

SOW El Hadji Dame

Master 2 geoter

Thématique : Le désert médical dans le bassin de vie du Loiret

1. Introduction sur la thématique

Les déserts médicaux représentent un enjeu majeur de santé publique en France. Ils sont définis comme « des espaces où le nombre de médecins (et plus généralement de professionnels de santé) pour 100 000 habitants est faible, et dans lequel les délais d'accès aux soins s'allongent du fait de l'augmentation du nombre de patients par médecin ». (Géoconfluences) Beaucoup de territoire que ce soient ruraux, urbains ou péri-urbains, sont confrontés à cette problématique.

Ainsi, ce travail consiste à créer une base de données permettant de mettre en avant le désert médical dans les bassins de vie du Loiret (département 45). Le découpage de la France « en bassins de vie » a été réalisé pour faciliter la compréhension de la structuration du territoire de la France métropolitaine. « Le bassin de vie est le plus petit territoire sur lequel les habitants ont accès aux équipements et services les plus courants : santé, services aux particuliers, commerce, enseignement, sports, loisirs et culture, transports ». (INSEE) L'accent est donc mis sur la répartition des professionnels de santé et l'accès aux équipements médicaux, faisant parties des enjeux majeurs pour garantir une couverture sanitaire équitable sur l'ensemble du territoire. Dans le département du Loiret, comme dans d'autres régions, certains bassins de vie sont mieux dotés en médecins, spécialistes entre autres, tandis que d'autres présentent des zones sous-dotées, ce qui peut impacter la qualité et la rapidité des soins.

Cette étude se concentre sur l'évaluation de la densité moyenne de six professions médicales (généralistes, spécialistes, chirurgiens-dentistes, kinésithérapeutes, infirmiers et Orthophonistes) dans les bassins de vie du Loiret en 2022 (exceptés : les bassins 41173, 77333, 89086 et 89419, exclus en raison de l'absence de données), ainsi que sur l'accessibilité aux équipements de santé. L'objectif est de structurer ces données dans une base relationnelle spatialisée, afin de faciliter l'analyse et la visualisation des disparités territoriales en matière de santé.

2. Description et sources des données :

La création de la base de données a nécessité la mobilisation de trois types de données notamment :

- Données sur l'accessibilité aux équipements : c'est un fichier csv nommé equip_acces et contenant : code_bassin , noms, nb_equipement et temps_acces
- Données de densité médicale : c'est un fichier csv avec plusieurs colonnes telles que code, nom, poptot ainsi le nombre et densité de professionnels de santé (médecins généralistes, spécialistes, chirurgiens-dentistes, kinésithérapeutes, infirmiers et Orthophonistes)
- Données sur la densité de sages femmes : un fichier csv(densite_sf) contenant num_bass_vie , popfemme16_59 , nb_sage_femme et dens_sf_pour100k
- Une couche shp appelé bassin_vie avec comme colonnes bv2022 et libbv2022 ainsi que geom

Ces données ont été téléchargées via :

Data.gouv.fr pour la couche shp des bassins de vie du Loiret : <https://www.data.gouv.fr/datasets/le-zonage-en-bassins-de-vie-de-2022-selon-linsee-dans-le-loiret/>

Le site de l'ARS centre val de loire pour les données sur les densités : <https://www.centre-val-de-loire.ars.sante.fr/media/40769/download?inline>

Et le site de l'INSEE pour les données concernant l'accès aux équipements : https://www.insee.fr/fr/statistiques/fichier/7457119/ce_ina_94.xlsx

3. Modélisation et structuration de la base de données

3.1 Présentation du modèle

Dans ce projet, la table de référence est bassin_vie, qui contient les informations géographiques et administratives de chaque bassin de vie du Loiret, identifié par la clé primaire bv2022. Cette table sert de référentiel pour toutes les analyses et les jointures avec les tables de faits.

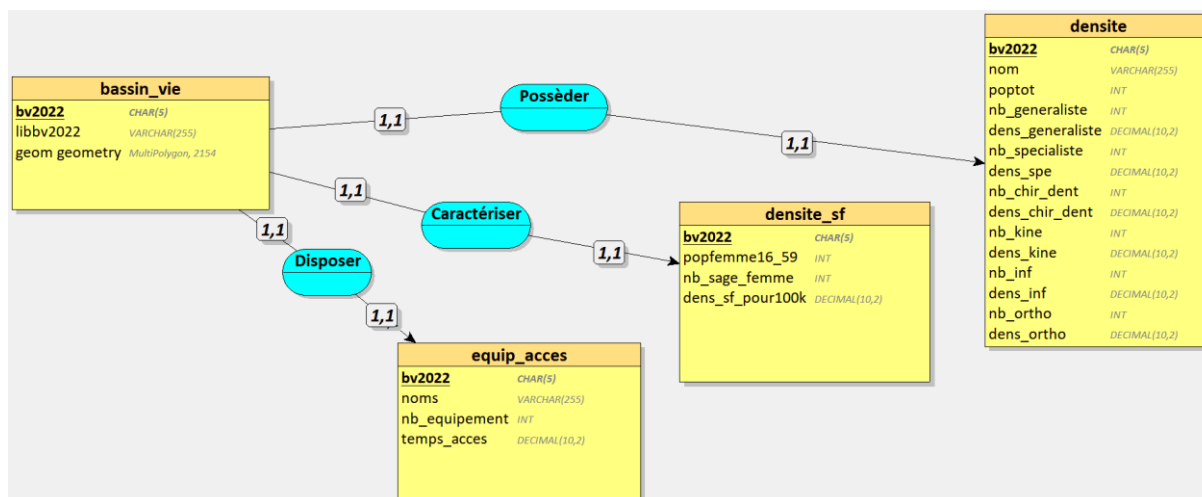
Les tables de faits densite, densite_sf et equip_acces regroupent respectivement les indicateurs de densité médicale par type de professionnel, la densité de sages-femmes et l'accès aux équipements. Dans chacune de ces tables, la colonne bv2022 joue un double rôle : elle est à la fois clé primaire, garantissant l'unicité de chaque bassin, et clé étrangère, référant à bassin_vie(bv2022), ce qui assure l'intégrité référentielle et supprime toute redondance.

Le modèle relationnel est ainsi entièrement normalisé et respecte la troisième forme normale (3FN). Les relations 1:1 entre la table de référence et les tables de faits permettent des jointures simples et efficaces, facilitant les analyses spatiales et la création de vues matérialisées ou de requêtes analytiques.

Cette structuration garantit que chaque bassin de vie est représenté dans toutes les tables de faits. Les valeurs de densité peuvent être nulles (0) pour certains bassins, tandis que d'autres présentent des densités supérieures ou égales à 1. Cette approche permet de réaliser des analyses complètes sur tous les bassins de vie du Loiret, sans perte d'information, même lorsque certaines densités sont faibles ou nulles.

3.2 Description des tables et MCD

| | | | |
|---|--|--|---|
| Table bassin_vie <ul style="list-style-type: none">• bv2022 (PK)• libbv2022• geom geometry (multipolygon,2154) | Table densite_sf <ul style="list-style-type: none">• bv2022 primary key references bassin_vie(bv2022)• reference bassin_vie(bv2022)• Popfemme16_59• Nb_sage_femme• Dens_sf_pour100k | Table accessibilite <ul style="list-style-type: none">• bv2022 primary key references bassin_vie(bv2022)• noms• nb_equipement• temps_acces | Table densite_medicale <ul style="list-style-type: none">• bv2022 primary key references bassin_vie(bv2022)• population• nb_generaliste, dens_generaliste• nb_specialiste, dens_spe• nb_chir_dent, dens_chir_dent• nb_kine, dens_kine• nb_inf, dens_inf• nb_ortho, dens_ortho |
|---|--|--|---|



4 – Requête et vue matérialisée

Cette requête permet de sélectionner les 5 bassins de vie du Loiret les moins couverts en sage-femmes. Il utilise une jointure interne (JOIN) entre la table spatiale bassin_vie et la table de faits densite_sf sur la colonne commune bv2022 pour relier les données démographiques aux informations géographiques.

```

SELECT
  b.bv2022 AS code_bassin_vie,
  b.libbv2022 AS nom_bassin_vie,
  dsf.popfemme16_59,
  dsf.nb_sage_femme,
  REPLACE(dsf.dens_sf_pour100k::TEXT, ',', '.')::NUMERIC AS densite_sage_femmes
FROM bassin_vie b
JOIN densite_sf dsf ON b.bv2022 = dsf.bv2022
WHERE
  REPLACE(dsf.dens_sf_pour100k::TEXT, ',', '.')::NUMERIC > 0
ORDER BY densite_sage_femmes ASC
LIMIT 5;

```

| | code_bassin_vie character (5) | nom_bassin_vie character varying (80) | popfemme16_59 integer | nb_sage_femme integer | densite_sage_femmes numeric |
|---|----------------------------------|--|--------------------------|--------------------------|--------------------------------|
| 1 | 45155 | Gien | 5923 | 1 | 16.90 |
| 2 | 45173 | Jargeau | 5410 | 1 | 18.50 |
| 3 | 45115 | Courtenay | 3893 | 1 | 25.70 |
| 4 | 45053 | Briare | 3702 | 1 | 27.00 |
| 5 | 41106 | Lamotte-Beuvron | 3595 | 1 | 27.80 |

La requête ci-dessous crée une vue matérialisée nommée vue_score_medical_mat pour stocker de manière permanente un indicateur de dotation médicale. Il calcule la densité moyenne de six professions de santé (généralistes, spécialistes, chirurgiens-dentistes, kinésithérapeutes, infirmiers et Orthophonistes) par bassin de vie en joignant les tables bassin_vie et densite

```
CREATE MATERIALIZED VIEW vue_score_medical_mat AS
```

```
SELECT
```

```
  b.bv2022 AS code_bassin_vie,
  b.libbv2022 AS nom_bassin,
  d.poptot AS population,
  d.dens_generaliste,
  d.dens_spe,
  d.dens_chir_dent,
  d.dens_kine,
  d.dens_inf,
  d.dens_ortho,
  (
    COALESCE(d.dens_generaliste, 0) +
    COALESCE(d.dens_spe, 0) +
    COALESCE(d.dens_chir_dent, 0) +
    COALESCE(d.dens_kine, 0) +
    COALESCE(d.dens_inf, 0) +
    COALESCE(d.dens_ortho, 0)
  ) / 6.0 AS score_medical,
```

```
  b.geom
```

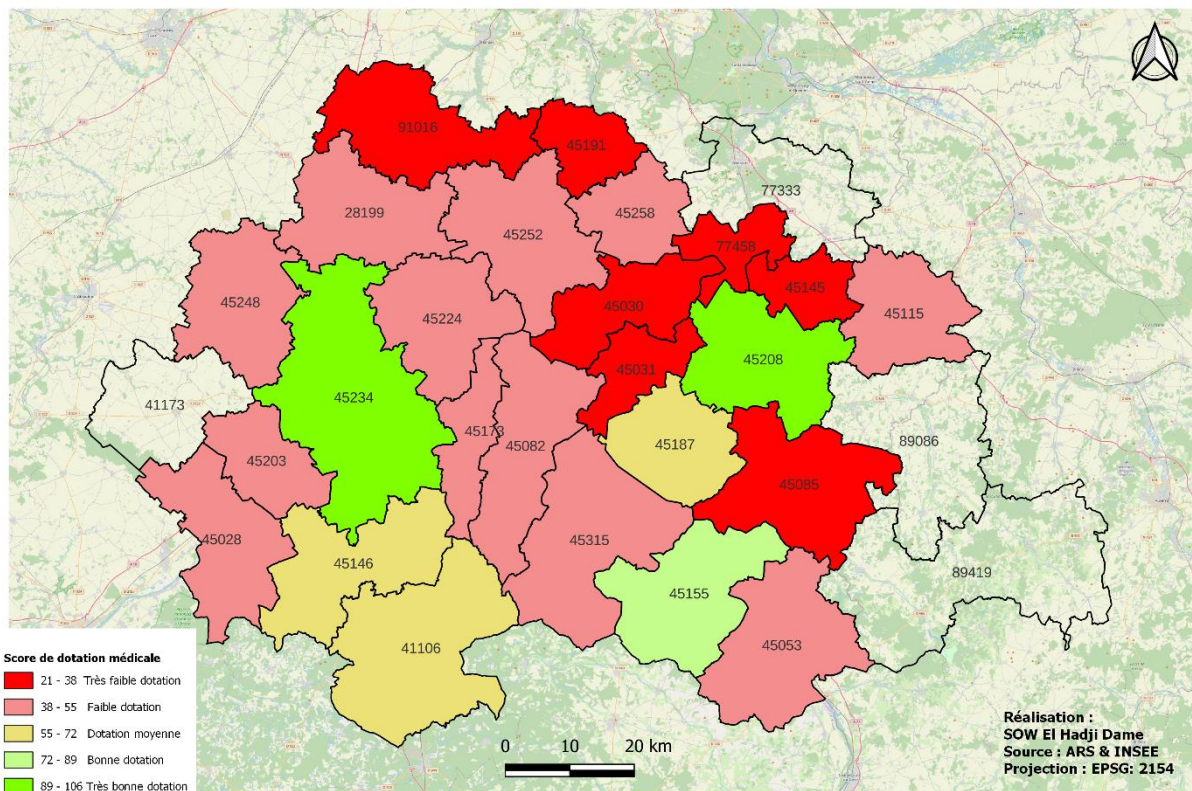
```
FROM bassin_vie b
```

```
JOIN densite d
```

```
ON b.bv2022 = d.bv2022;
```

| code_bassin_vie | nom_bassin | population | dens_generaliste | dens_spe | dens_chir_dent | dens_kine | dens_inf | dens_ortho | score_medical |
|-----------------|--------------------|------------|------------------|----------|----------------|-----------|----------|------------|---------------|
| 28199 | Janville-en-Bea... | 12722 | 39,30 | 15,70 | 7,90 | 94,30 | 70,70 | 15,70 | 40,6 |

Densité moyenne des six professions médicales par bassin de vie dans le Loiret (2022)



5 – Mise à la disposition dans un Dump :

```
"C:\Program Files\PostgreSQL\17\bin\pg_dump.exe" -Fc -v -U postgres -d Santeorleans -f
"C:\Users\Utilisateur\Desktop\Dump\Dump_Santeorleans.dump"
```

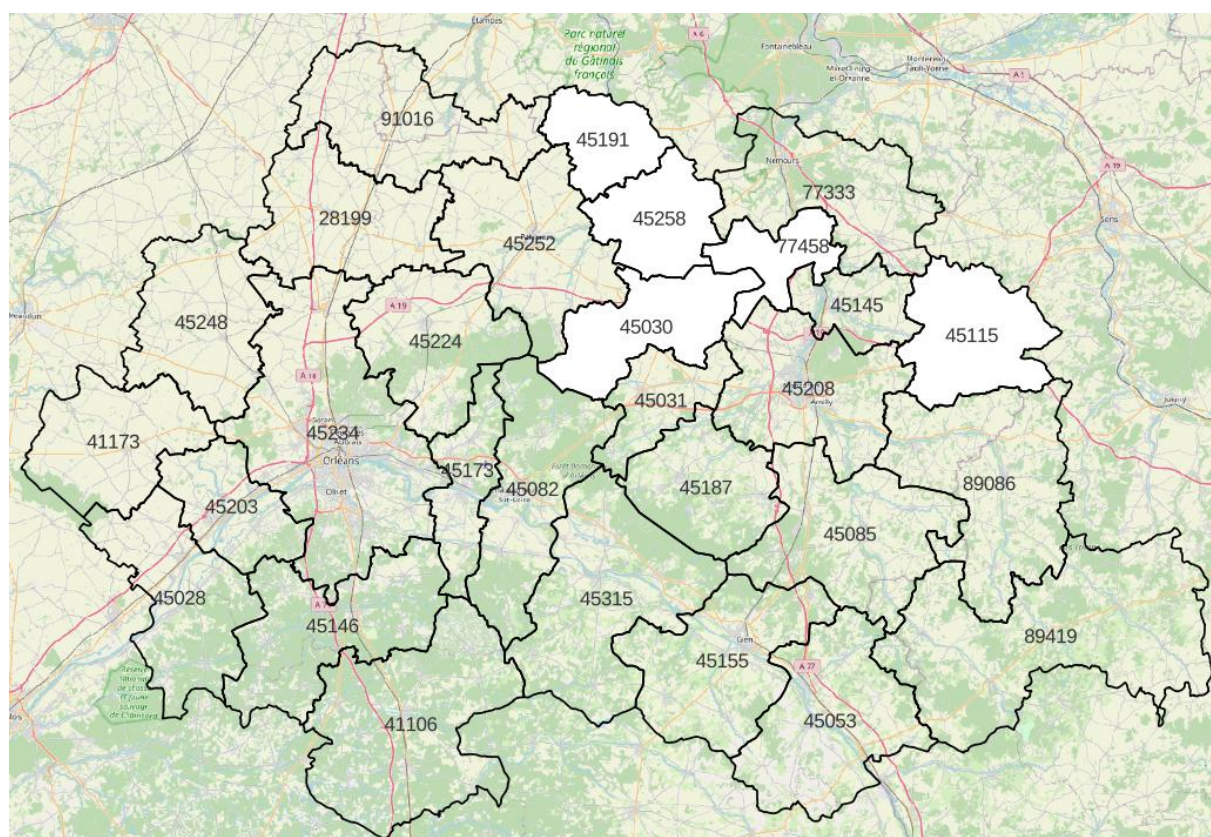
Mot de passe : Mon_mot_de_passe

Annexe 1 : autre requête

Cette requête permet d'identifier et d'enregistrer les bassins de vie où l'accès aux équipements est considéré comme défavorable (moins de 15 équipements et temps d'accès supérieur ou égal à 10 minutes) en joignant les tables bassin_vie et equip_acces.

```
CREATE MATERIALIZED VIEW vue_acces_defavorable_mat AS
SELECT
  b.bv2022 AS code_bassin,
  b.libbv2022 AS nom_bassin,
  a.nb_equipement,
  a.temps_acces,
  b.geom
FROM bassin_vie b
JOIN equip_acces a
  ON b.bv2022 = a.bv2022
WHERE a.nb_equipement <= 15
AND a.temps_acces >= 10;
```

| code_bassin | nom_bassin | nb_equipement | temps_acces |
|-------------|-------------------|---------------|-------------|
| 45030 | Beaune-la-Rola... | 14 | 18,70 |
| 45115 | Courtenay | 13 | 20,70 |
| 45191 | Le Malesherbois | 14 | 16,00 |
| 45258 | Puiseaux | 13 | 17,60 |
| 77458 | Souppes-sur-Lo... | 15 | 13,50 |



Annexe 2 : Dump

Le Dump et son résultat dans l'invite de commande

```
C:\Users\Utilisateur>"C:\Program Files\PostgreSQL\17\bin\pg_dump.exe" -Fc -v -U postgres -d Santeorleans -f "C:\Users\Utilisateur\Desktop\Dump\Dump_Santeorleans.dump"
Mot de passe :

pg_dump: le dernier OID interne est 16383
pg_dump: lecture des extensions
pg_dump: identification des membres d'extension
pg_dump: lecture des schémas
pg_dump: lecture des tables utilisateur
pg_dump: lecture des fonctions utilisateur
pg_dump: lecture des types utilisateur
pg_dump: lecture des langages procéduraux
pg_dump: lecture des fonctions d'agrégats utilisateur
pg_dump: lecture des opérateurs utilisateur
pg_dump: lecture des méthodes d'accès définis par les utilisateurs
pg_dump: lecture des classes d'opérateurs utilisateur
pg_dump: lecture des familles d'opérateurs utilisateur
pg_dump: lecture des analyseurs utilisateur pour la recherche plein texte
pg_dump: lecture des modèles utilisateur pour la recherche plein texte
pg_dump: lecture des dictionnaires utilisateur pour la recherche plein texte
pg_dump: lecture des configurations utilisateur pour la recherche plein texte
pg_dump: lecture des wrappers de données distantes utilisateur
pg_dump: lecture des serveurs distants utilisateur
pg_dump: lecture des droits par défaut
pg_dump: lecture des collationnements utilisateurs
pg_dump: lecture des conversions utilisateur
pg_dump: lecture des conversions de type
pg_dump: lecture des transformations
pg_dump: lecture des informations d'héritage des tables
pg_dump: lecture des triggers sur événement
pg_dump: recherche des tables d'extension
pg_dump: recherche des relations d'héritage
pg_dump: lecture des informations de colonnes des tables intéressantes
pg_dump: recherche des expressions par défaut de la table
pg_dump: recherche des contraintes CHECK de la table
pg_dump: marquage des colonnes héritées dans les sous-tables
pg_dump: lecture des données de partitionnement
pg_dump: lecture des index
```

```
pg_dump: décrit les index des tables partitionnées
pg_dump: lecture des statistiques étendues
pg_dump: lecture des contraintes
pg_dump: lecture des triggers
pg_dump: lecture des règles de réécriture
pg_dump: lecture des politiques
pg_dump: lecture des politiques de sécurité au niveau ligne
pg_dump: lecture des publications
pg_dump: lecture des appartenances aux publications des tables
pg_dump: lecture des appartenances aux publications des schémas
pg_dump: lecture des souscriptions
pg_dump: lecture des appartenances aux souscriptions des tables
pg_dump: lecture des « Large Objects »
pg_dump: lecture des données de dépendance
pg_dump: encodage de la sauvegarde = UTF8
pg_dump: sauvegarde de « standard_conforming_strings = on »
pg_dump: sauvegarde de « search_path = »
pg_dump: sauvegarde de la définition de la base de données
pg_dump: sauvegarde du contenu de la table « public.bassin_vie »
pg_dump: sauvegarde du contenu de la table « public.densite »
pg_dump: sauvegarde du contenu de la table « public.densite_backup »
pg_dump: sauvegarde du contenu de la table « public.densite_sf »
pg_dump: sauvegarde du contenu de la table « public.densite_sf_backup »
pg_dump: sauvegarde du contenu de la table « public.equip_acces »
pg_dump: sauvegarde du contenu de la table « public.equip_acces_backup »
pg_dump: sauvegarde du contenu de la table « public.spatial_ref_sys »
pg_dump: sauvegarde du contenu de la table « topology.topology »
pg_dump: sauvegarde du contenu de la table « topology.layer »
```

Annexe 3 : Les contraintes d'intégrité

```
674 SELECT conname, contype, pg_get_constraintdef(oid)
675 FROM pg_constraint
676 WHERE conrelid = 'bassin_vie'::regclass;
```

Data Output Messages Notifications

Showing rows: 1 to 1 Page No: 1

| | conname name | contype "char" | pg_get_constraintdef text |
|---|-----------------|-------------------|------------------------------|
| 1 | bassin_vie_pk | p | PRIMARY KEY (bv2022) |

```

671 SELECT conname, contype, pg_get_constraintdef(oid)
672 FROM pg_constraint
673 WHERE conrelid = 'equip_acces'::regclass;

```

Data Output Messages Notifications

| | conname name | contype "char" | pg_get_constraintdef text |
|---|-------------------------|-------------------|--|
| 1 | fk_equip_acces_bv2022 | f | FOREIGN KEY (bv2022) REFERENCES bassin_vie(bv2022) |
| 2 | equip_acces_pkey | p | PRIMARY KEY (bv2022) |
| 3 | equip_acces_bv2022_fkey | f | FOREIGN KEY (bv2022) REFERENCES bassin_vie(bv2022) ON DELETE CASCADE |

```

674 SELECT conname, contype, pg_get_constraintdef(oid)
675 FROM pg_constraint
676 WHERE conrelid = 'densite'::regclass;

```

Data Output Messages Notifications

| | conname name | contype "char" | pg_get_constraintdef text |
|---|---------------------|-------------------|--|
| 1 | fk_densite_bv2022 | f | FOREIGN KEY (bv2022) REFERENCES bassin_vie(bv2022) |
| 2 | densite_pk | p | PRIMARY KEY (bv2022) |
| 3 | densite_bv2022_fkey | f | FOREIGN KEY (bv2022) REFERENCES bassin_vie(bv2022) ON DELETE CASCADE |

```

674 SELECT conname, contype, pg_get_constraintdef(oid)
675 FROM pg_constraint
676 WHERE conrelid = 'densite_sf'::regclass;

```

Data Output Messages Notifications

| | conname name | contype "char" | pg_get_constraintdef text |
|---|------------------------|-------------------|--|
| 1 | fk_densite_sf_bv2022 | f | FOREIGN KEY (bv2022) REFERENCES bassin_vie(bv2022) |
| 2 | densite_sf_pkey | p | PRIMARY KEY (bv2022) |
| 3 | densite_sf_bv2022_fkey | f | FOREIGN KEY (bv2022) REFERENCES bassin_vie(bv2022) ON DELETE CASCADE |

Annexe 4 : Les différents index

Les index spatiaux :

```
CREATE INDEX idx_bassin_vie_geom ON bassin_vie USING GIST (geom);
```

```
CREATE UNIQUE INDEX idx_bassin_vie_pk ON bassin_vie (bv2022);
```

| | tablename name | indexname name | indexdef text |
|---|-------------------|---------------------|---|
| 1 | bassin_vie | bassin_vie_pk | CREATE UNIQUE INDEX bassin_vie_pk ON public.bassin_vie USING btree (bv2022) |
| 2 | bassin_vie | idx_bassin_vie_geom | CREATE INDEX idx_bassin_vie_geom ON public.bassin_vie USING gist (geom) |
| 3 | bassin_vie | idx_bassin_vie_pk | CREATE UNIQUE INDEX idx_bassin_vie_pk ON public.bassin_vie USING btree (bv2022) |

Index sur les Tables de Faits :

```
CREATE UNIQUE INDEX idx_densite_pk ON densite (bv2022);
```

```
CREATE UNIQUE INDEX idx_densite_sf_pk ON densite_sf (bv2022);
```

CREATE UNIQUE INDEX idx_equip_acces_pk ON equip_acces (bv2022);

| | tablename name | indexname name | indexdef text |
|---|-------------------|----------------------|---|
| 1 | bassin_vie | bassin_vie_pk | CREATE UNIQUE INDEX bassin_vie_pk ON public.bassin_vie USING btree (bv2022) |
| 2 | bassin_vie | idx_bassin_vie_ge... | CREATE INDEX idx_bassin_vie_geom ON public.bassin_vie USING gist (geom) |
| 3 | bassin_vie | idx_bassin_vie_pk | CREATE UNIQUE INDEX idx_bassin_vie_pk ON public.bassin_vie USING btree (bv2022) |
| 4 | densite | idx_densite_pk | CREATE UNIQUE INDEX idx_densite_pk ON public.densite USING btree (bv2022) |
| 5 | densite | densite_pk | CREATE UNIQUE INDEX densite_pk ON public.densite USING btree (bv2022) |
| 6 | densite_sf | densite_sf_pkey | CREATE UNIQUE INDEX densite_sf_pkey ON public.densite_sf USING btree (bv2022) |
| 7 | densite_sf | idx_densite_sf_pk | CREATE UNIQUE INDEX idx_densite_sf_pk ON public.densite_sf USING btree (bv2022) |
| 8 | equip_acces | idx_equip_acces_pk | CREATE UNIQUE INDEX idx_equip_acces_pk ON public.equip_acces USING btree (bv2022) |
| 9 | equip_acces | equip_acces_pkey | CREATE UNIQUE INDEX equip_acces_pkey ON public.equip_acces USING btree (bv2022) |