

## **SAE- BD : Bornes de recharge de véhicules électriques en Pays de la Loire**

Groupe : Souchet Thomas, Lamothe Pol, Arnaud Kyllian (TP 1-2)

### **Sommaire**

- I. Introduction - p.1
- II. Importation des données - p.1-2
- III. Corrections des données - p.2-4
- IV. Décomposition en sous-tables - p.4-8
- V. Répartition des tables par département - p.8-11
- VI. Requêtes - p.12-21
- VII. Vues - p.21-28
- VIII. Préparation d'un second jeu de données - p.28-29
- IX. Conclusion - p.30

### **I - Introduction**

Ce projet consiste à exploiter des données libres concernant les bornes de recharge de véhicules électriques en Pays de la Loire. Il vise à étudier et à gérer les données relatives aux infrastructures de recharge réparties dans la région. Les données proviennent de la plateforme régionale des Pays de la Loire : [Open Data](#), cette plateforme est utilisée pour que toutes les collectivités territoriales puissent mettre en commun leur données et permettre à tout le monde d'y accéder facilement de manière harmonisée. Dans ce projet, chaque étudiant aura la responsabilité des données d'un ou plusieurs départements. Ce rapport détaille toutes les étapes nécessaires à l'exploitation des données en commençant par l'importation et la correction puis la séparation en plusieurs tables puis l'exploitation des résultats. Enfin, un second jeu de données sera aussi utilisé pour être lui aussi préparé à l'exploitation.

#### **Source des données sur les bornes de recharge :**

[https://data.paysdelaloire.fr/explore/dataset/234400034\\_-bornes-de-recharge/export/?disjunctive.commune&disjunctive.puissance\\_nominale](https://data.paysdelaloire.fr/explore/dataset/234400034_-bornes-de-recharge/export/?disjunctive.commune&disjunctive.puissance_nominale)

### **II - Importation des données**

L'importation des données sur les bornes de recharge pour véhicules électriques se fait depuis le fichier CSV fourni. Il contient principalement les données suivantes : liste des aménageurs, l'adresse où se trouve la borne de recharge électrique, puissance de la prise

et le type de prise, longitude et latitude correspondent au lieu où est installée la prise et le nom de la commune.

Démarche :

- importation du fichier CSV avec Oracle SQL Developer
- sélection du délimiteur (",")
- choix des colonnes
- correction des noms de colonnes et types

Les colonnes sélectionnées sont les suivantes :

- nom\_amenageur
- siren\_amenageur
- contact\_amenageur
- nom\_operateur
- contact\_operateur
- telephone operateur
- nom\_enseigne
- nom\_station
- adresse\_station
- Nombre points de charge
- date\_mise\_en\_service
- Date de mise à jour
- datagouv\_organization\_or\_owner
- libelle\_region
- code\_region
- libelle\_departement
- libelle\_commune
- tranche\_puissance
- implantation\_station
- puissance\_nominale
- departement
- code\_insee\_commune

Les espaces dans le nom des colonnes ont été supprimés et les types ont été modifiés pour certaines avec des VARCHAR2 de 128 caractères ou des NUMBER de 38 chiffres. Finalement, la table obtenue est en première forme normale et contient 3723 lignes.

### III - Corrections des données

Avant la décomposition de la table en plusieurs tables, les données doivent être mises à jour pour les rendre exploitables, pour cela il faut limiter au maximum la présence des *null*, pour les besoins de la SAE certaines données seront inventées. Chacun des membres du groupe fait des requêtes *update* sur la table principale. Les requêtes peuvent être des corrections, des ajouts de données depuis d'autres bases et des créations de données.

*/\* Ajout des CODES INSEE depuis basetd \*/*

```
UPDATE stations br
SET code_insee_commune = (
    SELECT MAX(c.code_insee)
    FROM basetd.commune c
    WHERE c.nom_commune = br.libelle_commune
) WHERE EXISTS (
    SELECT 1
    FROM basetd.commune c
    WHERE c.nom_commune = br.libelle_commune
);
```

*/\* Ajout des NUMÉROS DE DÉPARTEMENT en fonction du nom du département \*/*

```
UPDATE STATIONS
SET department = 44
where LIBELLE_DEPARTEMENT = 'Loire-Atlantique';
```

```
UPDATE STATIONS
SET department = 49
where LIBELLE_DEPARTEMENT = 'Maine-et-Loire';
```

```
UPDATE STATIONS
SET department = 85
where LIBELLE_DEPARTEMENT = 'Vendée';
```

```
UPDATE STATIONS
SET department = 72
where LIBELLE_DEPARTEMENT = 'Sarthe';
```

```
UPDATE STATIONS
SET department = 53
where LIBELLE_DEPARTEMENT = 'Mayenne';
```

*/\* Création des DATE DE MISE EN SERVICE, 15 jours avant la date de mise à jour \*/*

```
update stations
set date_mise_en_service = (
    select MAX(date_mise_jour - 15)
    from stations
)
where date_mise_en_service is null;
```

*/\* Création des NUMÉROS DE TÉLÉPHONES aléatoirement, une table intermédiaire est utilisée pour permettre à un opérateur d'avoir le même numéro de téléphone \*/*

```
create table temptel as select distinct TELEPHONE_OPERATEUR, NOM_OPERATEUR from STATIONS where TELEPHONE_OPERATEUR is null;
```

```
update temptel T
set TELEPHONE_OPERATEUR = TO_CHAR(FLOOR(DBMS_RANDOM.VALUE(1, 10)), 'FM00')
|| TO_CHAR(FLOOR(DBMS_RANDOM.VALUE(10000000, 99999999)))
where TELEPHONE_OPERATEUR is NULL;
```

```
update STATIONS set TELEPHONE_OPERATEUR =
(select t.TELEPHONE_OPERATEUR from temptel t where t.NOM_OPERATEUR =
STATIONS.NOM_OPERATEUR) where STATIONS.TELEPHONE_OPERATEUR is null;
drop table temptel;
```

*/\* Ajout des NOM AMÉNAGEUR à partir des nom de domaines des adresses mail, utilisation d'une expression régulière \*/*

```
update STATIONS set nom_amenageur = REGEXP_SUBSTR(contact_amenageur, '@([^\.]*)\.',
1, 1, NULL, 1) where nom_amenageur is null ;
```

*/\* Ajout des NOM OPÉRATEUR à partir des nom de domaines des adresses mail, utilisation d'une expression régulière \*/*

```
update STATIONS set nom_operateur = REGEXP_SUBSTR(contact_operateur, '@([^\.]*)\.', 1, 1,
NULL, 1) where nom_amenageur is null ;
```

*/\* Création des N° de SIREN aléatoirement , une table intermédiaire est utilisée pour permettre à un aménageur d'avoir le même numéro de SIREN \*/*

```
UPDATE S2A06A.STATIONS SET SIREN_AMENAGEUR = null WHERE NOM_AMENAGEUR =
'RossiniEnergy'
```

```
create view tempSiren as (select distinct nom_amenageur,siren_amenageur from STATIONS
where siren_amenageur is null);
```

```
create table sirencopy (SIREN_AMENAGEUR NUMBER(38), NOM_AMENAGEUR
VARCHAR2(128));
```

```
insert into sirencopy (SIREN_AMENAGEUR, NOM_AMENAGEUR) select
SIREN_AMENAGEUR, NOM_AMENAGEUR from tempsiren;
```

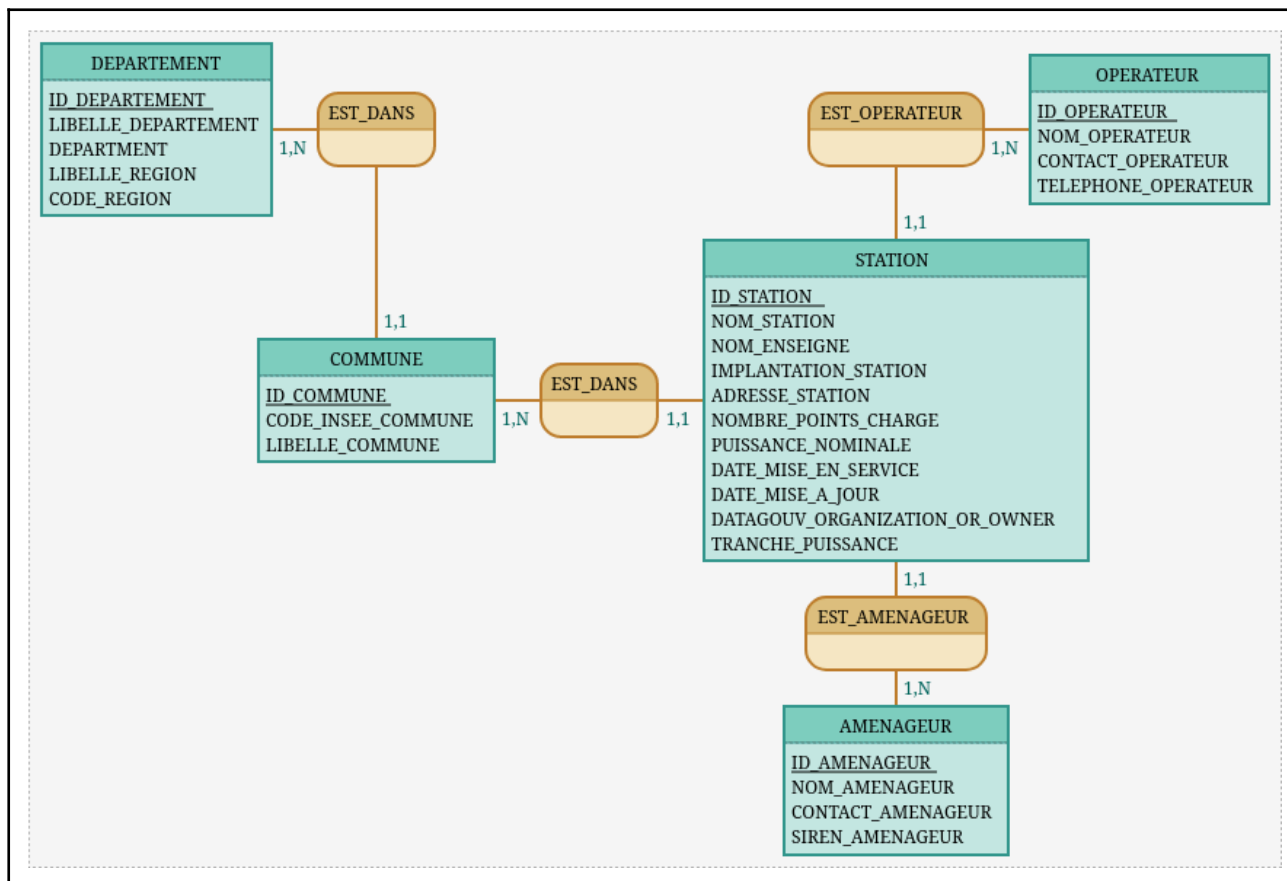
```
update sirencopy set siren_amenageur =
round(DBMS_RANDOM.VALUE(1000000000,9999999999));
```

```
update STATIONS set siren_amenageur =
(select siren_amenageur from sirencopy where NOM_AMENAGEUR =
STATIONS.NOM_AMENAGEUR) where siren_amenageur is null;
```

## IV - Décomposition en sous-tables

La table va maintenant être décomposée en plusieurs sous-table pour que l'ensemble puisse être en troisième forme normale. L'ensemble de ces tables globales sont sur un

seul compte : S2A06A (Thomas). Le schéma entité-association :



### Schéma relationnel :

- **STATIONS**(id\_station, nom\_station, nom\_enseigne, implantation\_station, adresse\_station, nombre\_points\_charge, puissance\_nominale, date\_mise\_en\_service, date\_mise\_a\_jour, datagouv\_organization\_or\_owner, tranche\_puissance, id\_amenageur, id\_operateur, id\_commune)
- **AMENAGEUR**(id\_amenageur, nom\_amenageur, contact\_amenageur, siren\_amenageur)
- **OPERATEUR**(id\_operateur, nom\_operateur, contact\_operateur, telephone\_operateur)
- **COMMUNE**(id\_commune, code\_insee\_commune, libelle\_commune, id\_departement)
- **DEPARTEMENT**(id\_departement, libelle\_departement, departement, libelle\_region, code\_region)

Les tables sont créées avec des id en auto-increment à partir de la table globale, les clés sont ensuite ajoutées puis les colonnes sont supprimées de la table principale.

**/\* Création de la table AMENAGEUR et insertion \*/**

```
Create table AMENAGEUR(  
id_Amenageur NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),  
NOM_AMENAGEUR varchar2(128),  
CONTACT_AMENAGEUR varchar2(128),  
SIREN_AMENAGEUR number(38)  
);
```

```
Insert into AMENAGEUR (nom_amenageur, contact_amenageur, siren_amenageur)  
select distinct NOM_AMENAGEUR, CONTACT_AMENAGEUR, SIREN_AMENAGEUR  
from STATIONS  
where not (NOM_AMENAGEUR is NULL and CONTACT_AMENAGEUR is NULL and  
SIREN_AMENAGEUR is null);
```

**/\* Création de la table OPÉRATEUR et insertion \*/**

```
CREATE TABLE OPEARATEUR(  
id_OPERATEUR NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by  
1),  
NOM_OPERATEUR varchar2(128),  
CONTACT_OPERATEUR varchar2(128),  
TELEPHONE_OPERATEUR varchar2(128)  
);
```

```
Insert into OPEARATEUR (NOM_OPERATEUR, CONTACT_OPERATEUR,  
TELEPHONE_OPERATEUR)  
select distinct NOM_OPERATEUR, CONTACT_OPERATEUR, TELEPHONE_OPERATEUR  
from STATIONS  
where not (NOM_OPERATEUR is NULL and CONTACT_OPERATEUR is NULL and  
TELEPHONE_OPERATEUR is null);
```

**/\* Création de la table COMMUNE et insertion \*/**

```
CREATE TABLE COMMUNE (  
ID_COMMUNE NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),  
CODE_INSEE_COMMUNE NUMBER(38),  
LIBELLE_COMMUNE VARCHAR2(128),  
ID_DEPARTMENT NUMBER,  
);
```

```
Insert into COMMUNE (CODE_INSEE_COMMUNE,LIBELLE_COMMUNE)  
select distinct CODE_INSEE_COMMUNE, LIBELLE_COMMUNE  
from STATIONS;
```

**/\* Création de la table DÉPARTEMENT et insertion \*/**

```
CREATE TABLE DEPARTMENT(  
ID_DEPARTMENT NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by  
1,  
LIBELLE_DEPARTMENT VARCHAR2(128),  
DEPARTMENT VARCHAR2(128),  
LIBELLE_REGION VARCHAR2(128),  
CODE_REGION NUMBER(38)  
);
```

```
INSERT INTO DEPARTMENT  
(LIBELLE_DEPARTMENT,DEPARTMENT,LIBELLLE_REGION,CODE_REGION)  
SELECT DISTINCT LIBELLE_DEPARTMENT,DEPARTMENT,LIBELLE_REGION,CODE_REGION  
FROM STATIONS;
```

*/\* Ajout des colonnes pour les clés étrangères \*/*

```
ALTER TABLE STATIONS ADD ID_AMENAGEUR NUMBER;  
ALTER TABLE STATIONS ADD ID_OPERATEUR NUMBER;  
ALTER TABLE STATIONS ADD ID_COMMUNE NUMBER;
```

*/\* Remplissage des clés étrangères \*/*

```
UPDATE S2A06A.STATIONS table1 SET table1.ID_AMENAGEUR = (select ID_AMENAGEUR  
FROM S2A06A.AMENAGEUR WHERE table1.NOM_AMENAGEUR =  
S2A06A.AMENAGEUR.NOM_AMENAGEUR AND table1.SIREN_AMENAGEUR =  
S2A06A.AMENAGEUR.SIREN_AMENAGEUR)
```

```
UPDATE S2A06A.STATIONS table1 SET table1.ID_OPERATEUR = (select ID_OPERATEUR  
FROM S2A06A.OPERATEUR WHERE table1.NOM_OPERATEUR =  
S2A06A.OPERATEUR.NOM_OPERATEUR AND table1.CONTACT_OPERATEUR =  
S2A06A.OPERATEUR.CONTACT_OPERATEUR AND table1.TELEPHONE_OPERATEUR =  
S2A06A.OPERATEUR.TELEPHONE_OPERATEUR)
```

```
UPDATE S2A06A.STATIONS table1 SET table1.ID_COMMUNE = (select ID_COMMUNE FROM  
S2A06A.COMMUNE WHERE table1.CODE_INSEE_COMMUNE =  
S2A06A.COMMUNE.CODE_INSEE_COMMUNE AND table1.LIBELLE_COMMUNE =  
S2A06A.COMMUNE.LIBELLE_COMMUNE)
```

```
UPDATE S2A06A.COMMUNE C  
SET C.ID_DEPARTMENT = (SELECT ID_DEPARTMENT FROM S2A06A.DEPARTMENT D  
WHERE D.DEPARTMENT = SUBSTR(C.CODE_INSEE_COMMUNE, 1, 2));
```

*/\* Suppression des colonnes de la table principale \*/*

```
ALTER TABLE s2a06a.STATIONS DROP (NOM_OPERATEUR,CONTACT_OPERATEUR,  
TELEPHONE_OPERATEUR,  
nom_amenageur,contact_amenageur,siren_amenageur,CODE_INSEE_COMMUNE,  
LIBELLE_REGION, CODE_REGION, LIBELLE_DEPARTEMENT, DEPARTMENT);
```

*/\* Ajout des contraintes de clés primaires \*/*

```
ALTER TABLE STATIONS  
ADD CONSTRAINT PK_STATIONS PRIMARY KEY (ID_STATIONS);
```

```
ALTER TABLE COMMUNE  
ADD CONSTRAINT PK_COMMUNE PRIMARY KEY (ID_COMMUNE);
```

```
ALTER TABLE DEPARTMENT  
ADD CONSTRAINT PK_DEPARTMENT PRIMARY KEY (ID_DEPARTMENT);
```

```
ALTER TABLE OPERATEUR  
ADD CONSTRAINT PK_OPERATEUR PRIMARY KEY (ID_OPERATEUR);
```

```
ALTER TABLE AMENAGEUR  
ADD CONSTRAINT PK_AMENAGEUR PRIMARY KEY (ID_AMENAGEUR);
```

```
/* Ajout des contraintes de clés étrangères */
```

```
ALTER TABLE COMMUNE ADD CONSTRAINT FK_COMMUNE FOREIGN KEY  
(ID_DEPARTMENT) REFERENCES DEPARTMENT (ID_DEPARTMENT);
```

```
ALTER TABLE STATIONS  
ADD CONSTRAINT FK_STATIONS_CO FOREIGN KEY (ID_COMMUNE) REFERENCES  
COMMUNE(ID_COMMUNE);
```

```
ALTER TABLE STATIONS  
ADD CONSTRAINT FK_STATIONS_AM FOREIGN KEY (ID_AMENAGEUR) REFERENCES  
AMENAGEUR(ID_AMENAGEUR);  
ALTER TABLE STATIONS
```

```
ADD CONSTRAINT FK_STATIONS_OP FOREIGN KEY (ID_OPERATEUR) REFERENCES  
OPERATEUR(ID_OPERATEUR);
```

## V - Répartition des tables par département

Les tables *COMMUNE* et *STATIONS* vont être divisées en sous-tables en fonction de leur département et chaque membre du groupe aura la responsabilité d'un ou plusieurs départements. Les autres tables n'ont pas été divisées car elles ne correspondent pas à un département particulier (exemple: *AMENAGEUR*).

### Répartition des départements :

- Thomas (S2A06A) : *STATIONSML*, *COMMUNEML*
- Pol (S2A04B) : *STATIONSLSA*, *STATIONSLSMA*, *COMMUNELSA*, *COMMUNELMA*
- Kyllian (S2A06B) : *STATIONSVE*, *STATIONSLSA*, *COMMUNEVE*, *COMMUNESA*

```
/* Création des tables pour la LOIRE-ATLANTIQUE */
```

```
CREATE TABLE COMMUNELSA as (SELECT S2A06A.COMMUNE.ID_COMMUNE,  
S2A06A.COMMUNE.CODE_INSEE_COMMUNE,  
S2A06A.COMMUNE.LIBELLE_COMMUNE,  
S2A06A.COMMUNE.ID_DEPARTMENT  
FROM S2A06A.COMMUNE, S2A06A.DEPARTMENT where  
S2A06A.COMMUNE.id_DEPARTMENT =  
S2A06A.DEPARTMENT.ID_DEPARTMENT AND  
S2A06A.DEPARTMENT.LIBELLE_DEPARTMENT =  
'Loire-Atlantique');
```

```
CREATE TABLE STATIONSLSA as (SELECT * from S2A06A.stations where ID_COMMUNE in  
(select ID_COMMUNE from COMMUNELSA));
```

```
/* Création des tables pour la VENDÉE */
```

```
CREATE TABLE COMMUNEVE as select * from S2A06A.COMMUNE where  
CODE_INSEE_COMMUNE like ('85%');
```

```
create table STATIONSVE as select * from S2A06A.STATIONS where ID_COMMUNE in (  
select ID_COMMUNE  
from COMMUNEVE  
);
```



```

/* Création des tables pour le MAINE-ET-LOIRE */
CREATE TABLE COMMUNEML as select * from COMMUNE where CODE_INSEE_COMMUNE
like ('49%');

create table STATIONSML as select * from STATIONS where ID_COMMUNE in (
    select ID_COMMUNE
    from COMMUNEML
);

/* Création des tables pour la SARTHE */
CREATE TABLE COMMUNESA as select * from S2A06A.COMMUNE where
CODE_INSEE_COMMUNE like ('72%');

create table STATIONSSA as select * from S2A06A.STATIONS where ID_COMMUNE in (
    select ID_COMMUNE
    from COMMUNESA
);

/* Création des tables pour la MAYENNE */
CREATE TABLE COMMUNEMA as (SELECT S2A06A.COMMUNE.ID_COMMUNE,
    S2A06A.COMMUNE.CODE_INSEE_COMMUNE,
    S2A06A.COMMUNE.LIBELLE_COMMUNE,
    S2A06A.COMMUNE.ID_DEPARTMENT
FROM S2A06A.COMMUNE, S2A06A.DEPARTMENT where
    S2A06A.COMMUNE.id_DEPARTMENT =
S2A06A.DEPARTMENT.ID_DEPARTMENT AND
    S2A06A.DEPARTMENT.LIBELLE_DEPARTMENT
= 'Mayenne');

CREATE TABLE STATIONSMA as (SELECT * from S2A06A.stations where ID_COMMUNE in
(select ID_COMMUNE from COMMUNEMA));

```

Les contraintes d'intégrités locales, qui correspondent aux tables créées, vont être ajoutées.

```

/* Clé primaire et étrangère sur les STATIONS de Loire-Atlantique */
ALTER TABLE STATIONSLA
ADD CONSTRAINT PK_STATIONS PRIMARY KEY (ID_STATIONS);

ALTER TABLE STATIONSLA
ADD CONSTRAINT FK_STATIONS_CO FOREIGN KEY (ID_COMMUNE) REFERENCES
COMMUNELA(ID_COMMUNE);

/* Clé primaire et étrangère sur les STATIONS de Vendée */
ALTER TABLE STATIONSVE
ADD CONSTRAINT PK_STATIONS PRIMARY KEY (ID_STATIONS);

ALTER TABLE STATIONSVE
ADD CONSTRAINT FK_STATIONS_CO FOREIGN KEY (ID_COMMUNE) REFERENCES
COMMUNELA(ID_COMMUNE);

```

*/\* Clé primaire et étrangère sur les STATIONS de Sarthe \*/*

```
ALTER TABLE STATIONSSA  
ADD CONSTRAINT PK_STATIONS PRIMARY KEY (ID_STATIONS);
```

```
ALTER TABLE STATIONSSA  
ADD CONSTRAINT FK_STATIONS_CO_SA FOREIGN KEY (ID_COMMUNE) REFERENCES  
COMMUNESA(ID_COMMUNE);
```

*/\* Clé primaire et étrangère sur les STATIONS de Mayenne \*/*

```
ALTER TABLE STATIONSMA  
ADD CONSTRAINT PK_STATIONS_MA PRIMARY KEY (ID_STATIONS);
```

```
ALTER TABLE STATIONSMA  
ADD CONSTRAINT FK_STATIONS_CO_MA FOREIGN KEY (ID_COMMUNE) REFERENCES  
COMMUNEMA(ID_COMMUNE);
```

*/\* Clé primaire et étrangère sur les STATIONS de MAINE ET LOIRE \*/*

```
ALTER TABLE STATIONSML  
ADD CONSTRAINT PK_STATIONSML PRIMARY KEY (ID_STATIONS);
```

```
ALTER TABLE STATIONSML  
ADD CONSTRAINT FK_STATIONSML_CO FOREIGN KEY (ID_COMMUNE) REFERENCES  
COMMUNEML(ID_COMMUNE);
```

Ajout des contraintes distantes qui font références à des clés dans les tables originales :  
*DÉPARTEMENT, AMÉNAGEUR, OPÉRATEUR.*

*/\* Droit donné par S2A06A aux autres membres du groupe pour pouvoir créer les clés étrangères \*/*

```
grant references on OPERATEUR to S2A04B;  
grant references on OPERATEUR to S2A06B;  
grant references on AMENAGEUR to S2A04B;  
grant references on AMENAGEUR to S2A06B;  
grant references on DEPARTMENT to S2A04B;  
grant references on DEPARTMENT to S2A06B;
```

*/\* Ajout des clés étrangères pour les tables de VENDEE \*/*

```
alter table COMMUNEVE add primary key (ID_COMMUNE);  
ALTER TABLE COMMUNEVE ADD CONSTRAINT FK_COMMUNE FOREIGN KEY  
(ID_DEPARTMENT) REFERENCES DEPARTMENT(ID_DEPARTMENT);
```

```
ALTER TABLE STATIONSVE  
ADD CONSTRAINT FK_STATIONS_AM FOREIGN KEY (ID_AMENAGEUR) REFERENCES  
S2A06A.AMENAGEUR(ID_AMENAGEUR);
```

```
ALTER TABLE STATIONSVE  
ADD CONSTRAINT FK_STATIONS_OP FOREIGN KEY (ID_OPERATEUR) REFERENCES  
S2A06A.OPERATEUR(ID_OPERATEUR);
```

*/\* Ajout des clés étrangères pour les tables de MAYENNE \*/*

```
alter table COMMUNEMA add primary key (ID_COMMUNE);  
ALTER TABLE COMMUNEMA ADD CONSTRAINT FK_COMMUNE_MA FOREIGN KEY  
(ID_DEPARTMENT) REFERENCES S2A06A.DEPARTMENT(ID_DEPARTMENT);
```

```
ALTER TABLE STATIONSMA  
ADD CONSTRAINT FK_STATIONS__AM_MA FOREIGN KEY (ID_AMENAGEUR)  
REFERENCES S2A06A.AMENAGEUR(ID_AMENAGEUR);
```

```
ALTER TABLE STATIONSMA  
ADD CONSTRAINT FK_STATIONS_OP_MA FOREIGN KEY (ID_OPERATEUR) REFERENCES  
S2A06A.OPERATEUR(ID_OPERATEUR);
```

*/\* Ajout des clés étrangères pour les tables de SARTHE \*/*

```
alter table COMMUNESA add primary key (ID_COMMUNE);  
ALTER TABLE COMMUNESA ADD CONSTRAINT FK_COMMUNE FOREIGN KEY  
(ID_DEPARTMENT) REFERENCES S2A06A.DEPARTMENT(ID_DEPARTMENT);
```

```
ALTER TABLE STATIONSSA  
ADD CONSTRAINT FK_STATIONS_AM_SA FOREIGN KEY (ID_AMENAGEUR) REFERENCES  
S2A06A.AMENAGEUR(ID_AMENAGEUR);
```

```
ALTER TABLE STATIONSSA  
ADD CONSTRAINT FK_STATIONS_OP_SA FOREIGN KEY (ID_OPERATEUR) REFERENCES  
S2A06A.OPERATEUR(ID_OPERATEUR);
```

*/\* Ajout des clés étrangères pour les tables de LOIRE ATLANTIQUE \*/*

```
alter table COMMUNELA add primary key (ID_COMMUNE);  
ALTER TABLE COMMUNELA ADD CONSTRAINT FK_COMMUNE FOREIGN KEY  
(ID_DEPARTMENT) REFERENCES S2A06A.DEPARTMENT(ID_DEPARTMENT);
```

```
ALTER TABLE STATIONSLA  
ADD CONSTRAINT FK_STATIONS_AM FOREIGN KEY (ID_AMENAGEUR) REFERENCES  
AMENAGEUR(ID_AMENAGEUR);
```

```
ALTER TABLE STATIONSLA  
ADD CONSTRAINT FK_STATIONS_OP FOREIGN KEY (ID_OPERATEUR) REFERENCES  
OPERATEUR(ID_OPERATEUR);
```

*/\* Ajout des clés étrangères pour les tables de MAINE ET LOIRE \*/*

```
alter table COMMUNEML add primary key (ID_COMMUNE);  
ALTER TABLE COMMUNEML ADD CONSTRAINT FK_COMMUNEML FOREIGN KEY  
(ID_DEPARTMENT) REFERENCES DEPARTMENT(ID_DEPARTMENT);
```

```
ALTER TABLE STATIONSML  
ADD CONSTRAINT FK_STATIONSML_AM FOREIGN KEY (ID_AMENAGEUR) REFERENCES  
AMENAGEUR(ID_AMENAGEUR);
```

```
ALTER TABLE STATIONSML  
ADD CONSTRAINT FK_STATIONSML_OP FOREIGN KEY (ID_OPERATEUR) REFERENCES  
OPERATEUR(ID_OPERATEUR);
```

## VI - Requêtes

Voici les requêtes effectuées :

- Affichez pour chaque Aménageur le nombre de prises par département.
- Afficher pour chaque Aménageur le nombre de prises avec la puissance nominale
- Affichez pour chaque département la répartition des prises en fonction des valeurs de la puissance nominale(22, 50, 43, 60, 120,...) seulement les 15 meilleures prises qui sont installées.
- Le nombre de stations par opérateurs.
- Le nombre de stations en fonction de leur implémentation.

**/\* Pour chaque aménageur le nombre de prises par département. Il y a donc une requête par département \*/**

```
-- ML
SELECT A.ID_AMENAGEUR,
       A.NOM_AMENAGEUR,
       SUM(NOMBRE_POINTS_CHARGE)
FROM S2A06A.AMENAGEUR A,
     S2A06A.STATIONSML S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;

-- LA
SELECT A.ID_AMENAGEUR,
       A.NOM_AMENAGEUR,
       SUM(NOMBRE_POINTS_CHARGE)
FROM S2A06A.AMENAGEUR A,
     S2A04B.STATIONSMA S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;

-- MA
SELECT A.ID_AMENAGEUR,
       A.NOM_AMENAGEUR,
       SUM(NOMBRE_POINTS_CHARGE)
FROM S2A06A.AMENAGEUR A,
     S2A04B.STATIONSMA S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;

-- VE
SELECT A.ID_AMENAGEUR,
       A.NOM_AMENAGEUR,
       SUM(NOMBRE_POINTS_CHARGE)
FROM S2A06A.AMENAGEUR A,
     S2A06B.STATIONSVE S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;

-- SA
SELECT A.ID_AMENAGEUR,
       A.NOM_AMENAGEUR,
       SUM(NOMBRE_POINTS_CHARGE)
FROM S2A06A.AMENAGEUR A,
     S2A06B.STATIONSSA S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;
```

*/\* Afficher pour chaque Aménageur le nombre de prises avec la puissance nominale \*/*

```
SELECT A.ID_AMENAGEUR,  
       A.NOM_AMENAGEUR,  
       S.PUISSANCE_NOMINALE,  
       SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A06A.STATIONS S  
INNER JOIN S2A06A.AMENAGEUR A ON A.ID_AMENAGEUR = S.ID_AMENAGEUR  
GROUP BY S.PUISSANCE_NOMINALE, A.NOM_AMENAGEUR, A.ID_AMENAGEUR;
```

*/\* Affichez pour chaque département la répartition des prises en fonction des valeurs de la puissance nominale(22, 50, 43, 60, 120,...) seulement les 15 meilleures prises qui sont installées. Il y a donc une requête par département. \*/*

-- ML

```
SELECT PUISSANCE_NOMINALE,  
       SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A06A.STATIONSML  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

-- LA

```
SELECT PUISSANCE_NOMINALE,  
       SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A04B.STATIONSLA  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

-- MA

```
SELECT PUISSANCE_NOMINALE,  
       SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A04B.STATIONSMA  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

-- VE

```
SELECT PUISSANCE_NOMINALE,  
       SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A06B.STATIONSVE  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

-- SA

```
SELECT PUISSANCE_NOMINALE,  
       SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A06B.STATIONSSA  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

*/\* Le nombre de stations par opérateur \*/*

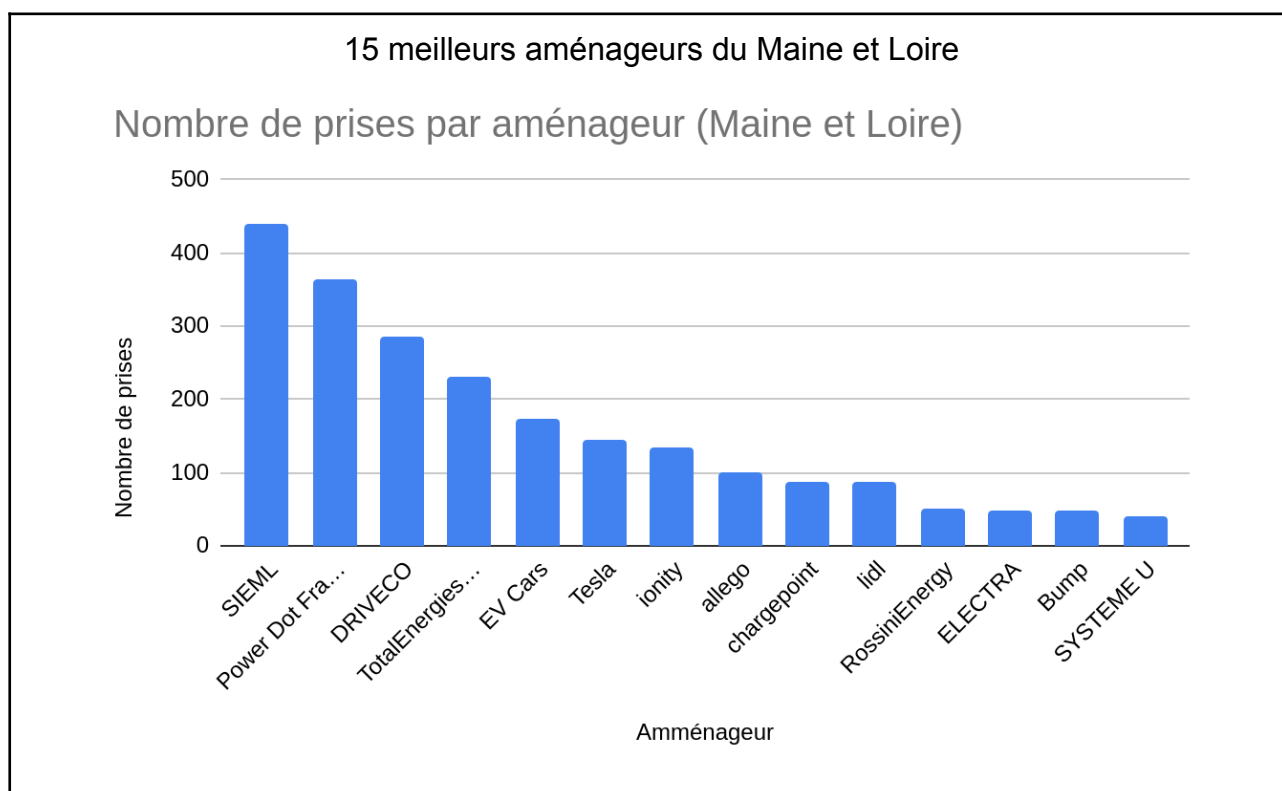
```
SELECT  
       NOM_OPERATEUR,  
       COUNT(*) AS NOMBRE_STATIONS  
FROM OPERATEUR O, STATIONS S  
WHERE O.id_OPERATEUR = S.ID_OPERATEUR  
GROUP BY NOM_OPERATEUR;
```

**/\* Le nombre de stations par implémentation \*/**

```
SELECT  
  IMPLANTATION_STATION,  
  COUNT(*) AS NOMBRE_STATIONS  
FROM STATIONS  
GROUP BY IMPLANTATION_STATION;
```

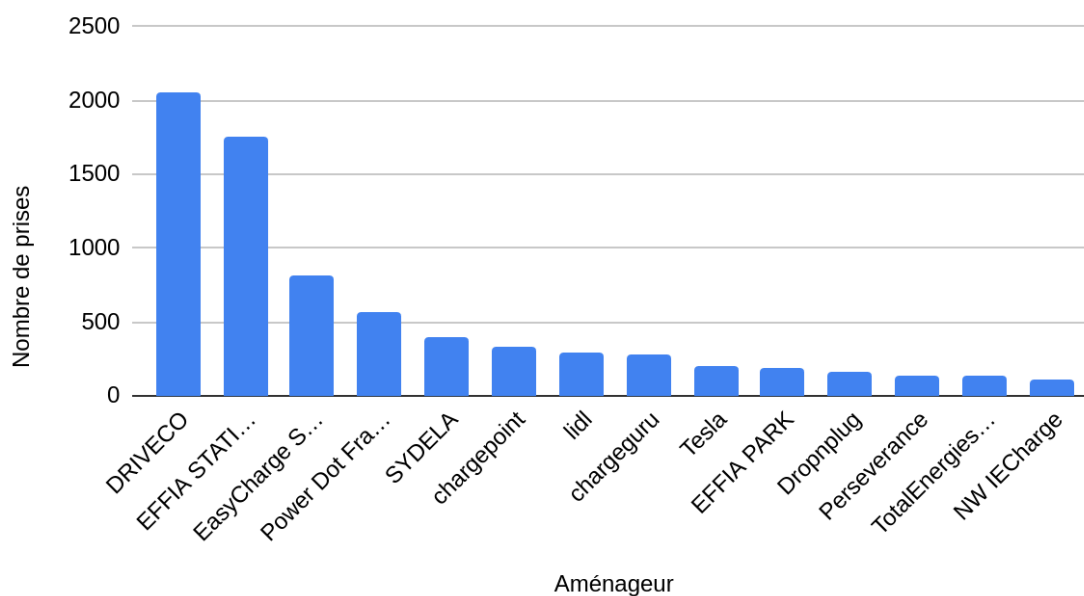
Voici les résultats des requêtes effectuées :

**Résultats pour la requête “Pour chaque aménageur le nombre de prises par département”**



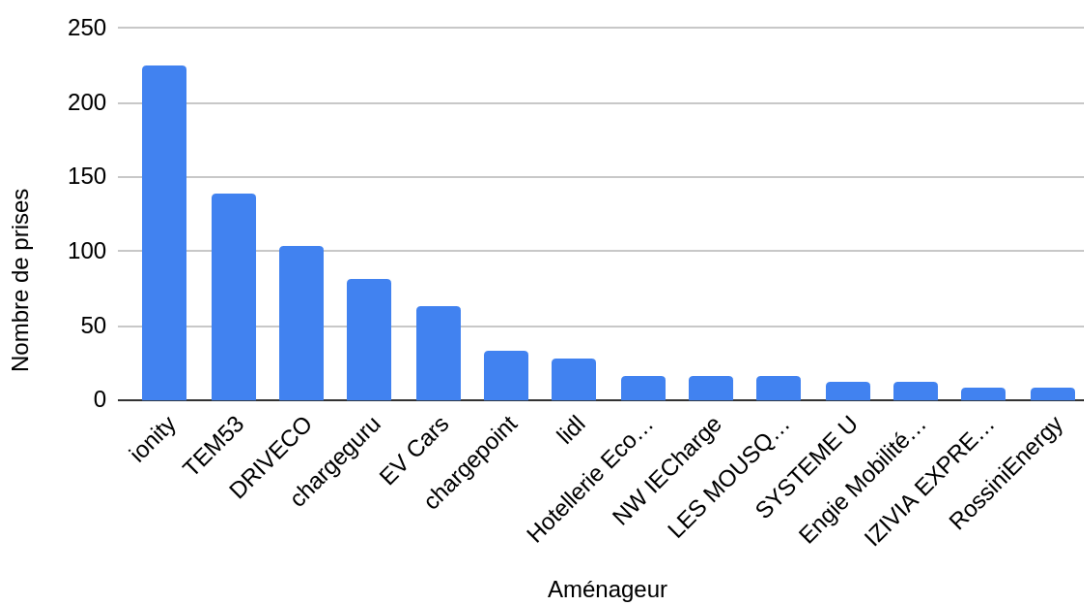
### 15 meilleurs aménageurs de la Loire Atlantique

#### Nombre de prises par aménageur (Loire Atlantique)



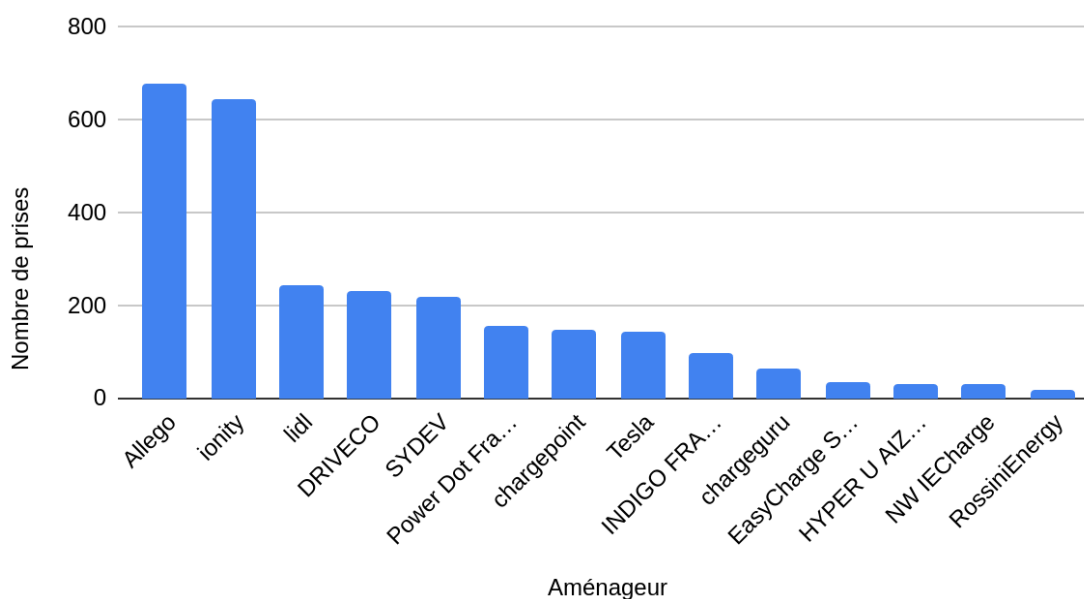
### 15 meilleurs aménageurs de la Mayenne

#### Nombre de prises par aménageur (Mayenne)



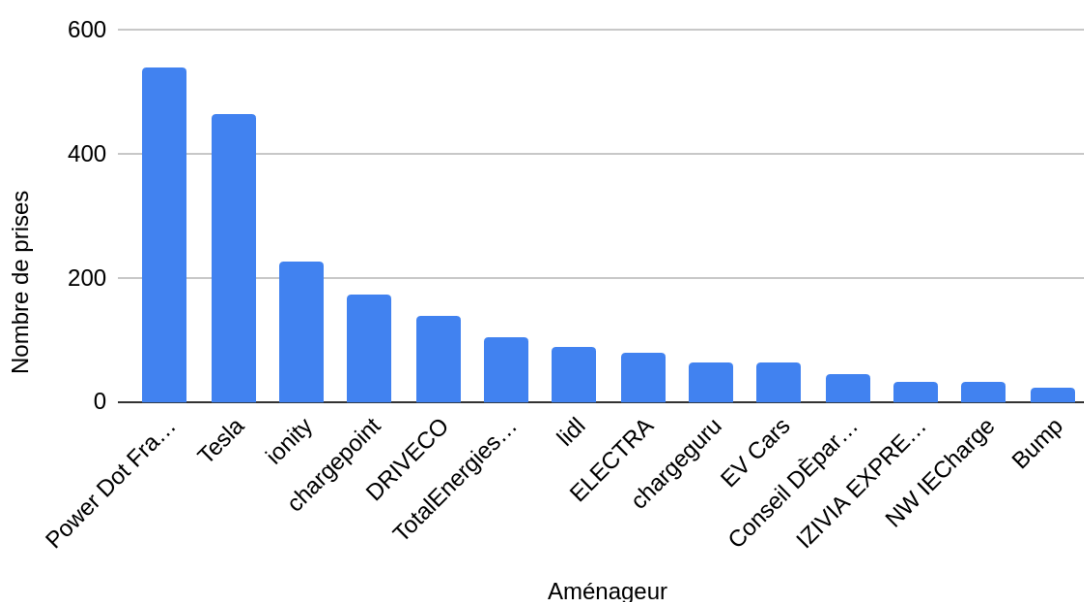
### 15 meilleurs aménageurs de la Vendée

#### Nombre de prises par aménageur (Vendée)



### 15 meilleurs aménageurs de la Sarthe

#### Nombre de prises par aménageur (Sarthe)



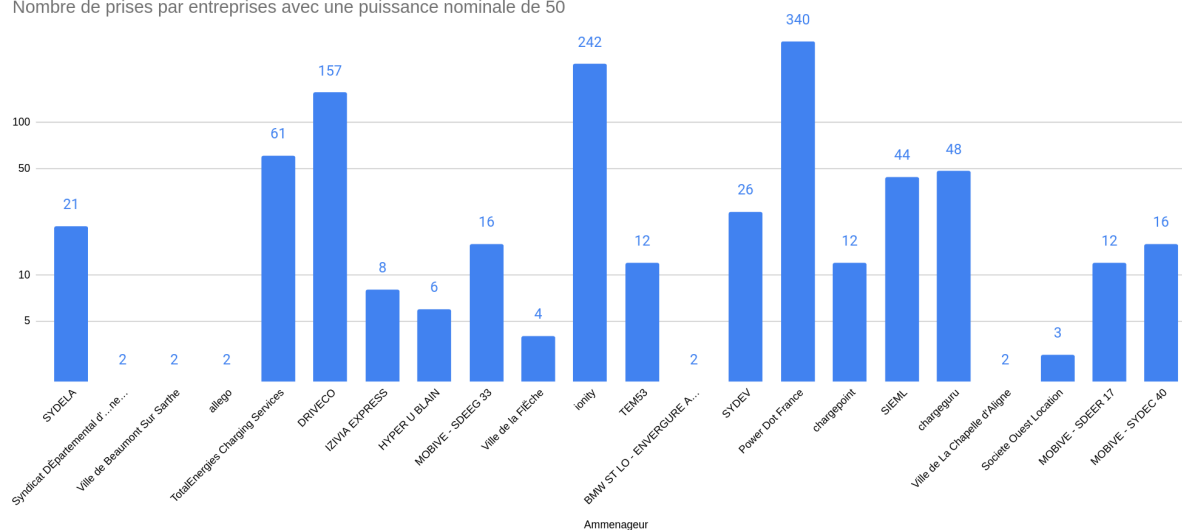
Sur ces graphiques qui représentent le nombre de prises par aménageur pour chaque département, on peut d'abord constater que l'on retrouve souvent les mêmes aménageurs dans les trois premiers avec Ionity, Driveco et Power Dot. On constate aussi que la Loire Atlantique compte beaucoup plus de prises que les autres départements.

**Résultats pour la requête "Pour chaque aménageur le nombre de prises avec la puissance nominale"**



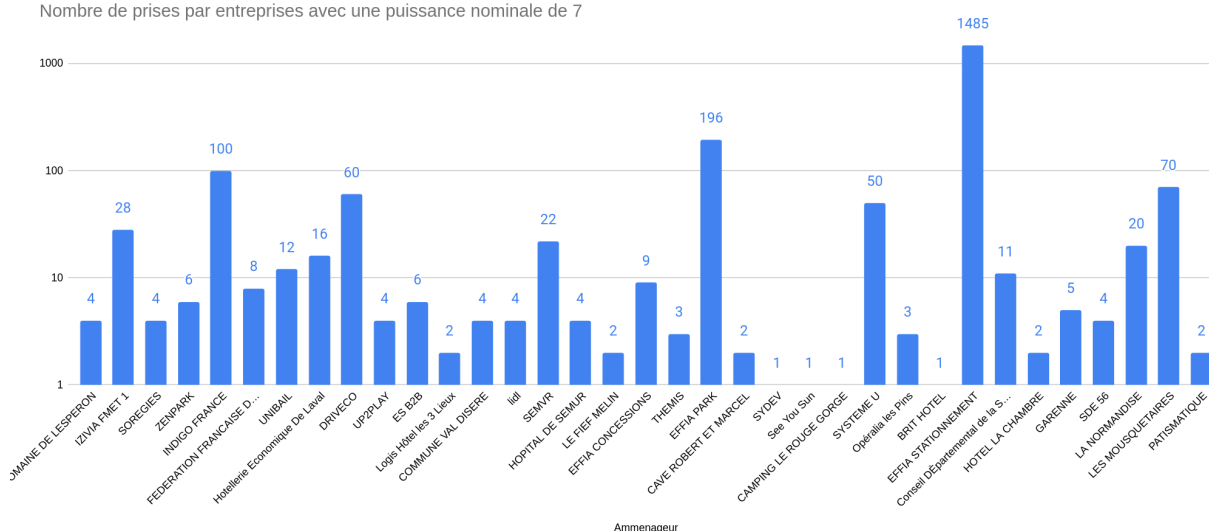
## Nombre de prises par aménageur avec une puissance nominale de 50 kW

Nombre de prises par entreprises avec une puissance nominale de 50



## Nombre de prises par aménageur avec une puissance nominale de 7 kW

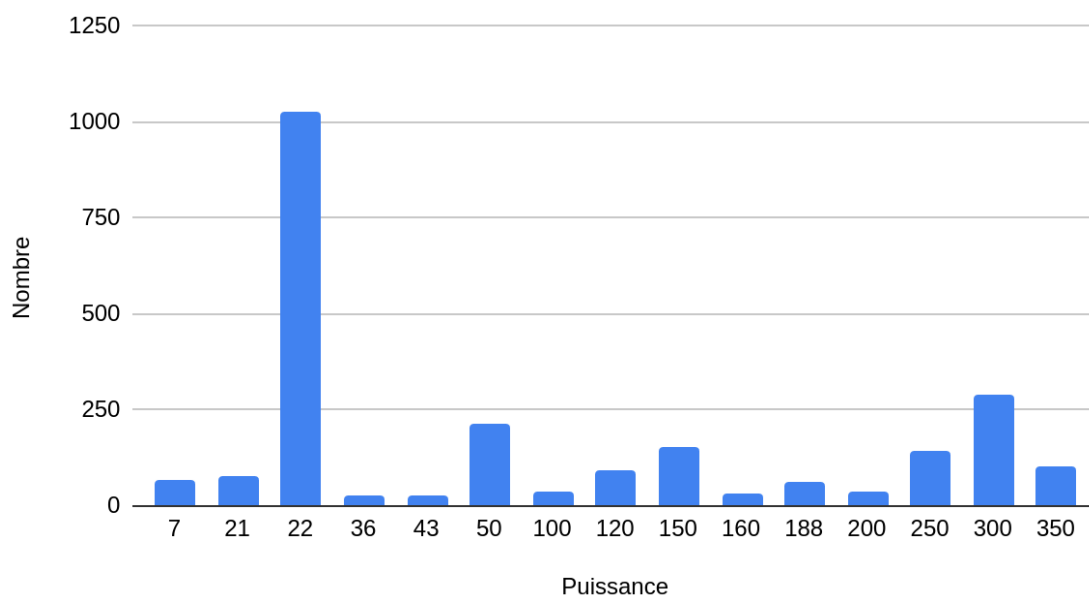
Nombre de prises par entreprises avec une puissance nominale de 7



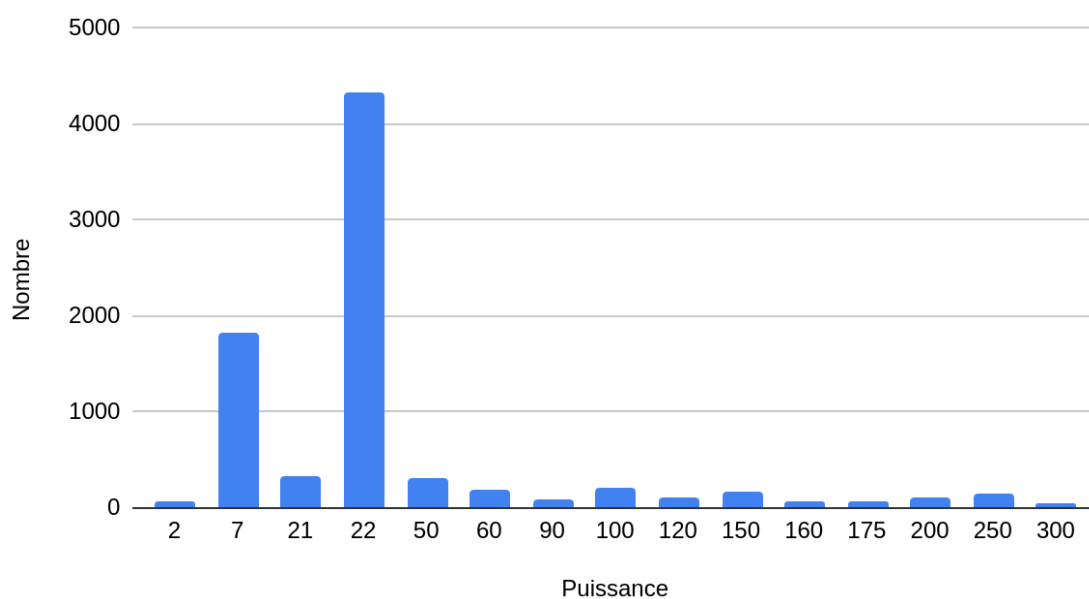
Les résultats présentés ici ne sont que sur les puissances nominales de 7 kW et 50 kW. On constate que pour les prises avec une puissance de 50 kW c'est Power Dot France qui domine le marché dans la région avec 340 prises mais il est suivi de très près par Ionity avec 242 prises. Pour les prises avec une puissance de 7 kW c'est EFFIA Stationnement avec 1485 prises.

**Résultats pour la requête "Pour chaque département la répartition des prises en fonction de leur puissance"**

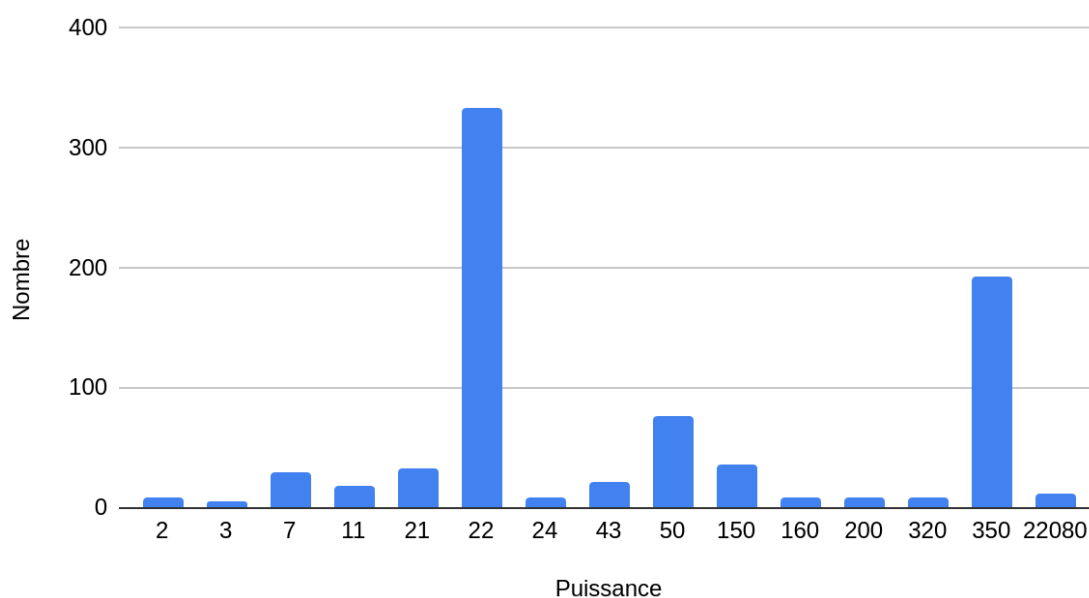
### Nombre de prises par rapport à la puissance (Maine et Loire)



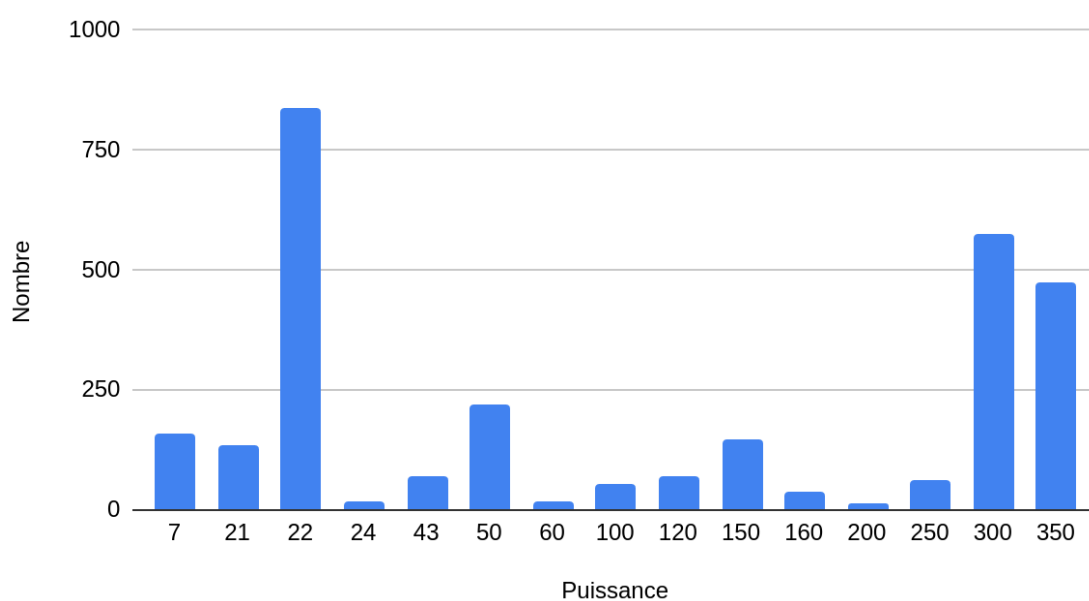
### Nombre de prises par rapport à la puissance (Loire Atlantique)



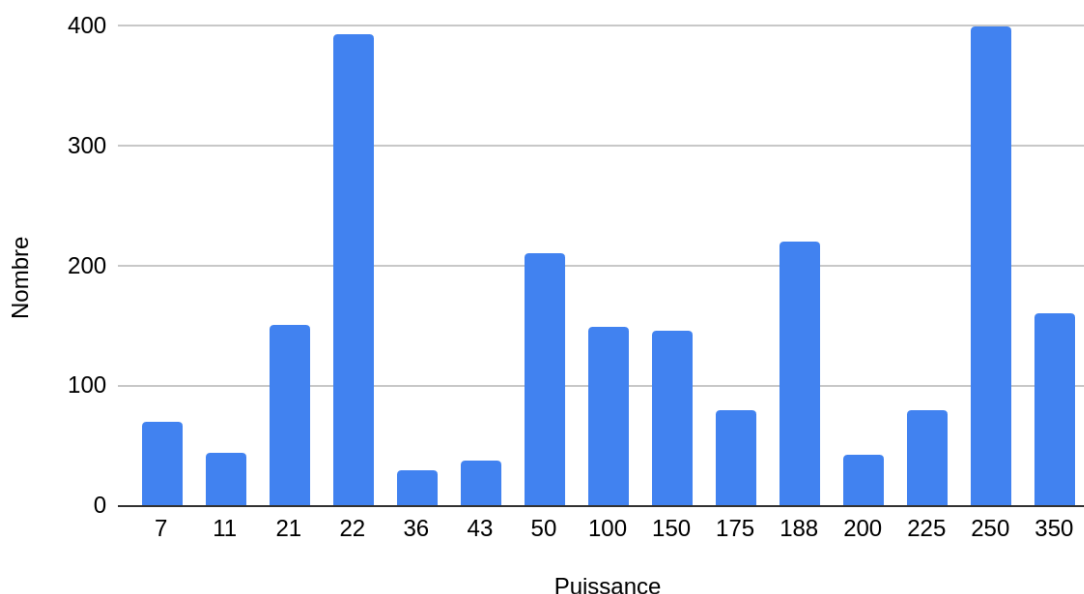
Nombre de prises par rapport à la puissance (Mayenne)



Nombre de prises par rapport à la puissance (Vendée)

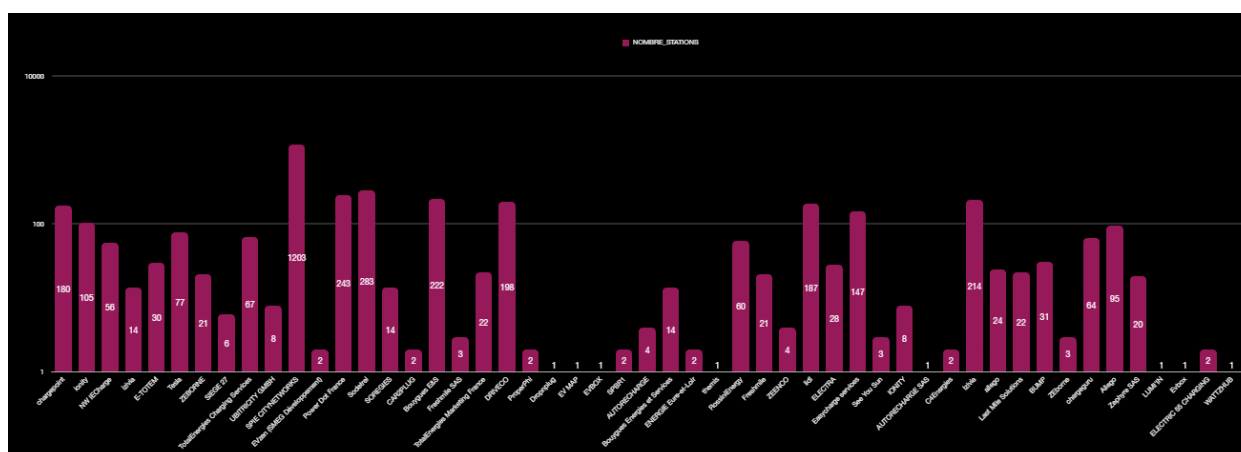


### Nombre de prises par rapport à la puissance (Sarthe)



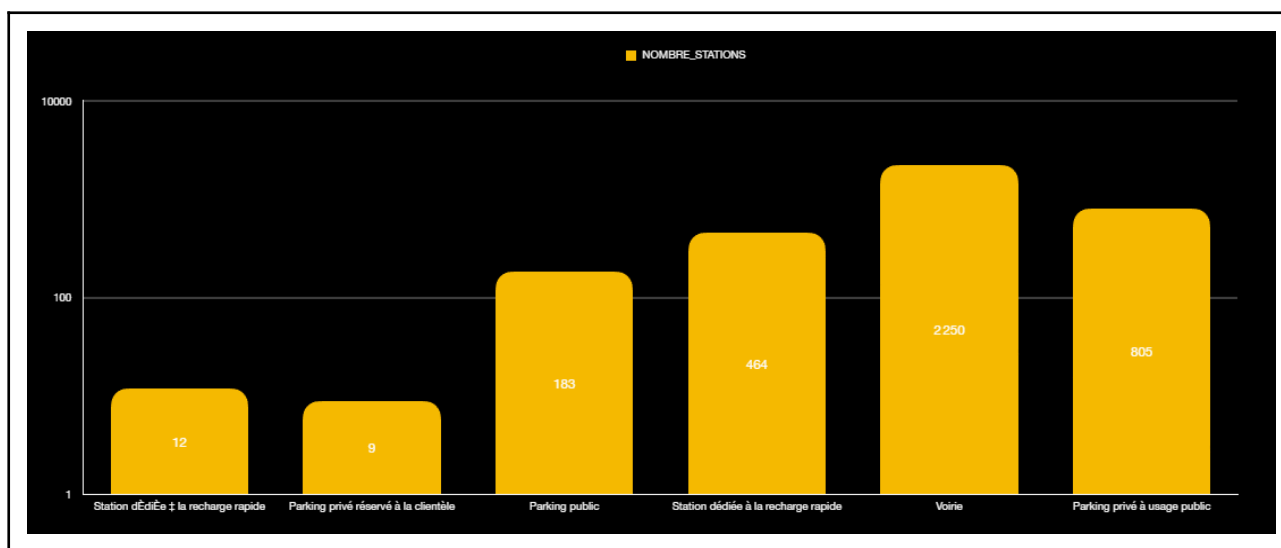
On constate tout d'abord que la puissance de 22 kW est celle qui est la plus répandue dans presque tous les départements. Cependant, il existe certaines variations en fonction des départements. La Sarthe par exemple compte beaucoup plus de prises avec des puissances élevées comme 250 kW ce qui correspond généralement à des bornes de recharge rapides, cela peut être expliqué par le fait que ce département est traversé par de nombreuses autoroutes.

### Résultats pour la requête “Nombre de stations par opérateur”



On constate ici que l'opérateur majoritaire dans la région est SPIE City Networks avec 1203 stations.

## Résultats pour la requête “Nombre de stations par implémentation”



Les résultats nous montrent ici que les stations sont majoritairement implantées sur la voirie, puis sur les parkings publics ou sur les stations de recharge rapide.

## VII - Vues

Les résultats des requêtes précédentes sont d’abord stockés dans des vues spécifiques à chaque département par chaque membre du groupe. Puis un seul membre du groupe (Pol - S2A04B) se charge de rassembler les résultats dans des vues globales.

**/\* Les droits suivants sont nécessaires pour permettre aux membres du groupe de créer des vues il sont données par le compte qui détient les tables originales : S2A06A \*/**

grant select on AMENAGEUR to S2A04B with grant option;

grant select on AMENAGEUR to S2A06B with grant option;

**/\* Chaque membre donne aussi les ces droits une fois qu’il à créer ses vues \*/**

grant select on PRISES\_AMENAGEUR\_ML to S2A04B with grant option;

grant select on PRISES\_AMENAGEUR\_SA to S2A04B with grant option;

grant select on PRISES\_AMENAGEUR\_VE to S2A04B with grant option;

*/\* Vues pour le Maine et Loire pour la première et la troisième requête \*/*

*/\* Requête 1 \*/*

```
UPDATE S2A06A.STATIONSML  
SET NOMBRE_POINTS_CHARGE = TO_NUMBER(NOMBRE_POINTS_CHARGE,  
'999.9');
```

```
CREATE VIEW PRISES_AMENAGEUR_ML AS SELECT A.ID_AMENAGEUR,  
      A.NOM_AMENAGEUR,  
      SUM(NOMBRE_POINTS_CHARGE) AS NB_PRISES  
FROM S2A06A.AMENAGEUR A,  
      S2A06A.STATIONSML S  
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR  
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;
```

*/\* Requête 3 \*/*

```
CREATE VIEW REPARTITIONML AS  
SELECT PUISSANCE_NOMINALE,  
      COUNT(*) AS NOMBRE_PRISES  
FROM S2A06A.STATIONSML  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

*/\* Vues pour la Loire Atlantique pour la première et la troisième requête \*/*

*/\* Requête 1 \*/*

```
UPDATE S2A04B.STATIONSMLA  
SET NOMBRE_POINTS_CHARGE = TO_NUMBER(NOMBRE_POINTS_CHARGE,  
'999.9');
```

```
CREATE VIEW PRISES_AMENAGEUR_LA AS SELECT A.ID_AMENAGEUR,  
      A.NOM_AMENAGEUR,  
      SUM(NOMBRE_POINTS_CHARGE) AS NB_PRISES  
FROM S2A06A.AMENAGEUR A,  
      S2A04B.STATIONSMLA S  
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR  
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;
```

*/\* Requête 3 \*/*

```
CREATE VIEW REPARTITIONLA AS  
SELECT PUISSANCE_NOMINALE,  
      COUNT(*) AS NOMBRE_PRISES  
FROM S2A04B.STATIONSMLA  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

/\* Vues pour la Mayenne pour la première et la troisième requête \*/

/\* Requête 1 \*/

```
UPDATE S2A04B.STATIONSMA
SET NOMBRE_POINTS_CHARGE = TO_NUMBER(NOMBRE_POINTS_CHARGE,
'999.9');
```

```
CREATE VIEW PRISES_AMENAGEUR_MA AS SELECT A.ID_AMENAGEUR,
      A.NOM_AMENAGEUR,
      SUM(NOMBRE_POINTS_CHARGE) AS NB_PRISES
FROM S2A06A.AMENAGEUR A,
      S2A04B.STATIONSMA S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;
```

/\* Requête 3 \*/

```
CREATE VIEW REPARTITIONMA AS
SELECT PUISSANCE_NOMINALE,
      COUNT(*) AS NOMBRE_PRISES
FROM S2A04B.STATIONSMA
GROUP BY PUISSANCE_NOMINALE
ORDER BY NOMBRE_PRISES DESC
FETCH FIRST 15 ROWS ONLY;
```

/\* Vues pour la Vendée pour la première et la troisième requête \*/

/\* Requête 1 \*/

```
UPDATE S2A06B.STATIONSVE
SET NOMBRE_POINTS_CHARGE = TO_NUMBER(NOMBRE_POINTS_CHARGE,
'999.9');
```

```
CREATE VIEW PRISES_AMENAGEUR_VE AS SELECT A.ID_AMENAGEUR,
      A.NOM_AMENAGEUR,
      SUM(NOMBRE_POINTS_CHARGE) AS NB_PRISES
FROM S2A06A.AMENAGEUR A,
      S2A06B.STATIONSVE S
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;
```

/\* Requête 3 \*/

```
CREATE VIEW REPARTITIONVE AS
SELECT PUISSANCE_NOMINALE,
      COUNT(*) AS NOMBRE_PRISES
FROM S2A06B.STATIONSVE
GROUP BY PUISSANCE_NOMINALE
ORDER BY NOMBRE_PRISES DESC
FETCH FIRST 15 ROWS ONLY;
```

*/\* Vues pour la Sarthe pour la première et la troisième requête \*/*

*/\* Requête 1 \*/*

```
UPDATE S2A06B.STATIONSSA  
SET NOMBRE_POINTS_CHARGE = TO_NUMBER(NOMBRE_POINTS_CHARGE,  
'999.9');
```

```
CREATE VIEW PRISES_AMENAGEUR_SA AS SELECT A.ID_AMENAGEUR,  
      A.NOM_AMENAGEUR,  
      SUM(NOMBRE_POINTS_CHARGE) AS NB_PRISES  
FROM S2A06A.AMENAGEUR A,  
      S2A06B.STATIONSSA S  
WHERE A.ID_AMENAGEUR = S.ID_AMENAGEUR  
GROUP BY A.ID_AMENAGEUR, A.NOM_AMENAGEUR;
```

*/\* Requête 3 \*/*

```
CREATE VIEW REPARTITIONSA AS  
SELECT PUISSANCE_NOMINALE,  
      COUNT(*) AS NOMBRE_PRISES  
FROM S2A06B.STATIONSSA  
GROUP BY PUISSANCE_NOMINALE  
ORDER BY NOMBRE_PRISES DESC  
FETCH FIRST 15 ROWS ONLY;
```

*/\* Vue globale pour la requête 2 \*/*

```
create view amenageurpuissancenom as (  
SELECT A.ID_AMENAGEUR,  
      A.NOM_AMENAGEUR,  
      S.PUISSANCE_NOMINALE,  
      SUM(NOMBRE_POINTS_CHARGE) AS NOMBRE_PRISES  
FROM S2A06A.STATIONS S  
INNER JOIN S2A06A.AMENAGEUR A ON A.ID_AMENAGEUR = S.ID_AMENAGEUR  
GROUP BY S.PUISSANCE_NOMINALE, A.NOM_AMENAGEUR, A.ID_AMENAGEUR  
);
```

*/\* Vue globale pour la requête 4 \*/*

```
create view stationsparop as (  
  SELECT  
    NOM_OPERATEUR,  
    COUNT(*) AS NOMBRE_STATIONS  
FROM s2a06A.OPERATEUR O, s2a06A.STATIONS S  
WHERE O.id_OPERATEUR = S.ID_OPERATEUR  
GROUP BY NOM_OPERATEUR  
);
```

*/\* Vue globale pour la requête 5 \*/*

```
create view stationimplementation as (  
SELECT  
  IMPLANTATION_STATION,  
  COUNT(*) AS NOMBRE_STATIONS  
FROM STATIONS  
GROUP BY IMPLANTATION_STATION  
);
```



/\* Vue globale pour la requête 1, cette vue utilise les autres vues créées précédemment de chaque département. \*/

```
create view prises_amenageur as
select COALESCE(previous.id_amenageur,sa.ID_AMENAGEUR) as
ID_AMENAGEUR,COALESCE(previous.NOM_AMENAGEUR,sa.NOM_AMENAGEUR) as
NOM_AMENAGEUR,
      COALESCE(previous.nb_prises_LA, 0) as
NB_PRISES_LA,COALESCE(previous.nb_prises_ML, 0) as NB_PRISES_ML,
COALESCE(previous.NB_PRISES_MA,0) as NB_PRISES_MA,
      COALESCE(previous.NB_PRISES_ve,0) as NB_PRISES_VE,
COALESCE(sa.NB_PRISES,0) as NB_PRISES_SA
from (
select COALESCE(previous.id_amenageur,ve.ID_AMENAGEUR) as
ID_AMENAGEUR,COALESCE(previous.NOM_AMENAGEUR,ve.NOM_AMENAGEUR) as
NOM_AMENAGEUR,
      COALESCE(previous.nb_prises_LA, 0) as
NB_PRISES_LA,COALESCE(previous.nb_prises_ML, 0) as NB_PRISES_ML,
COALESCE(previous.NB_PRISES_MA,0) as NB_PRISES_MA,
      COALESCE(ve.NB_PRISES,0) as NB_PRISES_VE
from (
select COALESCE(previous.id_amenageur,ma.ID_AMENAGEUR) as
ID_AMENAGEUR,COALESCE(previous.NOM_AMENAGEUR,ma.NOM_AMENAGEUR) as
NOM_AMENAGEUR,
      COALESCE(previous.nb_prises_LA, 0) as
NB_PRISES_LA,COALESCE(previous.nb_prises_ML, 0) as NB_PRISES_ML,
COALESCE(ma.NB_PRISES,0) as NB_PRISES_MA
from
      (select COALESCE(ml.id_amenageur,la.ID_AMENAGEUR) as
ID_AMENAGEUR,COALESCE(ml.NOM_AMENAGEUR,la.NOM_AMENAGEUR) as
NOM_AMENAGEUR, COALESCE(ml.nb_prises, 0) as NB_PRISES_ML,
COALESCE(la.NB_PRISES,0) as NB_PRISES_LA
      from S2A06A.prises_amenageur_ml ml full outer join
S2A04B.PRISES_AMENAGEUR_LA la on la.ID_AMENAGEUR = ml.ID_AMENAGEUR)
previous
full outer join S2A04B.prises_amenageur_ma ma on previous.ID_AMENAGEUR =
ma.ID_AMENAGEUR) previous
full outer join S2A06B.prises_amenageur_ve ve on previous.ID_AMENAGEUR =
ve.ID_AMENAGEUR) previous
full outer join S2A06B.prises_amenageur_sa sa on previous.ID_AMENAGEUR =
sa.ID_AMENAGEUR
```

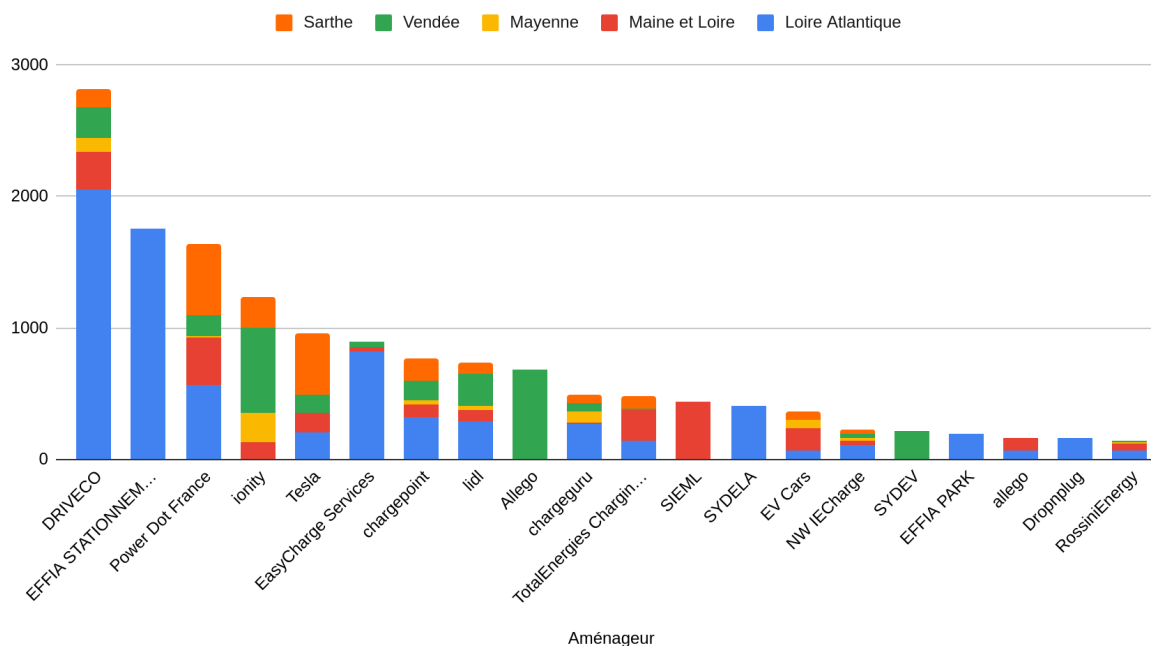
/\* Vue globale pour la requête 3, cette vue utilise les autres vues créées précédemment de chaque département. \*/

```
create view repartition as
select COALESCE(previous.puissance_nominale,sa.puissance_nominale) as
PUISSANCE_NOMINALE,
      COALESCE(previous.nombre_prises_la, 0) as
nombre_prises_la,COALESCE(previous.nombre_prises_ml, 0) as nombre_prises_ml,
COALESCE(previous.nombre_prises_ma,0) as nombre_prises_ma,
      COALESCE(previous.nombre_prises_ve,0) as nombre_prises_ve,
COALESCE(sa.nombre_prises,0) as nombre_prises_sa
from (
select COALESCE(previous.puissance_nominale,ve.puissance_nominale) as
PUISSANCE_NOMINALE,
      COALESCE(previous.nombre_prises_la, 0) as
nombre_prises_la,COALESCE(previous.nombre_prises_ml, 0) as nombre_prises_ml,
COALESCE(previous.nombre_prises_ma,0) as nombre_prises_ma,
      COALESCE(ve.nombre_prises,0) as nombre_prises_ve
from (
select COALESCE(previous.puissance_nominale,ma.puissance_nominale) as
PUISSANCE_NOMINALE,
      COALESCE(previous.nombre_prises_la, 0) as
nombre_prises_la,COALESCE(previous.nombre_prises_ml, 0) as nombre_prises_ml,
COALESCE(ma.nombre_prises,0) as nombre_prises_ma
      from
      (select COALESCE(ml.puissance_nominale,la.puissance_nominale) as
PUISSANCE_NOMINALE,
      COALESCE(ml.nombre_prises, 0) as nombre_prises_ml,
COALESCE(la.nombre_prises,0) as nombre_prises_la
      from S2A06A.repartitionml ml full outer join S2A04B.REPARTITIONLA la on
la.PUISSANCE_NOMINALE = ml.puissance_nominale) previous
full outer join S2A04B.repartitionma ma on previous.PUISSANCE_NOMINALE =
ma.PUISSANCE_NOMINALE) previous
full outer join S2A06B.repartitionve ve on previous.PUISSANCE_NOMINALE =
ve.PUISSANCE_NOMINALE) previous
full outer join S2A06B.repartitionsa sa on previous.PUISSANCE_NOMINALE =
sa.PUISSANCE_NOMINALE;
```

Voici donc les résultats obtenus avec les vues qui rassemblent les résultats des différents départements.

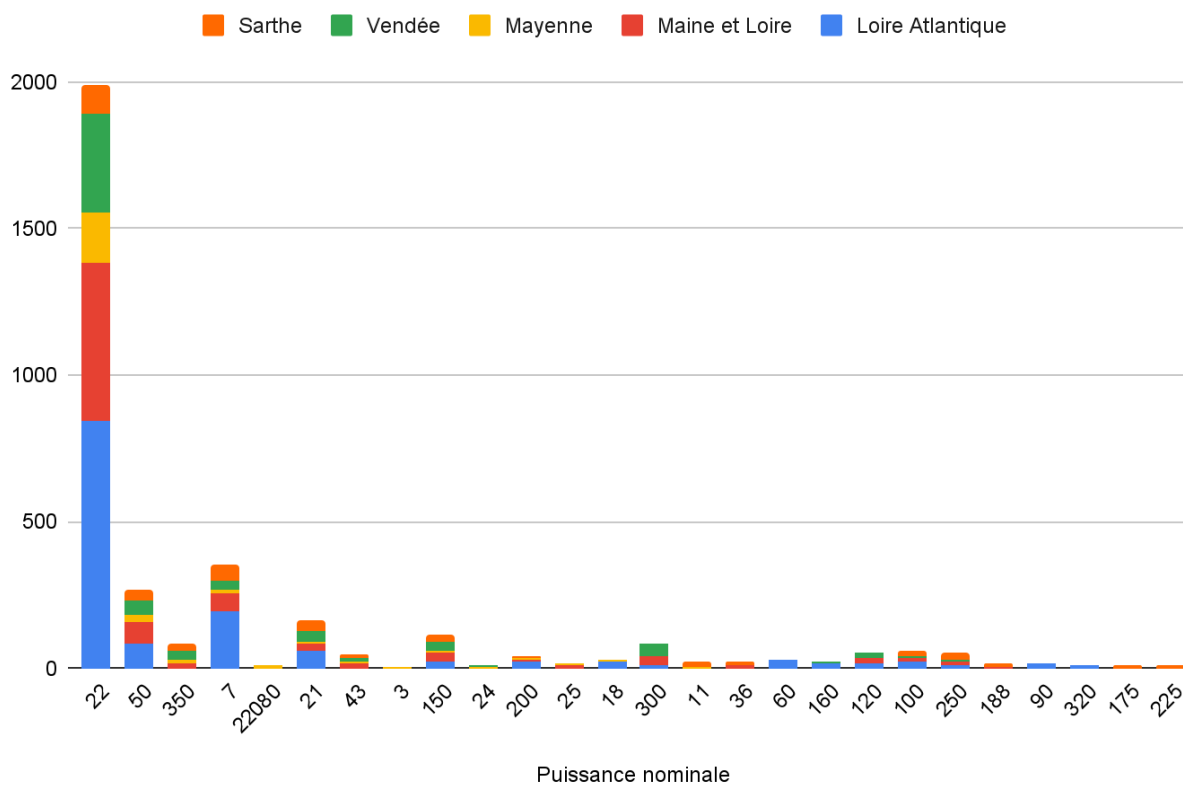
Graphique global pour chaque aménageur le nombre de prises par département.

Nombre de prises par aménageur et par département



Graphique global pour chaque département la répartition des prises.

Nombre de prises par puissance nominale et par département



Grâce à ces graphiques, on détermine que l'aménageur le plus présent en Pays de la Loire est DRIVECO, suivi de EFFIA et de Power Dot. Pour la répartition des prises, la très grande majorité ont une puissance de 22 kW, qui correspond aux bornes de recharge classique pour les véhicules électriques.

## VIII - Préparation d'un second jeu de données

Dans cette partie, nous avons préparé un second jeu de données. Il provient lui aussi de l'Open Data des Pays de la Loire et il regroupe l'ensemble des restaurants de la région. Les données ont donc d'abord été extraites puis traitées et enfin nous avons divisé les tables en plusieurs tables.

**Source des données :** [Offre touristique : restaurants en Pays de la Loire](#)

Colonnes sélectionnées :

- Nom de l'offre touristique
- Type de restaurant
- Catégorie du restaurant
- Adresse1
- Adresse2
- Code Postale
- Commune
- Code INSEE de la commune
- Numéro de téléphone Fixe
- Numéro de téléphone Mobile

**Traitement des données :**

```
/* Ajout des type par défaut pour ceux manquants */
update S2A06A.restaurants set typerestaurant = 'Restaurant' where typerestaurant is
null;

/* Regroupement des deux adresses en une colonne */
alter table s2A06A.restaurants add adresse varchar2(128);

update s2A06A.restaurants set adresse = adresse2;

update s2A06A.restaurants set adresse = adresse1 where adresse is null;

alter table s2a06A.restaurants drop column adresse1

alter table s2a06A.restaurants drop column adresse2

/* Ajout d'une catégorie par défaut pour celles manquantes */
update s2A06A.restaurants set CATÉGORIERESTAURANT = 'Cuisine traditionnelle'
where CATÉGORIERESTAURANT is null;

/* Regroupement des numéros de téléphones portables et fixes en une colonne */
alter table s2A06A.restaurants add telephone varchar2(128);

update s2A06A.restaurants set telephone = TÉLÉPHONEFIXE;
```

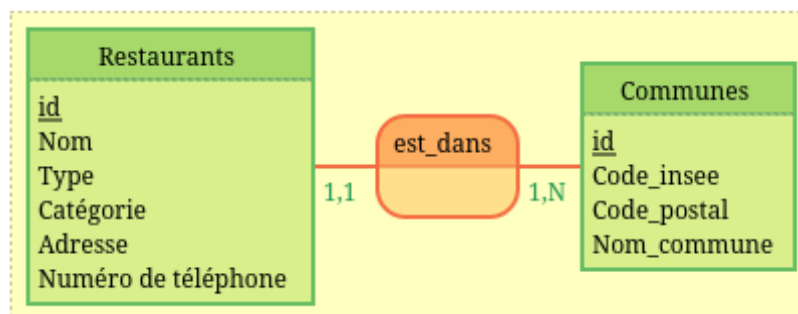
```
update s2A06A.restaurants set telephone = TÉLÉPHONEMOBILE where telephone is null;
```

```
alter table s2a06A.restaurants drop column TÉLÉPHONEMOBILE;
```

```
alter table s2a06A.restaurants drop column TÉLÉPHONEFIXE;
```

## Répartition des tables avec les contraintes :

Nous avons séparé la table principale en deux pour obtenir une table commune, la table restaurant contient donc une clé étrangère pour identifier la commune.



### */\* Création de la table COMMUNES et ajout de la clé primaire \*/*

```
CREATE TABLE COMMUNES (  
ID NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),  
CODE_INSEE NUMBER(38),  
NOM_COMMUNE VARCHAR2(128),  
CODÉ_POSTAL NUMBER(38)  
);
```

```
Insert into COMMUNES (CODE_INSEE, NOM_COMMUNE, CODE_POSTAL)  
select distinct CODEINSEE, COMMUNE, CODEPOSTAL  
from RESTAURANTS;
```

```
ALTER TABLE COMMUNES ADD PRIMARY KEY (ID);
```

### */\* Création de la table RESTAURANTS et ajout de la clé primaire et étrangère \*/*

```
ALTER TABLE RESTAURANTS ADD PRIMARY KEY (ID);
```

```
ALTER TABLE RESTAURANTS ADD ID_COMMUNE NUMBER;  
ALTER TABLE RESTAURANTS ADD ID NUMBER GENERATED ALWAYS as  
IDENTITY(START with 1 INCREMENT by 1);
```

```
UPDATE S2A06A.RESTAURANTS r SET r.ID_COMMUNE = (select MAX(ID) FROM  
S2A06A.COMMUNES WHERE r.CODEINSEE = S2A06A.COMMUNES.CODE_INSEE);
```

```
ALTER TABLE RESTAURANTS  
ADD CONSTRAINT FK_RESTAURANTS FOREIGN KEY (ID_COMMUNE) REFERENCES  
COMMUNES(ID);
```

```
ALTER TABLE RESTAURANTS DROP (CODEINSEE, CODEPOSTAL, COMMUNE)
```

## **IX - Conclusion**

En conclusion, ce projet à permis de se familiariser avec les étapes nécessaires à l'exploitation de données libres, avec l'importation, le traitement, la division des tables et enfin l'exploitation avec des requêtes. L'exploitation à aussi permis d'analyser les données des bornes de recharge de véhicules électriques en Pays de la Loire pour en tirer différentes informations comme la répartition des bornes en fonction de leur puissance ou alors les entreprises qui détiennent le plus de bornes. Finalement nous avons aussi préparé un autre jeu de données qui concerne les restaurants en Pays de la Loire.