

PROJET OCTOPLUS

DÉVELOPPEMENT ET EXPLOITATION D'UNE BASE DE DONNÉES SUR LES CRIMES ET DÉLITS

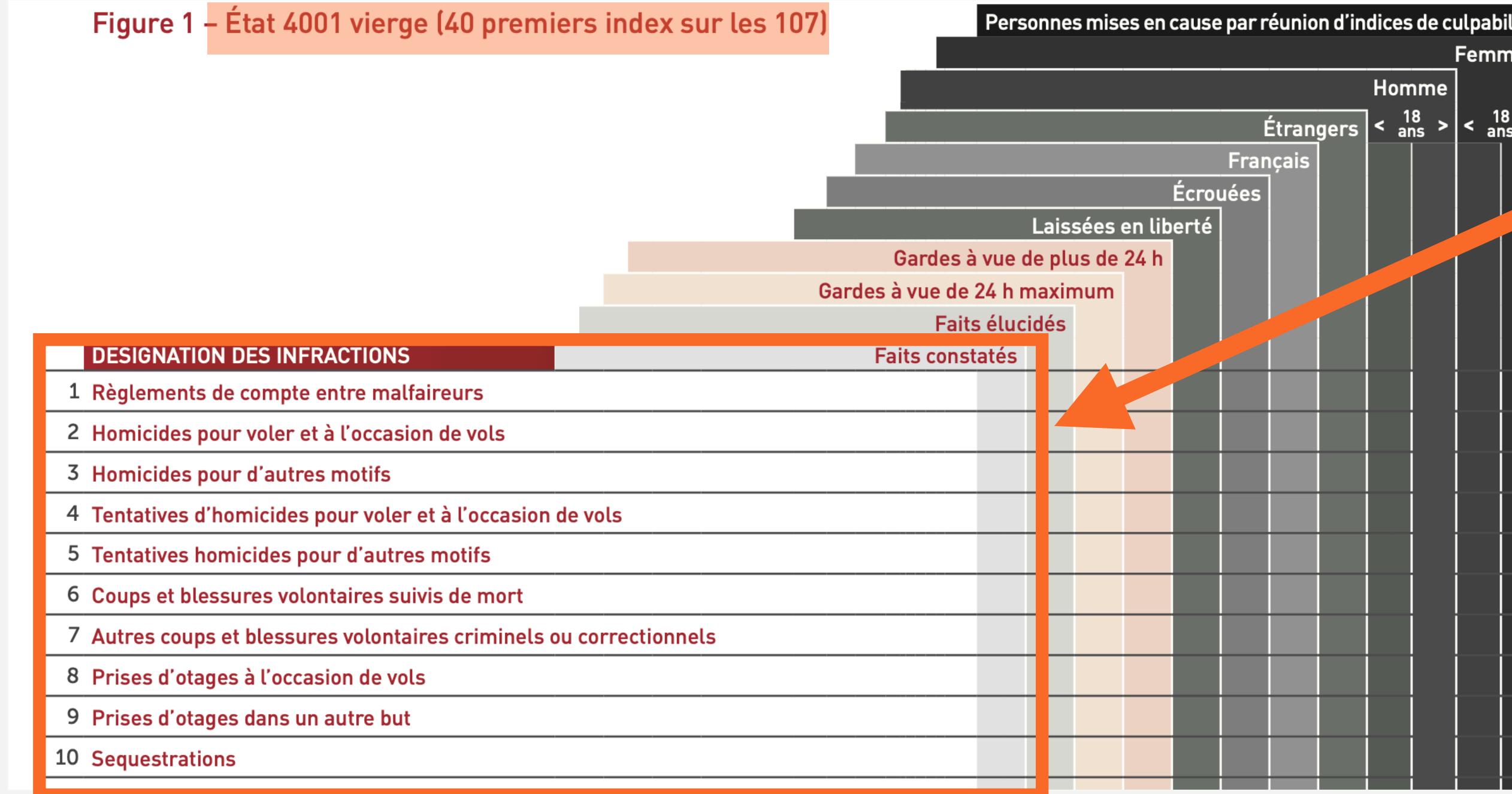
Gérald Bouget - Juillet 2020

Certifications : développer une base données - exploiter une base de données

Simplon - Formation Développeur Data - Fabrique Élancourt

UNE SOURCE DE DONNÉES HISTORIQUE : L'ÉTAT 4001

- ▶ **1972** : création d'un outil de suivi statistique qui comptabilise crimes et délits constatés par les forces de sécurité : l'État 4001.



Données en open data depuis 2015

Tous les crimes et délits hors contravention

Faits constatés pour la première fois

Métrique unique : les unités de compte

DES DONNÉES OPEN DATA ANNUELLES


data.gouv.fr
 Liberté • Égalité • Fraternité
 RÉPUBLIQUE FRANÇAISE

Plateforme ouverte des données publiques françaises

Données Réutilisations Organisations Tableau de bord Documentation

Recherche

Crimes et délits enregistrés par les services de gendarmerie et de police depuis 2012

Ce jeu de données provient d'un service public certifié



Ces données constituent une actualisation des volumes Criminalité et délinquance constatées en France édités par la Direction centrale de la police judiciaire, publiés jusqu'en 2013 à la Documentation française, et disponibles sur son site [internet](#). Elles contiennent des informations, de caractère administratif, sur l'activité judiciaire des services de gendarmerie et de police, y compris celles des DOM-COM, depuis 2012.

Ressources

crimes-et-delits-enregistres-par-les-services-de-gendarmerie-et-de-police-depuis-2012.xlsx
 Ces données constituent une actualisation des volumes Criminalité et délinquance constatées en France édités par la Direction centrale de la police judiciaire, publiés jusqu'en...

 3.4Mo  Disponible

[PRÉVISUALISER](#) [TÉLÉCHARGER](#)

Documentation des chiffres annuels relatifs aux crimes et délits enregistrés par les services de gendarmerie...
 1.1Mo  Disponible

[TÉLÉCHARGER](#)

[CONTACTER LE PRODUCTEUR](#)  

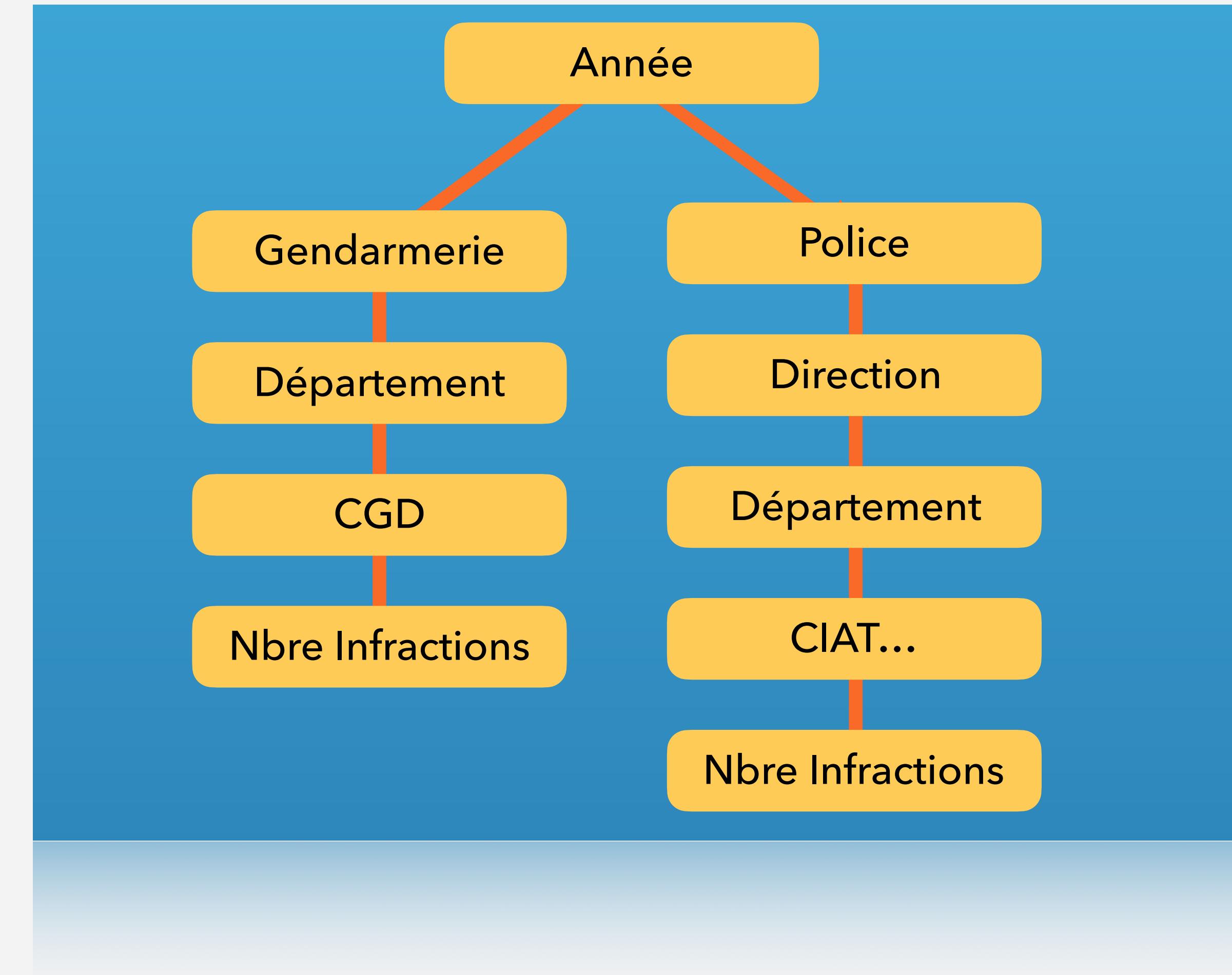
Producteur


MINISTÈRE DE L'INTÉRIEUR

Ministère de l'Intérieur

Placé au cœur de l'État, le ministère de l'Intérieur assure la permanence et la continuité de l'État. Cette fonction régaliennes se concrétise par le rôle majeur et les services...

[VOIR LE PROFIL](#)
[CONTACTER](#)
[SUIVRE](#)



ENJEUX : RÉALISER UNE BASE DE DONNÉES FACILEMENT ACCESSIBLE

Limites du dataset

- ▶ Dataset peu lisible : large fichier excel, multiples onglets, multi-indexés
- ▶ Données limitées au nombre d'infractions et donc difficilement interprétables.
- ▶ Un nombre de type d'infractions important (107) et pas toujours compréhensible

Solution proposée

- ▶ Développer une base de données qui permette une exploitation optimale de cette source.
- ▶ Compléter ce dataset de données externes (indicateurs statistiques, données démographiques, économiques, sociales) pour une analyse croisée et comparative.

CONCEPTION ET DÉVELOPPEMENT D'UNE BASE DE DONNÉES SUR LA DÉLINQUANCE : ANALYSE DE LA DEMANDE

UNE BASE DE DONNÉES SUR LA DÉLINQUANCE : POURQUOI ? POUR QUI ?

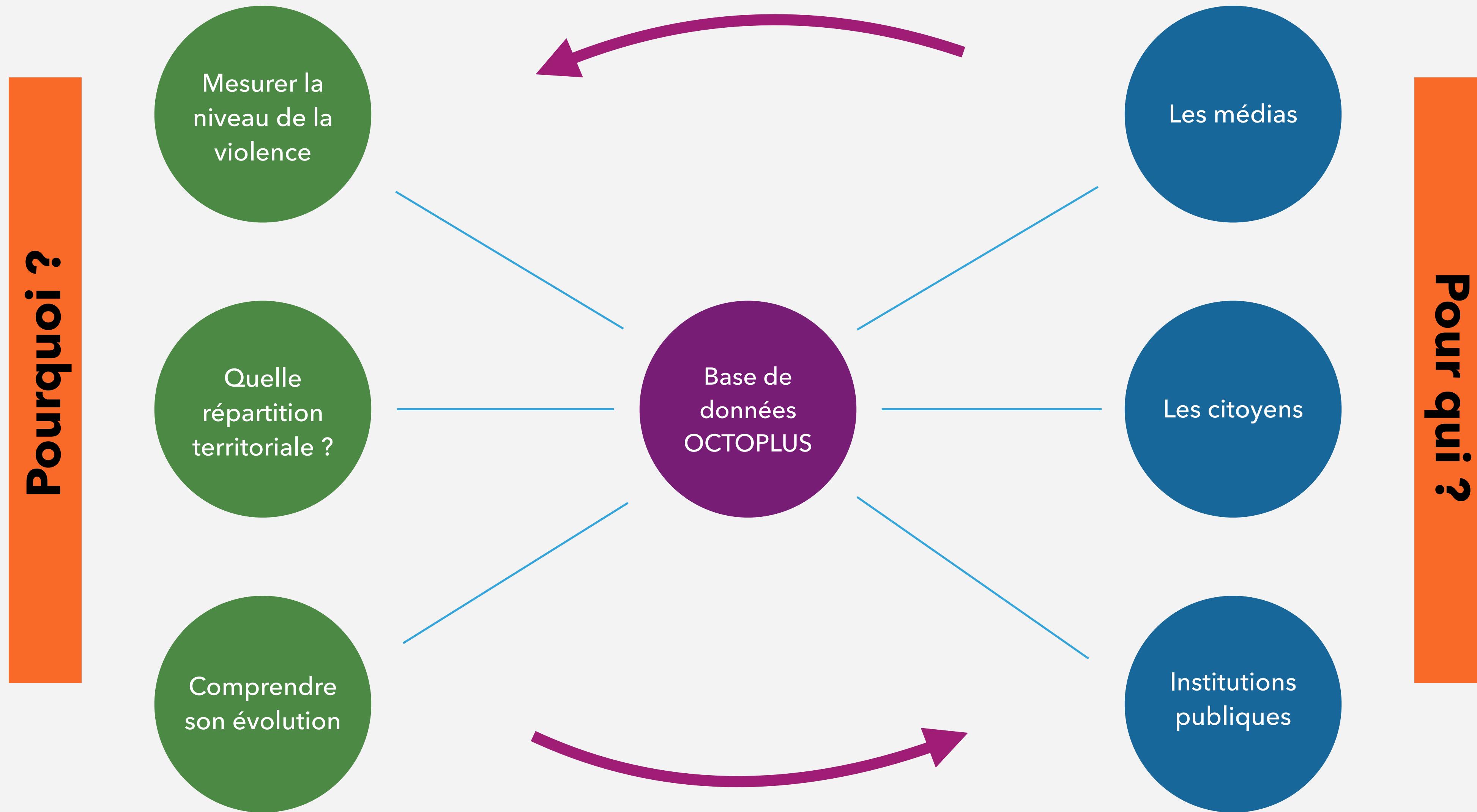
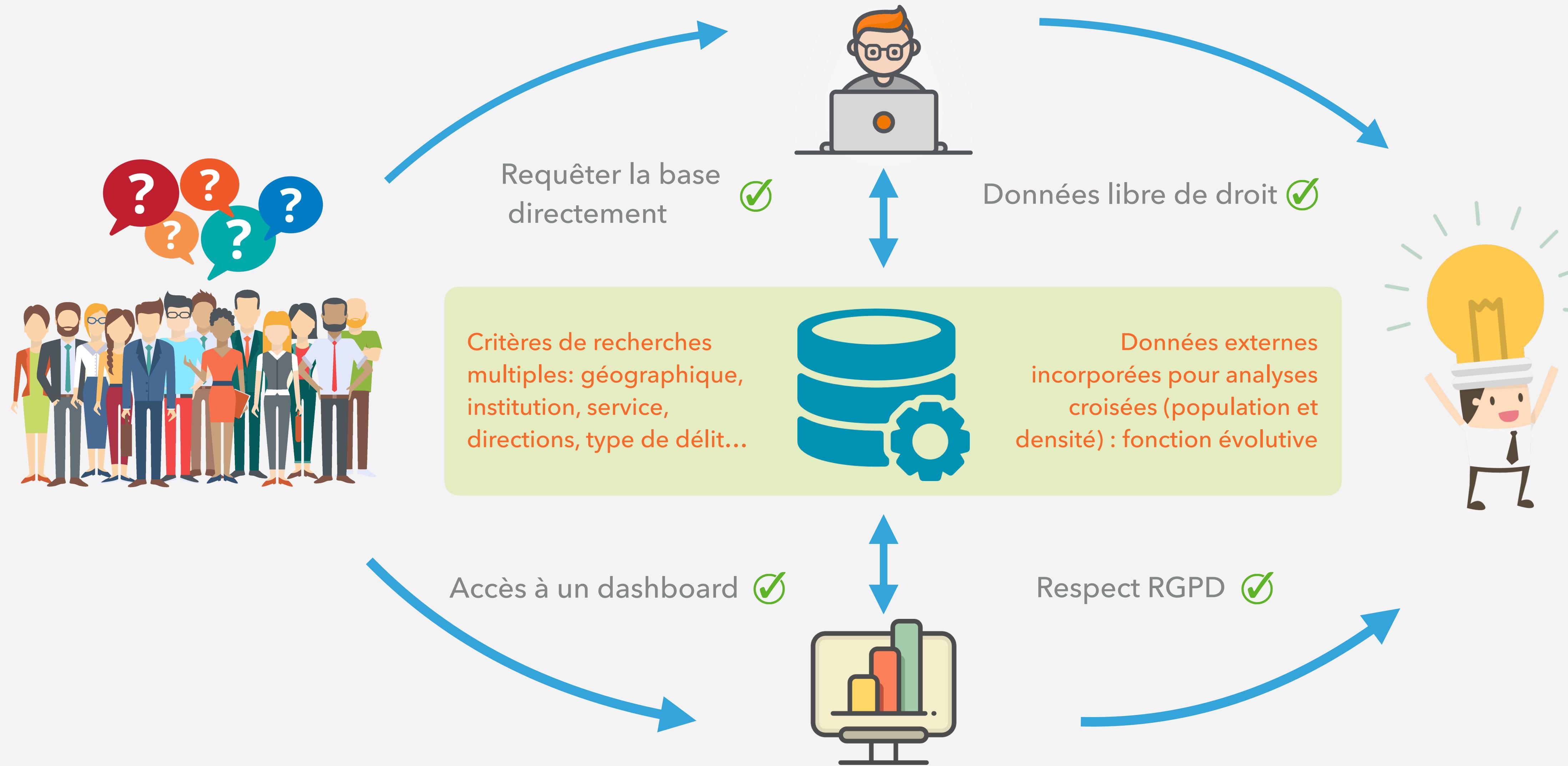


SCHÉMA FONCTIONNEL DU PROJET



MACRO PLANNING

MACRO - PLANNING INITIAL VS RÉALISÉ

| 7/05 - 10/05 | 10/05 - 20/05 | 20/05 - 31/05 | 01/06 - 05/06 | 10/06 - 30/06 | 05/07 - 12/07 |
|--------------------------------------|--|---|---|--|--|
| Analyse des besoins et des objectifs | Développement de la base de données | Exploitation de la base de données | Préparation 'prez' client + marge de sécurité | Optimisation + dashboard | Déploiement BD + dashboard en ligne |
| Analyse des besoins et des objectifs | <ol style="list-style-type: none"> 1. Modélisation 2. Préparation données 3. Création tables et import 4. Contraintes, optimisation 5. Sauvegarde | <ol style="list-style-type: none"> 1. Requêtes et vérification cohérence des données 2. Analyse de données 3. Visualisation via plotly dans NB | 1ere 'prez' client + finitions | <ol style="list-style-type: none"> 1. App automatisation backup et MAJ optimisation BD 2. Dashboard Dash | <ol style="list-style-type: none"> 1. Déploiement BD sur RDS 2. Dashboard sur Heroku |
| | | | | | |

GESTION PLANNING

OCTOPLUS - GESTION DE PROJET

Alibaba_team Free | Privé | Inviter | Trello | Calendrier | Butler (5 Tips) | Afficher le menu

BACKLOG

- C8 - Analyse des besoins du client
- C8 - Analyse du sujet et du jeu de données. Choix d'un angle
- C8 - Rédaction d'un cahier des charges
- C2 - Identifier les sources de données
- C1 - sélectionner une base de donnée et justifier son choix
- C1 - MCD/MLD/MPD
- C2 - Sauvegarde copie dataset

SPRINT 22/06 - 26-06

- C2 - MAJ rep de données suite ajouts attributs
- C3 - MAJ automatisé de la base de données >> ajouter exceptions/gestion d'erreurs pour partie connexion base et insertion données
- C7 - Dashboard - onglet paris

ISSUES

- C5 - Dashboard >> lenteur sur certaines requêtes : faire vues matérialisées pour amélioration temps de requête

REVIEW AND TESTING

+ Ajouter une carte

FINALISÉ

- C4 - Création utilisateur en mode consultation only
- C1 - Automatiser le processus de sauvegarde de la base de données
- C4 - automatiser la recherche de doublons et d'erreurs dans les données stockées
- C4 - mesurer et rendre compte des performances des requêtes du projet
- C5 - Définir des requêtes de récupération des données
- C8 - Rédaction d'un cahier des charges
- C5 - Interroger et traiter, simultanément et au niveau approprié, des données afin de les stocker en sécurité, brutes ou traitées, provisoirement ou durablement, en fonction du

CERTIFICATION DVE DATA

- C1. Concevoir et structurer physiquement une base de données relationnelle ou non, à partir des besoins, contraintes et données du commanditaire
- C2. Acquérir des données, les combiner et les structurer en données propres en vue de leur intégration dans la structure de la base de données
- C3. Intégrer des données propres et préparées dans la base de données finale, en utilisant des langages informatiques, logiciels ou outils
- C4. Optimiser une base de données afin d'en maintenir la fiabilité et la qualité des données. Nettoyer et améliorer les performances
- C5 Interroger et traiter, simultanément et au niveau approprié, des données afin de les stocker en sécurité, brutes ou traitées, provisoirement ou durablement, en fonction du

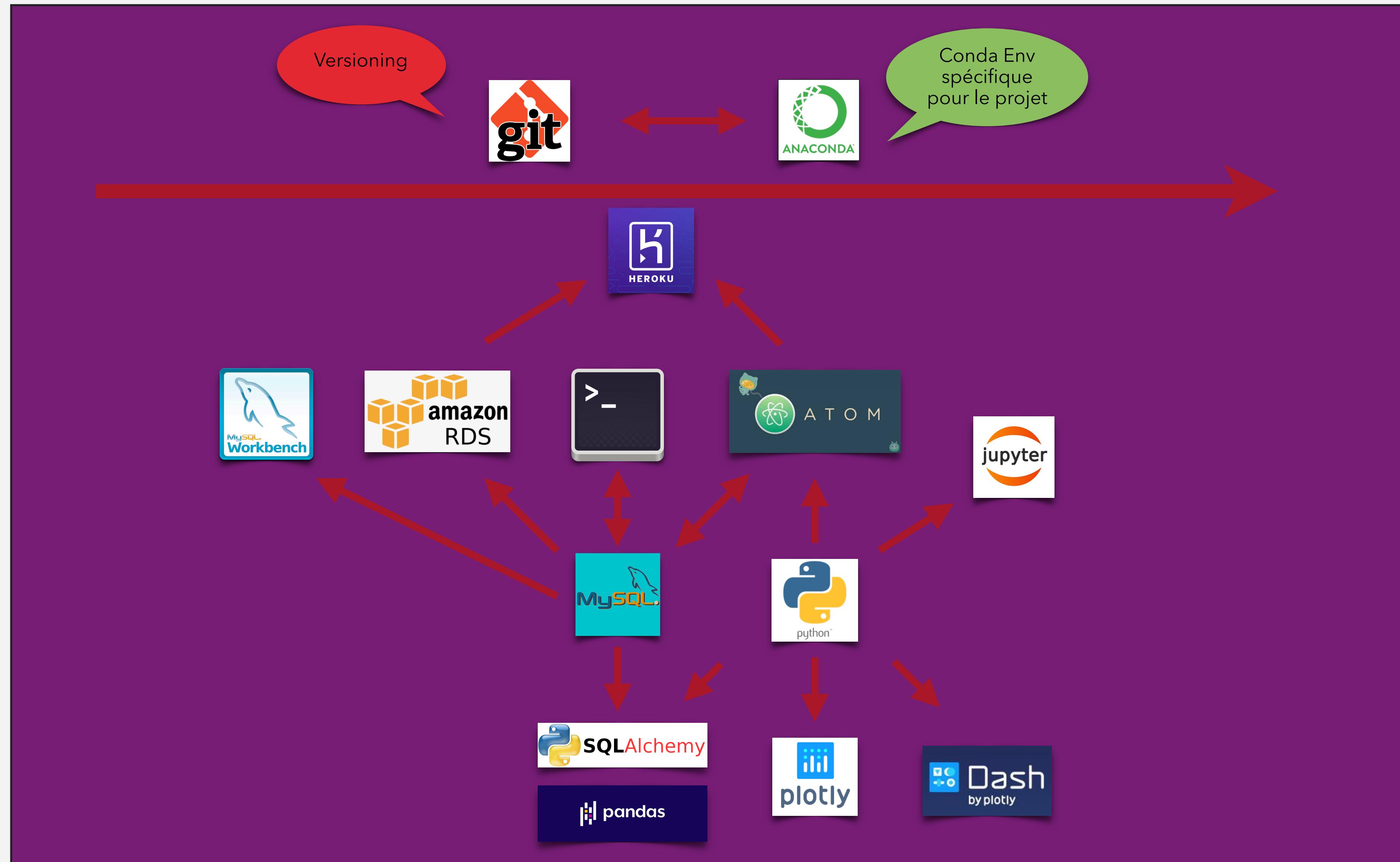
+ Ajouter une autre carte

CONCEPTION ET DÉVELOPPEMENT D'UNE BASE DE DONNÉES SUR LA DÉLINQUANCE :

CONCEPTION

CONCEPTION TECHNIQUE

CHOIX DES TECHNOLOGIES 3/3



CHOIX DES TECHNOLOGIES 1/3

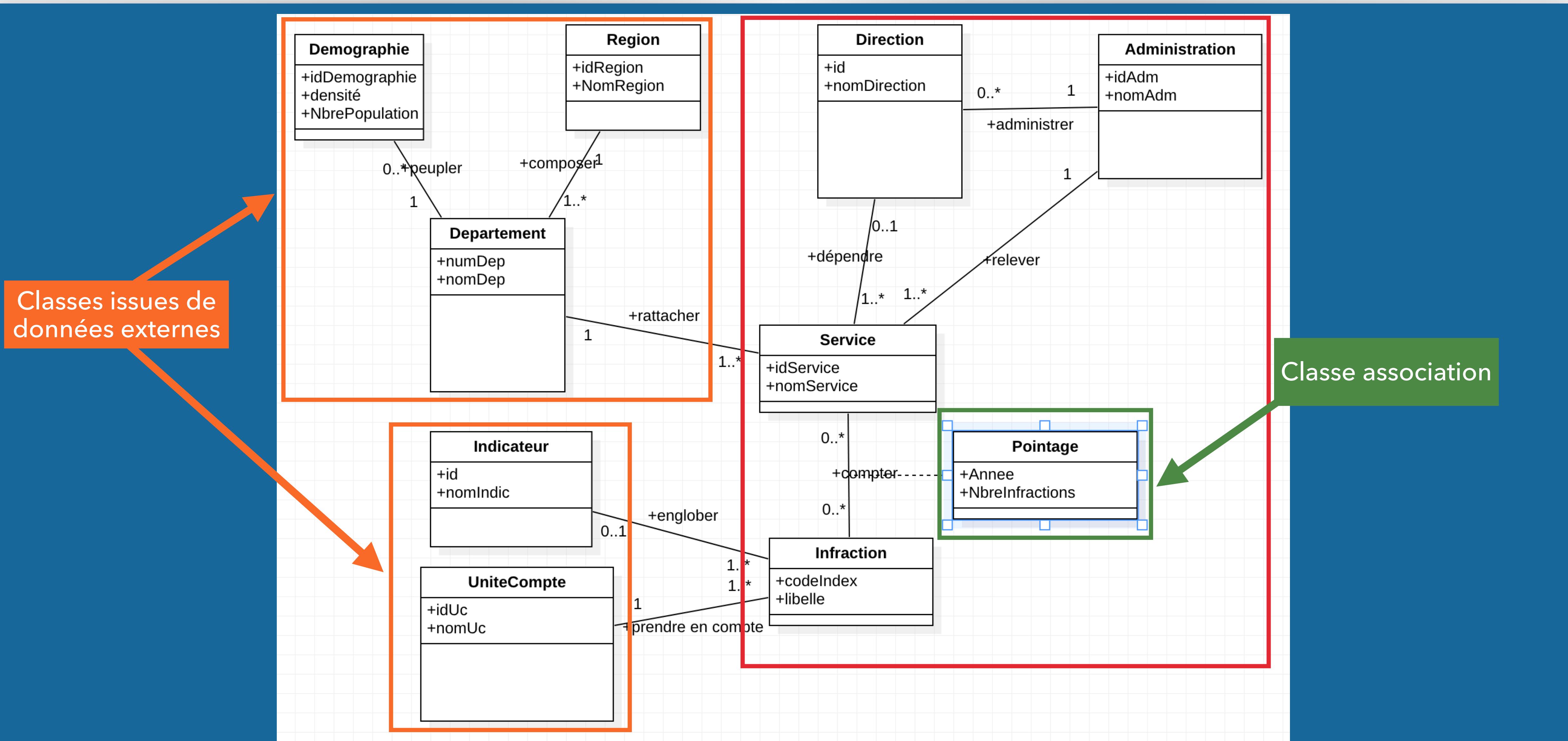
| Outils | Utilisation | Raison |
|-----------------------------|---|---|
| MySQL | - Conception de la base de données | - Open source et gratuit - Documentation importante - Adapté pour une utilisation web |
| MySQL Workbench | - Modélisation MPD - Utilisation Procédures stockées | - Interface intuitive - Gratuit - Inclu dans le package MySQL |
| Python | - Tout au long du processus de préparation / structuration des données / Exploitation / visualisation | - Bonne maîtrise du langage - Librairies adaptées à mon projet |
| Openpyxl (librairie Python) | - préparation et structuration des tables | - Format du dataset en excel - Large possibilités de manipulation de données en excel |
| Pandas (librairie python) | - préparation / nettoyage des données - exploitation des données - Requêtes SQL | - Outil de base de manipulation des données - Très performant - Facile d'utilisation |

CHOIX DES TECHNOLOGIES 2/3

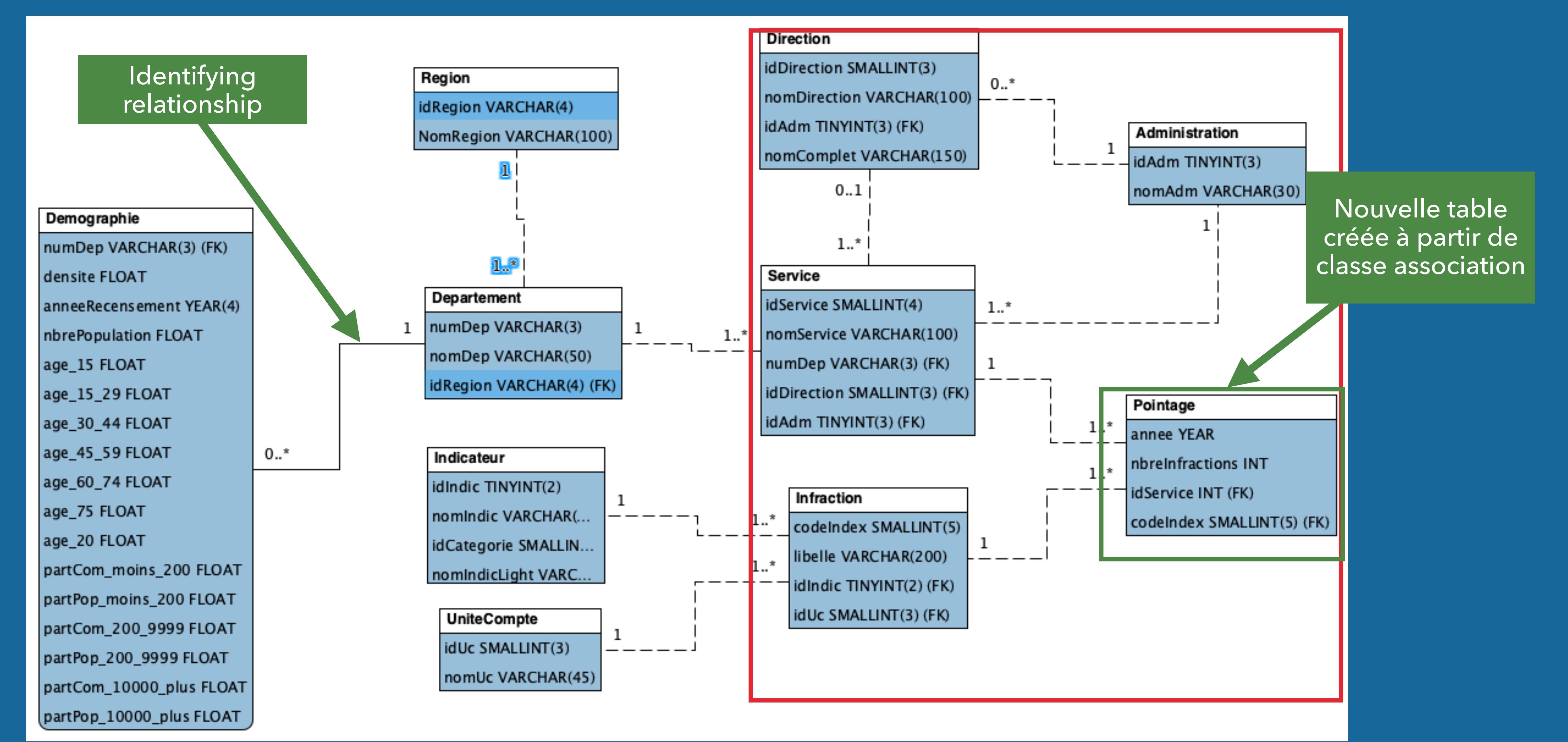
| | | |
|-----------------------|---|---|
| SQL Alchemy | - Connexion à la base données MySQL dans un environnement Python | - ORM performant |
| Plotly / ipyWidget | - Visualisation des données | - Larges options - Interactivité - Implémenté dans Dash -ipydiget : pour interaction dans Jupiter notebook / Prévisualisation pour Dash |
| Dash | - Dashboard | - Utilise python - Gratuit - Performant pour l'interactivité |
| Open Office | - Ouverture et lecture du dataset | - Gratuit - Mêmes fonctionnalités que Excel en lecture |
| Jupyter Notebook | - Préparation des données -Exploitation des données - Visualisation | - Distribution performante et souple pour utiliser avec Python |
| Atom | - Scripts SQL Applications /Dashboard | - Gratuit et habitué |
| UML | - MCD | - Interface ergonomique |
| Git | -utiliser pour versionning | - ères pratique pour suivi version et performant |
| Conda Env. | -Pour applications | -Permet pérennité du projet et cross platform |

MPD ET MCD - DICTIONNAIRE DE DONNÉES 1/3

| Attribut | Désignation | Type |
|-------------------------|---|----------------------|
| numDep | numéro de département | CHAINE DE CARACTÈRES |
| densité | densité de la population au km2 | FLOAT |
| population | nombre de population | FLOAT |
| anneeRecensement | année de recensement ou de collecte des données démographiques/économiques | YEAR |
| IdRegion | numéro de la région (code INSEE region) | CHAINE DE CARACTÈRES |
| nomRegion | nom de la région | CHAINE DE CARACTÈRES |
| nomDep | nom du département | CHAINE DE CARACTÈRES |
| idDirection | numéro auto-incrémenté pour identifier chaque Direction de police | INT |
| nomDirection | sigle du nom de la direction de police | CHAINE DE CARACTÈRES |
| idAdm | numéro auto-incrémenté pour identifier chaque type d'administration (police ou gendarmerie) | INT |
| nomComplet | nom complet du nom des directions de police | CHAINE DE CARACTÈRES |
| nomAdm | nom de l'administration | CHAINE DE CARACTÈRES |



MODÈLES PHYSIQUE DE DONNÉES (MPD) - 3/3



CONCEPTION ET DÉVELOPPEMENT D'UNE BASE DE DONNÉES SUR LA DÉLINQUANCE : DÉVELOPPEMENT

DATA PROCESS



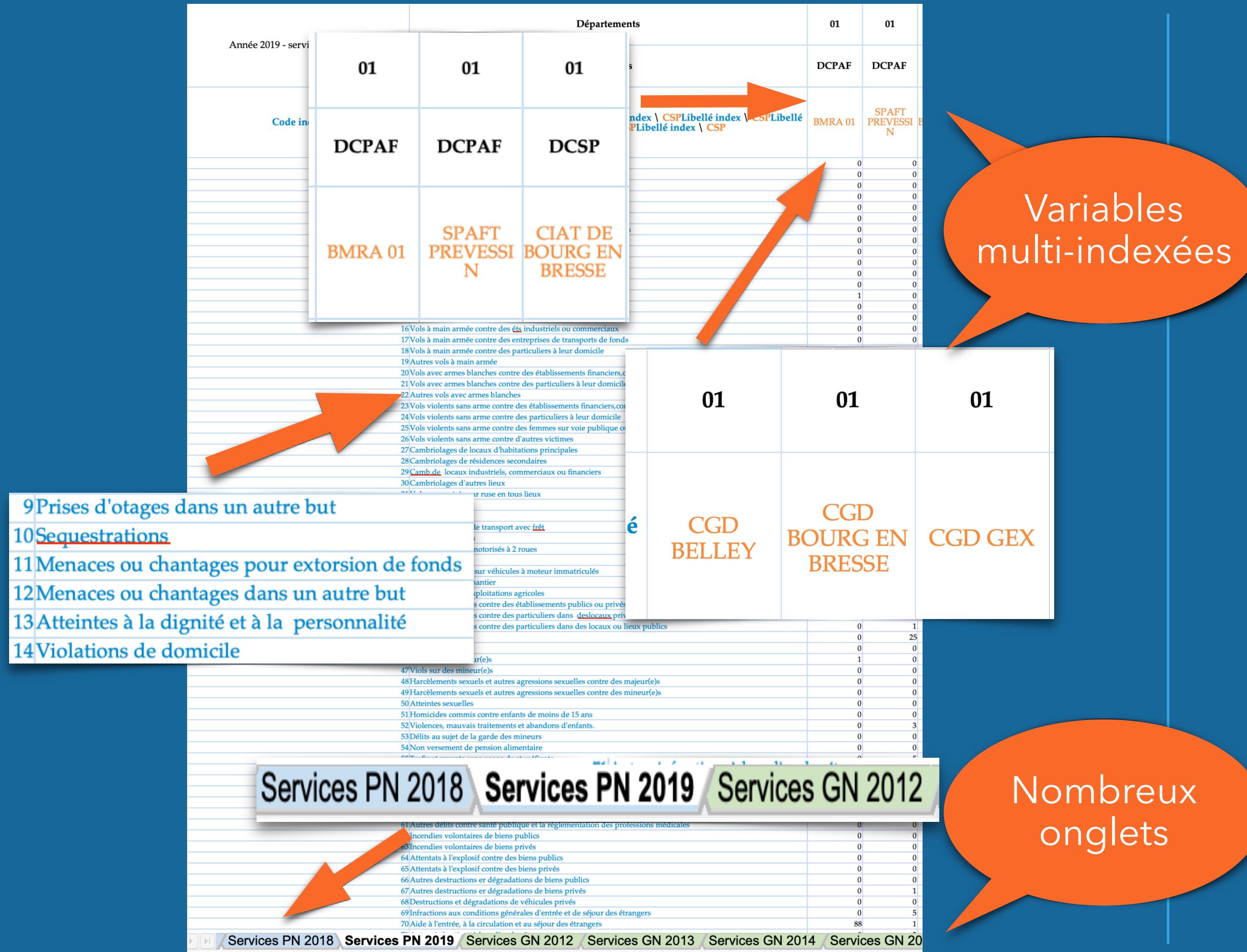
Dataset au format Excel pour dataset principal et CSV pour datasets supplémentaires (département, Région, Démographie..)

1. Import fichier via Openpyxl/ Pandas.excelFile
2. Nettoyage des données et structuration des tables avec Openpyxl et Pandas

1. Export des tables en csv dans dossier 'dump' (un fichier par table)
2. Export sans PK pour clées Auto-Inc..
3. valeurs des FK incluses

1. Création de la base et des tables
2. Import des tables avec l'instruction LOAD DATA INFILE
3. Ajout des contraintes SQL
4. Optimisation, utilisateurs et sécurisation
5. Procédure de sauvegarde

Difficulté majeure : restructurer les données



| | 1 | df_stat | | | |
|----|-------------------------|------------|----------------|--------------|----------------|
| | | nomOnglets | nombreColonnes | nombreLignes | nombreCellules |
| 0 | Résumé de l'exportation | | 4 | 42 | 168 |
| 1 | Présentation | | 1 | 1 | 1 |
| 2 | Services PN 2012 | | 609 | 110 | 66990 |
| 3 | Services PN 2013 | | 604 | 110 | 66440 |
| 4 | Services PN 2014 | | 599 | 110 | 65890 |
| 5 | Services PN 2015 | | 733 | 110 | 80630 |
| 6 | Services PN 2016 | | 683 | 110 | 75130 |
| 7 | Services PN 2017 | | 680 | 110 | 74800 |
| 8 | Services PN 2018 | | 680 | 110 | 74800 |
| 9 | Services PN 2019 | | 681 | 110 | 74910 |
| 10 | Services GN 2012 | | 383 | 109 | 41747 |
| 11 | Services GN 2013 | | 385 | 109 | 41965 |
| 12 | Services GN 2014 | | 382 | 109 | 41638 |
| 13 | Services GN 2015 | | 382 | 109 | 41638 |
| 14 | Services GN 2016 | | 379 | 109 | 41311 |
| 15 | Services GN 2017 | | 379 | 109 | 41311 |
| 16 | Services GN 2018 | | 379 | 109 | 41311 |
| 17 | Services GN 2019 | | 379 | 113 | 42827 |

JUPYTER NOTEBOOK

Solution choisie :

- structurer chaque table SQL séparément en utilisant openpyxl et pandas
- contrôle de la cohérence et intégrité des données à chaque étape.

EXEMPLES DE PROCESS AVEC LES TABLES 'SERVICE' ET 'POINTAGE'

| 1 table_service_police.head() | | | | |
|---------------------------------|-------------------------|-------------|-------|--|
| numDep | nomService | idDirection | idAdm | |
| 0 01 | DIDPAF PREVESSIN 01 | 1 | 1 | |
| 1 01 | CIAT DE BOURG EN BRESSE | 2 | 1 | |
| 2 01 | CIAT D'OYONNAX | 2 | 1 | |
| 3 02 | CIAT DE CHATEAU THIERRY | 2 | 1 | |
| 4 02 | CIAT DE LAON | 2 | 1 | |

| 1 table_service_gendarmerie.head() | | | | |
|--------------------------------------|---------------------|-------------|-------|--|
| numDep | nomService | idDirection | idAdm | |
| 0 01 | CGD BELLEY | Nan | 2 | |
| 1 01 | CGD BOURG EN BRESSE | Nan | 2 | |
| 2 01 | CGD GEX | Nan | 2 | |
| 3 01 | CGD TREVOUX | Nan | 2 | |
| 4 02 | CGD CHATEAU THIERRY | Nan | 2 | |

| 1 table_pointage_gendarmerie.head() | | | | |
|---------------------------------------|---------------------|-----------------|-------|--|
| codeIndex | nomService | nbreInfractions | annee | |
| 0 1 | CGD BELLEY | 0 | 2012 | |
| 1 1 | CGD BOURG EN BRESSE | 0 | 2012 | |
| 2 1 | CGD GEX | 0 | 2012 | |
| 3 1 | CGD TREVOUX | 0 | 2012 | |
| 4 1 | CGD CHATEAU THIERRY | 0 | 2012 | |

| 1 table_pointage_police.head() | | | | |
|----------------------------------|-------------------------|-----------------|-------|--|
| codeIndex | nomService | nbreInfractions | annee | |
| 0 1 | DIDPAF PREVESSIN 01 | 0 | 2012 | |
| 1 1 | CIAT DE BOURG EN BRESSE | 0 | 2012 | |
| 2 1 | CIAT D'OYONNAX | 0 | 2012 | |
| 3 1 | CIAT DE CHATEAU THIERRY | 0 | 2012 | |
| 4 1 | CIAT DE LAON | 0 | 2012 | |

| numDep | nomService | idDirection | idAdm | idService |
|--------|-------------------------|-------------|-------|-----------|
| 0 01 | DIDPAF PREVESSIN 01 | 1 | 1 | 1 |
| 1 01 | CIAT DE BOURG EN BRESSE | 2 | 1 | 2 |
| 2 01 | CIAT D'OYONNAX | 2 | 1 | 3 |
| 3 02 | CIAT DE CHATEAU THIERRY | 2 | 1 | 4 |
| 4 02 | CIAT DE LAON | 2 | 1 | 5 |

| 1 table_pointage_PG.head() | | | | |
|------------------------------|-------------------------|-----------------|-------|--|
| codeIndex | nomService | nbreInfractions | annee | |
| 0 1 | DIDPAF PREVESSIN 01 | 0 | 2012 | |
| 1 1 | CIAT DE BOURG EN BRESSE | 0 | 2012 | |
| 2 1 | CIAT D'OYONNAX | 0 | 2012 | |
| 3 1 | CIAT DE CHATEAU THIERRY | 0 | 2012 | |
| 4 1 | CIAT DE LAON | 0 | 2012 | |

| Pointage | | | | |
|----------|------|-----------------|-----|----------------------------|
| annee | YEAR | nbreInfractions | INT | idService |
| | | | | INT (FK) |
| | | | | codeIndex SMALLINT(5) (FK) |

| 1 table_pointage_PG_def.head() | | | | | | |
|----------------------------------|-------------------------|-----------------|-------|-----------|-------------|--|
| codeIndex | nomService | nbreInfractions | annee | idService | idDirection | |
| 0 1 | DIDPAF PREVESSIN 01 | 0 | 2012 | 1 | 1 | |
| 1 1 | CIAT DE BOURG EN BRESSE | 0 | 2012 | 2 | 2 | |
| 2 1 | CIAT D'OYONNAX | 0 | 2012 | 3 | 2 | |
| 3 1 | CIAT DE CHATEAU THIERRY | 0 | 2012 | 4 | 2 | |
| 4 1 | CIAT DE LAON | 0 | 2012 | 5 | 2 | |

EXEMPLE DE VÉRIFICATION D'INTÉGRITÉ DES DONNÉES

```
1 #chargement workbook excel
2 wb=openpyxl.load_workbook('crimes-et-delits-enregistres-par-les-services-de-gendarmerie-et-de-police-depuis-2012_C
1 # CHARGEMENT FICHIER IDENTIQUE QUE PRÉCEDENT (UTILISATION CODE DIFFERENTE)
2 wbpd=pd.ExcelFile('crimes-et-delits-enregistres-par-les-services-de-gendarmerie-et-de-police-depuis-2012_ORIGINAL.
```

méthodes de chargement du dataset

```
183]: 1 def depDiff():
2     #comparaison département entre années
3     for i in range(len(liste)-1):
4         if set(police[liste[i]]['departement']) == set(police[liste[i+1]]['departement']):
5             print(str(liste[i]) + ' ' + 'a les mêmes départements que' + ' ' + str(liste[i+1]))
6         else:
7             print(str(liste[i]) + ' n\'a pas les mêmes départements que ' + str(liste[i+1]))
8
```

```
184]: 1 depDiff()
```

```
Services PN 2012 a les mêmes départements que Services PN 2013
Services PN 2013 a les mêmes départements que Services PN 2014
Services PN 2014 a les mêmes départements que Services PN 2015
Services PN 2015 a les mêmes départements que Services PN 2016
Services PN 2016 a les mêmes départements que Services PN 2017
Services PN 2017 a les mêmes départements que Services PN 2018
Services PN 2018 a les mêmes départements que Services PN 2019
```

Vérification que j'ai bien les mêmes départements dans chaque onglet

EXEMPLE DE VÉRIFICATION D'INTÉGRITÉ DES DONNÉES

```
1  liste_dep_dir_Serv=depDirServList()
2  liste_dep_dir_Serv[0][:6]

01', 'DCPAF', 'DIDPAF PREVESSIN 01', '01', 'DCSP', 'CIAT DE BOURG EN BRESSE']

1  def CombinaisonUniqueService():
2      # DF vierge
3      df_triplets = pd.DataFrame({
4          'numDep':[],
5          'nomDirection':[],
6          'nomService':[]
7      })
8
9      # creation liste des dataframe des combinaisons uniques dep-direction-service pour chaque années
10     # ATTENTION : doublons présents à éliminer après pour avoir uniquement des combinaisons uniques
11     frames_triplets = []
12     for l in liste_dep_dir_Serv:
13         subList = [l[n:n+3] for n in range(0, len(l), 3)] #sous-liste des combinaisons dep-dir-service
14         df_triple_temp = pd.DataFrame(subList).rename(columns={0:'numDep', 1:'nomDirection', 2:'nomService'})
15         frames_triplets.append(df_triple_temp)
16
17     # assemblage des dataframe de la liste
18     df_triplets = pd.concat(frames_triplets) #assemblage de tous les df en un seul
19
20     # suppression des duplicate rows
21     df_triplets_def = df_triplets.drop_duplicates(keep='first').dropna(axis=0)
22     #keep first pour garder uniquement valeur du 1er duplicate
23     #print(f'longeur du dataframe avec suppression doublon : {len(df_triplets)}\nlongeur du dataframe avec suppression doublon : {len(df_triplets_def)}')
24
25     return df_triplets_def
```

Suppression doublons

EXEMPLE DE VÉRIFICATION D'INTÉGRITÉ DES DONNÉES

```
1 # check libelle et codeindex identiques pour tous les onglets
2 def checkLibelle(fichier_excel):
3     liste_probleme=[ ]
4     df_check=pd.DataFrame({'Code index':[], 'libelle':[]})
5     # on concat df_check avec le 1er onglet année 2012
6     # pour éviter multiindexage on ne retient que les variables service (header 1 pour gendarmerie)
7     df=pd.read_excel(fichier_excel,sheet_name=liste[0], header=1).rename(columns={'Libellé index \\ CGD':'libel'])
8     df=df.iloc[:,[0,1]] # df avec uniquement colonne code index et libelle
9     df_check=pd.concat([df_check, df])
10
11
12     # on concat df check avec les années suivantes
13     # on sépare les 2 process de concat pour que la comparaison se fasse à partir de la 2eme année
14     for sheetname in liste[1::]:
15         # on garde uniquement variable service pour éviter multiindexage (header=1 pour gendarmerie)
16         df=pd.read_excel(fichier_excel,sheet_name=sheetname, header=1).rename(columns={'Libellé index \\ CGD':'libel'})
17         df=df.iloc[:,[0,1]] # df avec uniquement colonne code index et libelle
18         df_check=pd.concat([df_check, df])
19         if len(df_check[['Code index', 'libelle']].drop_duplicates(keep=False))>0:
20             liste_probleme.append([str(sheetname), df_check[['Code index', 'libelle']].drop_duplicates(keep=False)])
21
22 return liste_probleme
```

EXEMPLE DE VÉRIFICATION D'INTÉGRITÉ DES DONNÉES

```
76]: 1 checkLibelle(wbpd)
:
[[ 'Services GN 2019',
  Code index                                libelle
  95      98.0                                Index non utilisé
  96      101.0                               Index non utilisé
  97      102.0  Banqueroutes, abus de biens sociaux et autres ...
  98      103.0                               Index non utilisé
  99      104.0                               Index non utilisé
  100     105.0  Prix illicites, publicité fausse et infractio...
  101     106.0      Achats et ventes sans factures
  102     107.0  Infractions à l'exercice d'une profession régl...
  103      NaN  Infractions au droit de l'urbanisme et de la c...
  104      NaN      Fraudes fiscales
  105      NaN  Autres délits économiques et financiers
  106      NaN      Autres délits]]
```

```
1 >>> erreur dans le dataset pour l'année 2019:
2 décalage dans les lignes à partir de de l'index 98
3 choix de corriger le dataset original et de travailler à partir de ce nouveau dataset
```

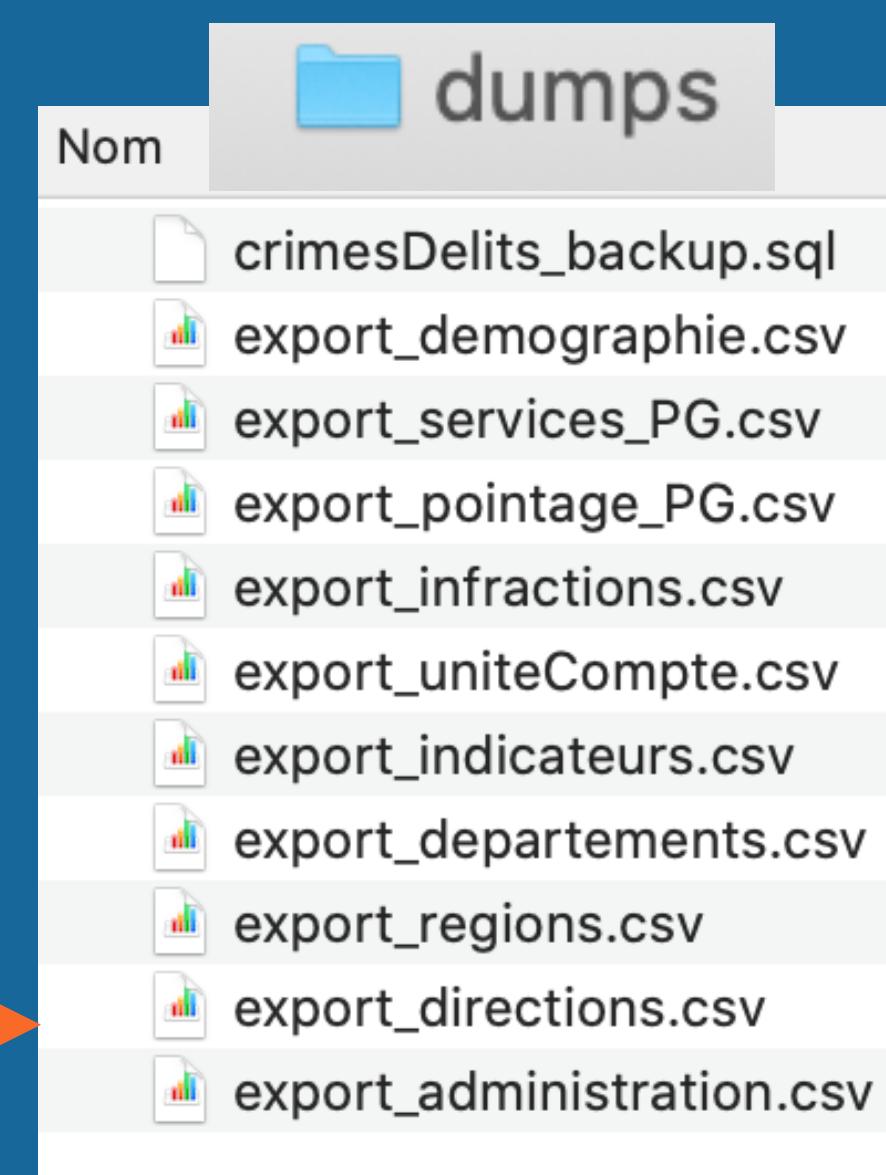
WARNING : DATASET ORGINAL MODIFIE : LE BON DATASET EST LE SUIVANT

CRÉATION DES TABLES DANS MYSQL ET IMPORTATION DES DONNÉES

```
CREATE TABLE Service(
    idService SMALLINT(4) NOT NULL AUTO_INCREMENT,
    nomService VARCHAR(100) NULL,
    numDep VARCHAR(3) NULL,
    idDirection SMALLINT(3) NULL,
    idAdm SMALLINT(3) NULL,
    PRIMARY KEY(idService)
)
ENGINE = InnoDB;

CREATE TABLE Pointage(
    annee YEAR NOT NULL,
    nbreInfractions INT NOT NULL,
    idService SMALLINT(4) NOT NULL,
    codeIndex SMALLINT(5) NOT NULL
)
ENGINE = InnoDB;
```

fichiers exportés à partir du
Notebook après nettoyage et
structuration des tables



```
LOAD DATA INFILE '/Users/geraldbouget/dumps/export_services_PG.csv'
INTO TABLE Service
FIELDS TERMINATED BY ","
ENCLOSED BY ""
LINES TERMINATED BY "\n"
IGNORE 1 LINES
(numDep, nomService, @value, idAdm)
SET
idDirection = if(@value=0, NULL, @value);

LOAD DATA INFILE '/Users/geraldbouget/dumps/export_pointage_PG.csv'
INTO TABLE Pointage
FIELDS TERMINATED BY ","
ENCLOSED BY ""
LINES TERMINATED BY "\n"
IGNORE 1 LINES
(annee, nbreInfractions, idService, codeIndex);
```

AJOUT DES CONTRAINTES SQL

```
-- CREATION FOREIGN KEYS --  
  
ALTER TABLE Service  
ADD CONSTRAINT FK_numDep  
FOREIGN KEY (numDep) REFERENCES Departement(numDep)  
ON UPDATE CASCADE,  
ADD CONSTRAINT FK_idDirection  
FOREIGN KEY (idDirection) REFERENCES Direction(idDirection)  
ON UPDATE CASCADE,  
ADD CONSTRAINT FK_idAdm  
FOREIGN KEY (idAdm) REFERENCES Administration(idAdm)  
ON UPDATE CASCADE;
```

```
mysql> describe service;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| idService | smallint(4) | NO | PRI | NULL | auto_increment |  
| nomService | varchar(100) | YES | MUL | NULL | |  
| numDep | varchar(3) | YES | MUL | NULL | |  
| idDirection | smallint(3) | YES | MUL | NULL | |  
| idAdm | smallint(3) | YES | MUL | NULL | |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0,00 sec)
```

```
ALTER TABLE Pointage  
ADD CONSTRAINT FK_idService  
FOREIGN KEY (idService) REFERENCES Service(idService)  
ON UPDATE CASCADE,  
ADD CONSTRAINT FK_codeIndex  
FOREIGN KEY (codeIndex) REFERENCES Infraction(codeIndex)  
ON UPDATE CASCADE;
```

```
mysql> describe pointage;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| annee | year(4) | NO | | NULL | |  
| nbreInfractions | int(11) | YES | | NULL | |  
| idService | smallint(4) | NO | MUL | NULL | |  
| codeIndex | smallint(5) | NO | MUL | NULL | |  
+-----+-----+-----+-----+-----+-----+  
4 rows in set (0,00 sec)
```

AJOUT DES INDEXES

```
mysql> show tables;
+-----+
| Tables_in_crimesdelits |
+-----+
| Administration
| Demographie
| Departement
| Direction
| Indicateur
| infraction
| Pointage
| Region
| repart_1000hbts
| service
| uniteCompte
+-----+
11 rows in set (0,00 sec)
```

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality |
|------------|------------|----------------|--------------|-------------|-----------|-------------|
| infraction | 0 | PRIMARY | 1 | codeIndex | A | 107 |
| infraction | 1 | FK_idIndic | 1 | idIndic | A | 13 |
| infraction | 1 | FK_idUc | 1 | idUc | A | 9 |
| infraction | 1 | i_indexLibelle | 1 | codeIndex | A | 107 |
| infraction | 1 | i_indexLibelle | 2 | libelle | A | 107 |

| Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality |
|---------|------------|----------------|--------------|-------------|-----------|-------------|
| service | 0 | PRIMARY | 1 | idService | A | 1209 |
| service | 1 | FK_numDep | 1 | numDep | A | 105 |
| service | 1 | FK_idDirection | 1 | idDirection | A | 9 |
| service | 1 | FK_idAdm | 1 | idAdm | A | 2 |
| service | 1 | i_nomService | 1 | nomService | A | 1209 |

PROCÉDURE DE SAUVEGARDE AUTOMATIQUE DE LA BASE DE DONNÉES

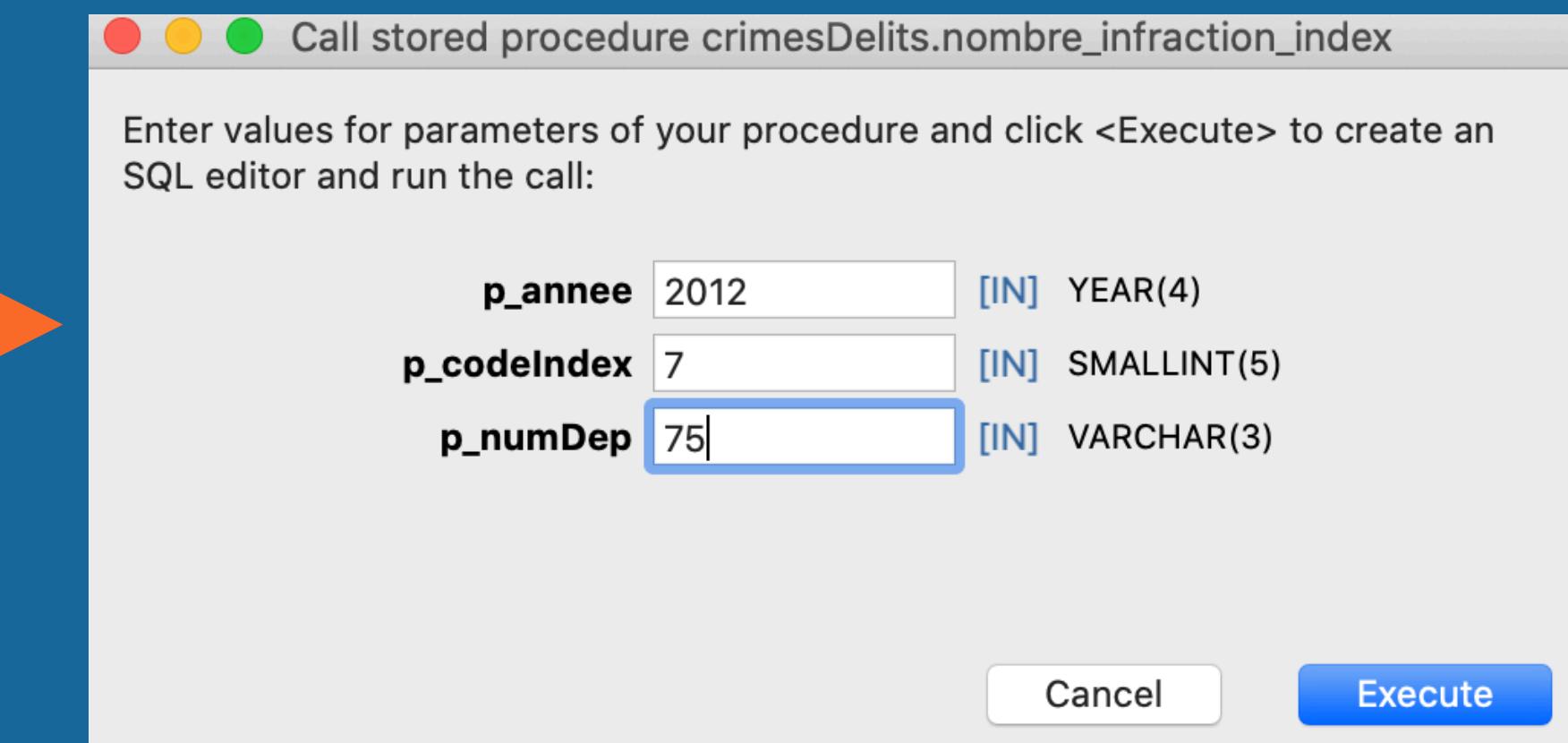
#Création d'un script bash backupOctoplus:

```
#!/bin/bash
# back auto base de donnee octoplus
cd /usr/local/mysql/bin;MYSQL_PWD='rootroot' ./mysqldump -u root --opt --databases octoplus > \
/Users/geraldbouget/dumps/local_octoplus_backup/octoplus_backup_$(date +\%Y\%m\%dT\%H\%M\%S).sql \
2>/Users/geraldbouget/dumps/local_octoplus_backup/octoplus_backup_LOG_$(date +\%Y\%m\%dT\%H\%M\%S).txt

>>crontab -e
0 0 * * 5 /Users/geraldbouget/bin/backupOctoplus
```

PROCÉDURES STOCKÉES

```
DELIMITER |
CREATE PROCEDURE `nombre_infraction_index`(
    p_annee YEAR(4),
    p_codeIndex SMALLINT(5),
    p_numDep VARCHAR(3))
BEGIN
    SELECT p.annee, p.codeIndex, libelle, s.numDep, sum(nbreInfracti... as NombreInfracti...
    FROM pointage as p
    JOIN service AS s ON s.idService=p.idService
    JOIN infraction as i ON i.codeIndex=p.codeIndex
    GROUP BY annee,p.codeIndex, s.numDep, libelle
    HAVING p.annee=p_annee
    AND numDep=p_numDep
    AND p.codeIndex=p_codeIndex;
END |
```



| annee | codeIndex | libelle | numDep | NombreInfracti... |
|-------|-----------|---|--------|-------------------|
| 2012 | 7 | Autres coups et blessures volontaires criminels ou correctionnels | 75 | 8664 |

VUES MATERIALISÉES POUR OPTIMISER REQUETES À PARTIR DE DASH

```
-- vue matérialisée pour query à partir de dash
CREATE TABLE query_dash_main
ENGINE = InnoDB
SELECT dp.numDep, dp.nomDep, ic.nomIndicLight, i.libelle, annee, sum(p.nbreInfractions) AS cumulInfraction,
d.nbrePopulation, d.densite, SUM((nbreInfractions/nbrePopulation)*1000) AS InfPour1000
FROM pointage AS p
JOIN service AS s ON p.idService=s.idService
JOIN infraction AS i ON i.codeIndex=p.codeIndex
JOIN indicateur AS ic ON ic.idIndic=i.idIndic
JOIN departement AS dp ON s.numDep=dp.numDep
JOIN demographie AS d ON dp.numDep=d.numDep
GROUP BY dp.numDep, dp.nomDep, ic.nomIndicLight, i.libelle, annee, d.nbrePopulation, d.densite;
```

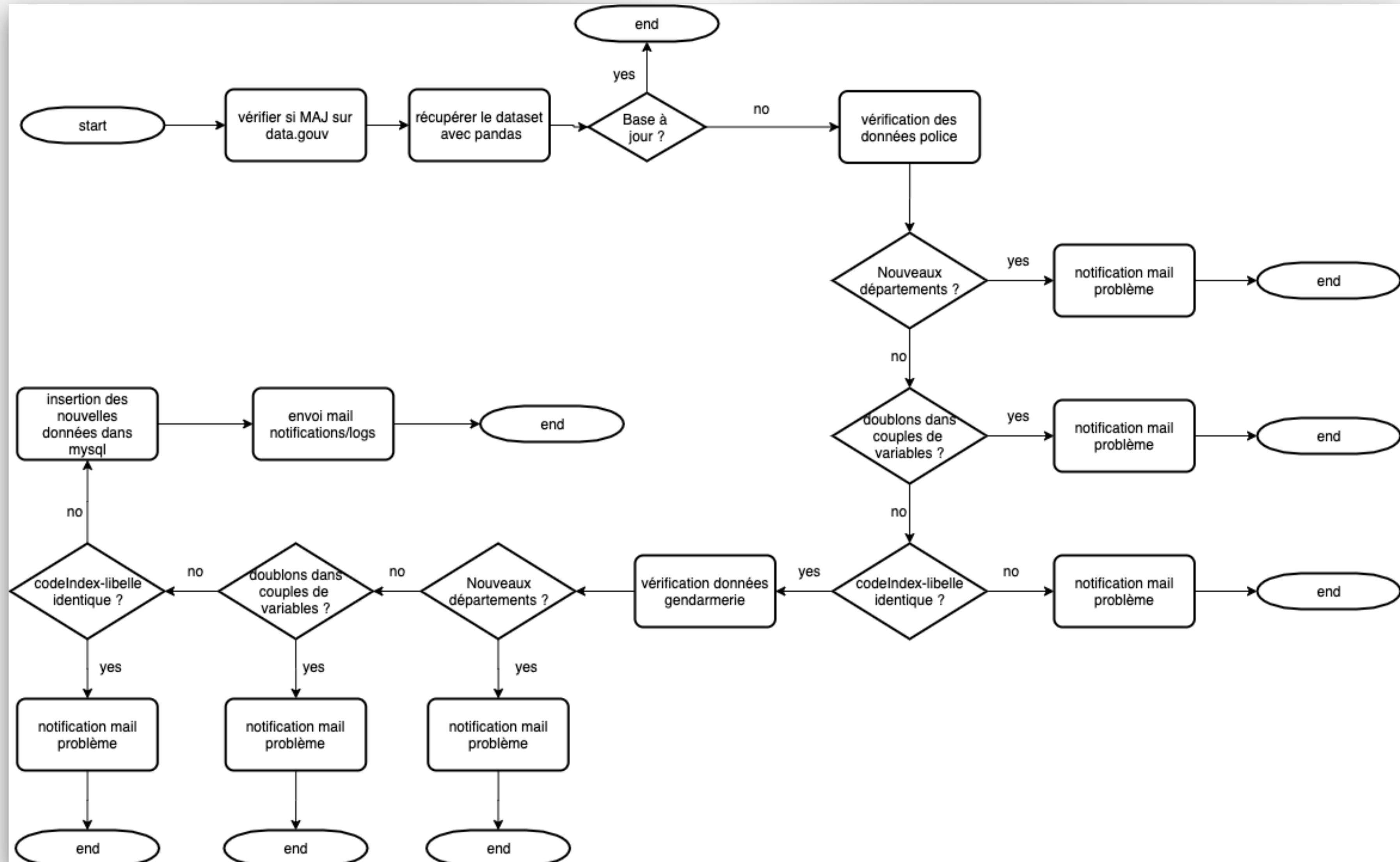
CRÉATION UTILISATEUR AVEC PRIVILEGES LIMITÉS

User spécifique sécurisée pour connexion à octuples sur AWS à partir de Heroku



```
-- create user query pour RDS octoplus --
CREATE USER 'heroku-user'@'%' IDENTIFIED BY 'XXXXXXXXXX' REQUIRE SSL;
GRANT SELECT
ON octoplus.*
TO 'heroku-user'@'%';
```

MISE À JOUR AUTOMATIQUE DE LA BASE DE DONNÉES - APPLICATION PYTHON



MISE À JOUR AUTOMATIQUE DE LA BASE DE DONNÉES - APPLICATION PYTHON

```
├── _main.py
├── app.log
├── connexionMySql.py
├── dashIssues.py
├── dashQueryUpdate.py
├── data
│   ├── gendarmerie_for_update.csv
│   └── police_for_update.csv
├── environment.yml
├── gendarmerieUpdate.py
├── issues.py
├── loadingData.py
├── maj_octoplus.txt
├── notes_automatisation.rtf
├── notifUpdate.py
├── onlineUpdateDate.py
└── policeUpdate.py
└── security.py
```

```
cd /Users/geraldbouget/SFR\ Cloud/Documents/
Simplon_DataDev_formation/Projet_final/
automatisation/
/Users/geraldbouget/opt/anaconda3/bin/python
_main.py
```



```
# MAJ auto de octoplus
0 0 * * 5 /Users/geraldbouget/bin/majocto 2>/Users/
geraldbouget/SFR\ Cloud/Documents/
Simplon_DataDev_formation/Projet_final/
automatisation/log_exec_main_auto.txt
```

EXPLOITATION ET VISUALISATION DES DONNÉES



Les résultats des analyses à suivre ne doivent pas donner lieu à des conclusions hâtives et ne reflètent la réalité que partiellement.

- Seuls les faits ayant fait l'objet d'une plainte sont pris en compte
- Les variations peuvent être le fait de raisons autres que l' augmentation ou diminution réelles de telle infraction.
- Plusieurs sources de données supplémentaires seraient nécessaires pour avoir une analyse plus précise.

DATA PROCESS



1. Connexion de la base de données SQL dans Python
2. Analyse à partir de requêtes sur la base SQL via l'instruction `pd.read_sql()`

1. Dashboard avec Dash
2. Visualisation des graphiques avec Plotly
3. Visualisation géographique avec mapbox/plotly.

1. déploiement BD sur RDS/AWS
2. Déploiement dashboard sur Heroku

APPLICATION DASHBOARD

```
class Map1:

    def __init__(self):
        self.MYSQL_USER = 'heroku-user'
        self.MYSQL_PASSWORD = [REDACTED]
        self.MYSQL_HOST_IP = 'octoplus-db.[REDACTED].amazonaws.com'
        self.MYSQL_DATABASE = 'octoplus'
        self.engine = create_engine('mysql+mysqlconnector://' + self.MYSQL_USER +
                                    ':' + self.MYSQL_PASSWORD + '@' + self.MYSQL_HOST_IP
                                    + '/' + self.MYSQL_DATABASE,
                                    connect_args={'auth_plugin': 'mysql_native_password',
                                    'ssl_ca': 'rds-ca-2019-root.pem'})

        self.conn = self.engine.connect()
```

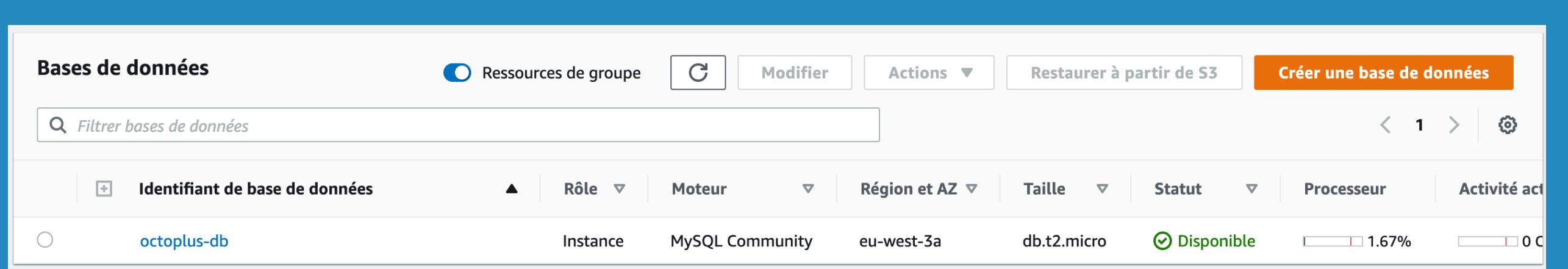
Structuring a Multi-Page App

Here's how to structure a multi-page app, where each app is contained in a separate file.

File structure:

```
- app.py
- index.py
- apps
  |-- __init__.py
  |-- app1.py
  |-- app2.py
```

DEPOIEMENT OCPLUS SUR RDS /HEROKU



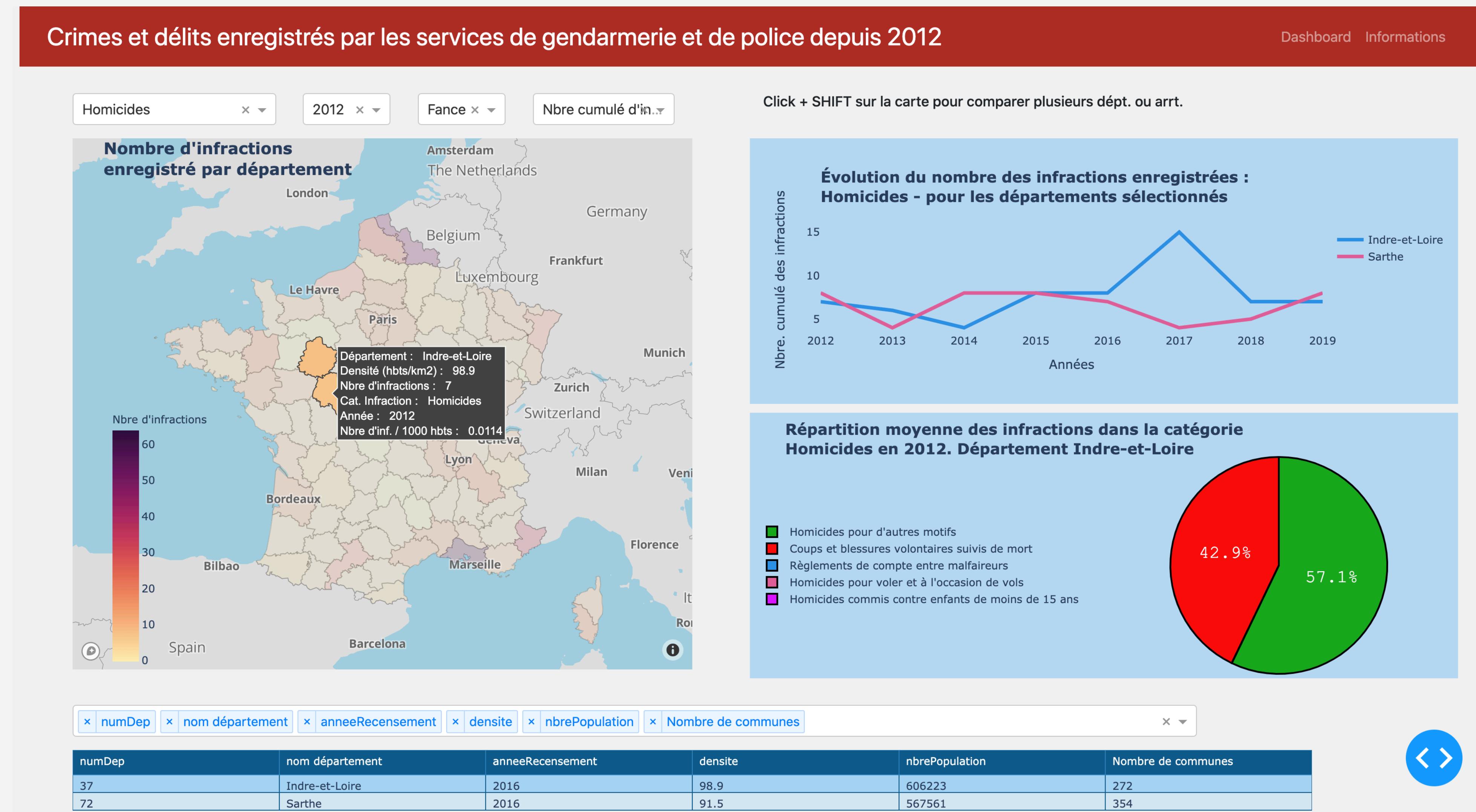
The screenshot shows the AWS RDS 'Bases de données' (Data Bases) page. It lists a single database instance named 'octoplus-db'. The instance is an 'Instance' of 'MySQL Community' type, located in the 'eu-west-3a' region, with a 'Taille' (Size) of 'db.t2.micro'. The 'Statut' (Status) is 'Disponible' (Available) with a green checkmark. The 'Processeur' (Processor) usage is at 1.67%, and the 'Activité active' (Active Activity) is at 0. The page includes a search bar, a 'Créer une base de données' (Create a database) button, and various filtering and sorting options.

1. Création d'un compte sur Heroku
2. Installer command line pour Heroku
3. Installer gunicorn
4. Exporter dans un fichier requirements.txt les modules importés dans mon environnement virtuel (conda) >> pip freeze > requirements.txt (modules installés avec pip sur conda) + environnement.yaml
5. Créer un fichier Procfile pour indiquer quel fichier doit être exécuté en premier
6. Téléchargement du fichier de certificat Amazon pour connexion SSL à partir de l'application
7. Upload du fichier certificat sur Heroku:

```
heroku config:set DATABASE_URL=<> mysql2://admin:<mot de passe>@octoplus-db.*****.amazonaws.com/octoplus?sslca=config/*****.pem >> -a octoplus-dash
```

+ commit avec fichier .pem à la racine
8. Script de connexion SSL pour assurer la sécurité des échanges entre l'application et la base de données sur RDS:
9. Création d'un utilisateur spécifique avec uniquement des priviléges 'SELECT' et l'obligation de se connecter en SSL.
10. Se servir de Git pour committer les derniers changement et 'push' sur le serveur Heroku pour exporter l'application :

DASHBOARD LOCAL/HEROKU



POUR ALLER PLUS LOIN

POINTS À AMÉLIORER :

- Ajouter des données socio/économiques à la base de données
- Ajouter des analyses
- Optimiser et simplifier code des applications
- Version multi-pages de la version en ligne du dashboard

MERCI POUR VOTRE ATTENTION !