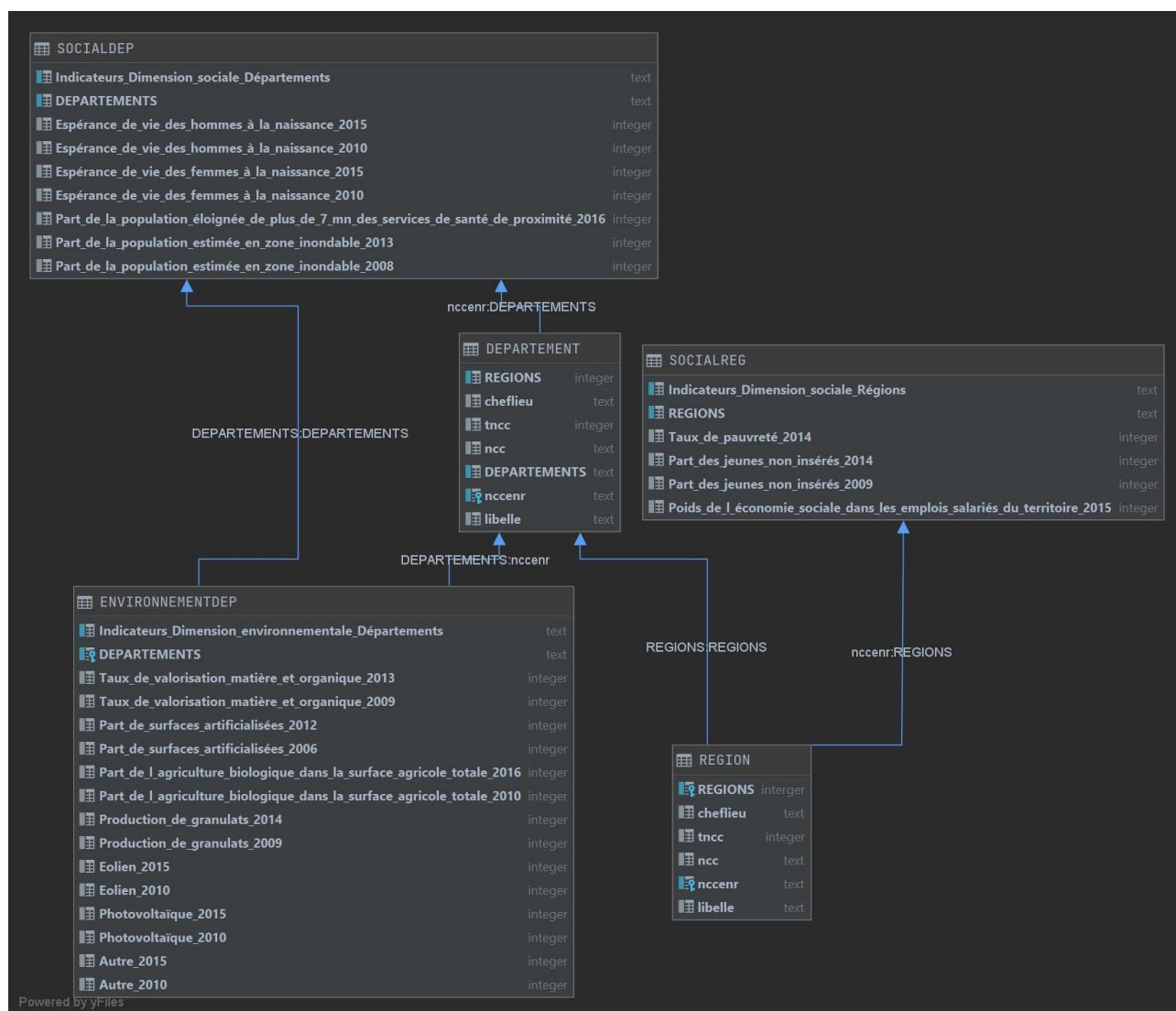


Illustration 1: Schéma de la base de données

Ce schéma (**Illustration 1**) a été réalisé à l'aide du programme de gestion de base de données DataGrip.

Les **clés primaires** et les **clés étrangères** de chaque table sont :

- ➔ REGION (**REGIONS**, nccnr)
- ➔ DEPARTEMENT (**nccnr**, REGIONS)
- ➔ ENVIRONNEMENTDEP (**DEPARTEMENTS**, DEPARTEMENTS)
- ➔ SOCIALREG (**Indicateurs\_Dimension\_sociale\_Régions**, REGIONS)
- ➔ SOCIALDEP (**Indicateurs\_Dimension\_sociale\_Départements**, DEPARTEMENTS)

# REQUÊTES SQL

- afficher la liste des régions

```
5 SELECT nccenr FROM REGION;
```

- afficher la liste des départements

```
9 SELECT nccenr FROM DEPARTEMENT;
```

- demander à l'utilisateur de choisir une région et afficher les données de la région choisie

```
13 SELECT R.nccenr,  
14         S.Taux_de_pauvreté_2014,  
15         S.Part_des_jeunes_non_insérés_2014,  
16         S.Part_des_jeunes_non_insérés_2009,  
17         S.Poids_de_l'économie_sociale_dans_les_emplois_salariés_du_territoire_2015  
18 FROM REGION R join SOCIALREG S on R.nccenr = S.REGIONS  
19 WHERE nccenr LIKE \"%" + choix + "%\";
```

*choix* correspond au choix de l'utilisateur dans le programme Python.

- demander à l'utilisateur de choisir un département et un thème : social ou environnemental, et afficher les données demandées pour le département choisi

Social :

```
107 SELECT Espérance_de_vie_des_hommes_à_la_naissance_2015,  
108         Espérance_de_vie_des_hommes_à_la_naissance_2010,  
109         Espérance_de_vie_des_femmes_à_la_naissance_2015,  
110         Espérance_de_vie_des_femmes_à_la_naissance_2010,  
111         Part_de_la_population_éloignée_de_plus_de_7_mn_des_services_de_santé_de_proximité_2016,  
112         Part_de_la_population_estimée_en_zone_inondable_2013,  
113         Part_de_la_population_estimée_en_zone_inondable_2008  
114 FROM SOCIALDEP  
115 WHERE DEPARTEMENTS LIKE \"%" + choix_Dep + "%\";
```

Environnemental :

```
123 SELECT Taux_de_valorisation_matière_et_organique_2013,  
124         Taux_de_valorisation_matière_et_organique_2009,  
125         Part_de_surfaces_artificialisées_2012,  
126         Part_de_surfaces_artificialisées_2006,  
127         Part_de_l'agriculture_biologique_dans_la_surface_agricole_totale_2016,  
128         Part_de_l'agriculture_biologique_dans_la_surface_agricole_totale_2010,  
129         Production_de_granulats_2014,  
130         Production_de_granulats_2009,  
131         Eolien_2015,  
132         Eolien_2010,  
133         Photovoltaïque_2015,  
134         Photovoltaïque_2010,  
135         Autre_2015,  
136         Autre_2010  
137 FROM ENVIRONNEMENTDEP  
138 WHERE DEPARTEMENTS LIKE \"%" + choix_Dep + "%\";
```

- demander à l'utilisateur de choisir un type d'énergie (éolien, photovoltaïque, autre) et en fonction de ce choix retourner la liste des départements où la part de cette énergie a augmenté entre les deux années de référence, classés de la plus forte augmentation à la plus faible

Éolien :

```
145 SELECT DEPARTEMENTS
146 FROM ENVIRONNEMENTDEP
147 WHERE \"Eolien_2015\" - \"Eolien_2010\" > 0
148 ORDER BY \"Eolien_2015\" - \"Eolien_2010\" DESC;
```

Photovoltaïque :

```
154 SELECT DEPARTEMENTS
155 FROM ENVIRONNEMENTDEP
156 WHERE \"Photovoltaïque_2015\" - \"Photovoltaïque_2010\" > 0
157 ORDER BY \"Photovoltaïque_2015\" - \"Photovoltaïque_2010\" DESC;
```

Autre :

```
164 SELECT DEPARTEMENTS
165 FROM ENVIRONNEMENTDEP
166 WHERE \"Autre_2015\" - \"Autre_2010\" > 0
167 ORDER BY \"Autre_2015\" - \"Autre_2010\" DESC;
```

1. Quels sont les départements dont la région a eu une production de granulats supérieure à 25 000 000 tonnes en 2014 ?

J'ai préféré utiliser la technique de la sous requête plutôt que celle de la division [6].

```
20 SELECT D.libelle
21 FROM (
22     SELECT sr.REGIONES, sr.total
23     FROM(
24         SELECT REGIONES, SUM(Production_de_granulats_2014) total
25         FROM DEPARTEMENT D
26         INNER JOIN ENVIRONNEMENTDEP E on D.nccenr = E.DEPARTEMENTS
27         GROUP BY REGIONES) as sr
28     WHERE sr.total > 25_000_000
29     AND sr.total !=0) as sv
30 INNER JOIN DEPARTEMENT D on D.REGIONES = sv.REGIONES
31 JOIN REGION R on D.REGIONES = R.REGIONES
32 ORDER BY sv.REGIONES;
```

2. Quels sont les 5 départements avec le plus grand taux d'énergie éolienne comme source de la puissance électrique en 2015 ?

```
40 SELECT DEPARTEMENTS
41 FROM ENVIRONNEMENTDEP
42 ORDER BY Eolien_2015 DESC
43 LIMIT 5;
```

3. Dans quelle région se trouve le département ayant le plus faible taux de valorisation matière et organique en 2013 ?

```
51 SELECT R.libelle
52 FROM REGION R join DEPARTEMENT D on R.REGIONS = D.REGIONS
53 join ENVIRONNEMENTDEP E on D.nccenr = E.DEPARTEMENTS
54 WHERE Taux_de_valorisation_matière_et_organique_2013 !=0
55 ORDER BY Taux_de_valorisation_matière_et_organique_2013
56 LIMIT 1;
```

4. En 2016, quelle est la part (en %) de l'agriculture biologique dans la surface agricole totale du département contenant le plus grand pourcentage de population éloignée de plus de 7 minutes des services de santé de proximité ?

```
68 SELECT E.Part_de_l_agriculture_biologique_dans_la_surface_agricole_totale_2016
69 FROM ENVIRONNEMENTDEP E join SOCIALDEP S on E.DEPARTEMENTS = S.DEPARTEMENTS
70 WHERE Part_de_la_population_éloignée_de_plus_de_7_mn_des_services_de_santé_de_proximité_2016 !=0
71 ORDER BY Part_de_la_population_éloignée_de_plus_de_7_mn_des_services_de_santé_de_proximité_2016 DESC
72 LIMIT 1;
```

5. Quel est le taux de pauvreté en 2014 des régions dont la part des jeunes non insérés est supérieure à 30% en 2014 ?

```
78 SELECT Taux_de_pauvreté_2014
79 FROM SOCIALREG
80 WHERE Part_des_jeunes_non_insérés_2014 > 30
81 AND Taux_de_pauvreté_2014 !=0;
```

6. En 2015, quelle était le poids de l'économie sociale dans les emplois salariés de la région dont la source de la puissance électrique en énergies renouvelables provenait à au moins 10% de l'énergie photovoltaïque et dont la part de l'agriculture biologique dans la surface agricole totale était d'au moins 5% ?

```
103 SELECT AVG(S.Poids_de_l_économie_sociale_dans_les_emplois_salariés_du_territoire_2015)
104 FROM SOCIALREG S join REGION R on S.REGIONS = R.nccenr
105 JOIN DEPARTEMENT D on R.REGIONS = D.REGIONS
106 JOIN ENVIRONNEMENTDEP E on D.nccenr = E.DEPARTEMENTS
107 GROUP BY R.nccenr
108 HAVING AVG(Photovoltaïque_2015) >= 10
109 AND AVG(Part_de_l_agriculture_biologique_dans_la_surface_agricole_totale_2016) >= 5;
```

## **BIBLIOGRAPHIE**

- [1] Python #32 - base de données, n.d.
- [2] How to Use SQL with Excel using Python, n.d.
- [3] Importing a CSV file into a sqlite3 database table using Python, Stack Overflow, (n.d.).
- [4] Read xls with Pandas, Pythonspot, (n.d.).
- [5] Cours et Tutoriels sur le Langage SQL, SQL, (n.d.).
- [6] Langage SQL: Sous requetes - YouTube, (n.d.).