

# IUT de NANTES

## SAE- BD :

### Bornes de recharge de véhicules électriques en Pays de la Loire

Dans ce projet de SAE vous allez travailler par groupe de 4 étudiants. **Faire un seul rapport de ce projet. Une soutenance sera programmée par groupe en fin de cette SAE.**

Dans ce projet, vous allez travailler sur les bornes de recharge de véhicules électrique en pays de la loire, Chaque étudiant sera chargé de travailler sur les données concernant un seul département ou deux. Il est responsable de ces données avec les contraintes d'intégrités. Il est aussi responsable des contraintes avec les données des autres membres du groupe.

**Ces Données proviennent du site de la région :**

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

**Vous utilisez les données qui se trouvent sur madoc :**

Vous trouvez les informations 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.
- etc....

### Exercice 1 :

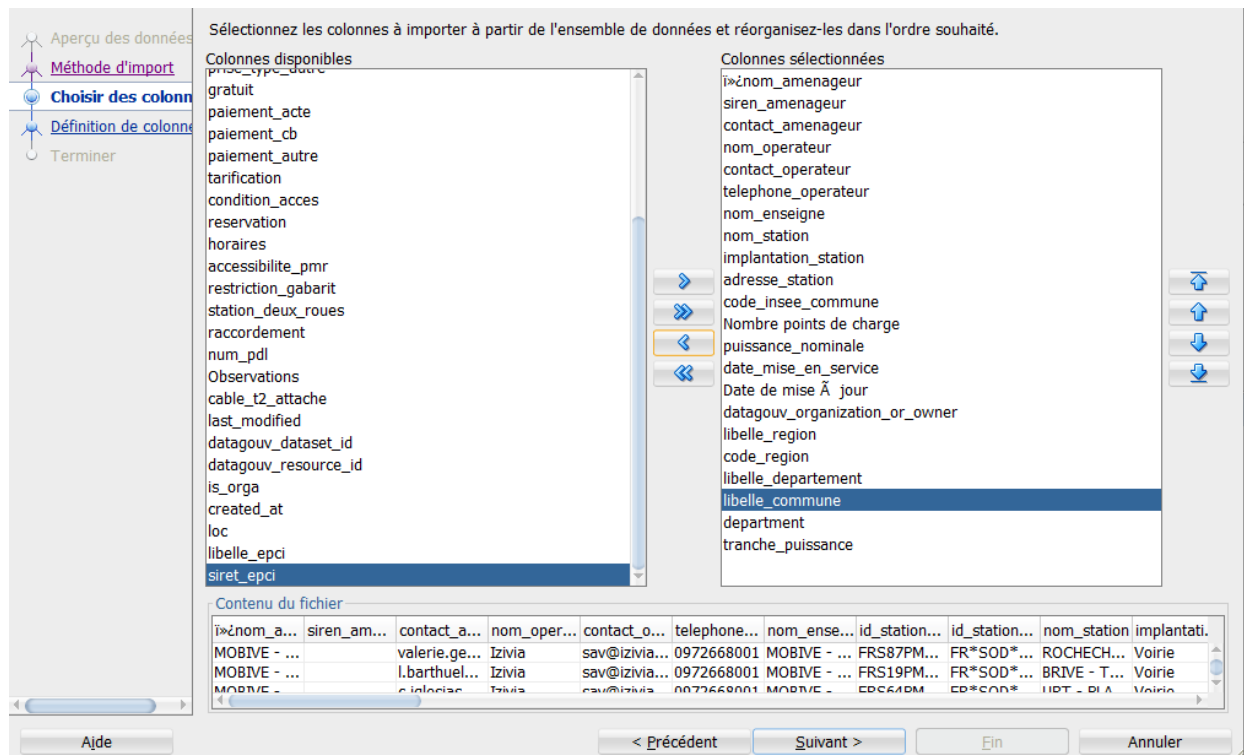
**Avant de commencer à répondre aux différentes questions, vous supprimez toutes les tables qui sont créées avec la requête suivante :**

**Drop table *nom\_table* cascade constraints purge ;**

Dans un premier temps vous allez importer les données qui sont en format **csv** à partir de SqlDeveloper. Vous le faites au moins une fois pour se familiariser avec cet import.

Vous importez que les colonnes qui se trouvent à droite(voir figure)

- Tables....importer des Données
- Choisir le fichier csv et le délimiteur de ce fichier(**normalement le délimiteur est un ;**)
- Suivant...Vous donnez le Nom à votre table. Cette table **plate est en première Forme Normale.**
- Suivant...Sélectionnez les colonnes à importer (voir Figure ci-dessous)...Suivant



- Maintenant vous avez des colonnes qui sont **erronées** :

#### Exemples :

- Date de Mise à jour est remplacée par **Date\_de\_Mise\_à\_jour** ou un autre nom.
- Le nom de l'attribut est trop long.
- La taille des valeurs des attributs dépasse la taille max dans la base de données ; **Dans ce cas vous mettez la taille à 128 caractères pour les attributs erronés ou tous les attributs de type VARCHAR2.** Exemple : *nom\_amenageur*; A vous de les trouver.

**Attention à chaque itération vous devriez supprimer la table si celle-ci est créée. Au final votre table contiendra 3723 lignes.**

- 1) Avant la décomposition de la table en plusieurs tables, vous devriez mettre à jour les données : Dans ce cas, un du groupe possède la table créée et donne le droit aux autres de faire des update dans cette table(chacun fait des *update* sur des colonnes bien identifiées).

Exemple :

```
update login.Nom_table set nom_amenageur='Ionity'
where contact_amenageur like '%ionity%';
```

Pour les codes INSEE vous utilisez les codes\_insee de *basetd*.

- 2) Décomposez cette table en plusieurs tables. Vous pouvez fournir le **modèle conceptuel de données(ou schéma entité-association)**. A partir de ce modèle, déduisez le schéma relationnel qui doit-être en **3ième forme normale**. Nous supposons que une borne de recharge électrique est liée à un **opérateur**, une **station** et à un **aménageur**.

### Explications :

**3FN** : Les attributs non clés d'une relation sont en relation **directe avec la clé**.

**Un seul compte fait la décomposition.**

**Exemple** : T(....., code\_insee, libelle\_commune, ....) est

**Commune (code\_insee, libelle\_commune,.....).**

La table commune est créée à partir de la table T. C.à.d faire que la commune contient tout les attributs qui la concerne et se trouvent dans la table T.

Après suppression de ces attribut, sauf le code\_insee qui fera le lein.

Exemple : le libellé est supprimé de la table T.

**Deuxième exemple : voir ci-dessous.**

**Exemple** : Soit le regroupement de plusieurs attributs et création d'une clé primaire avec un auto incrément.

Soit la relation principale T(....., a1,a2,a3,.....).

- Création d'une nouvelle relation avec les attributs a1,a2,a3 et une clé primaire avec un auto-incrément.

**Create table T1(**

id\_T1 NUMBER GENERATED ALWAYS as IDENTITY(START with 1 INCREMENT by 1),

a1.....,

a2.....,

a3....)

- Insertion dans la table T1 à partir d'un select des attributs a1,a2,a2 de **la table T**. Vous utilisez la clause distinct.
- Lier la table T avec T1, en ajoutant un attribut dans T(id\_T1), **qui sera la clé étrangère**.
- Modifier la valeur de cet attribut. Le lien se fait entre les attributs(a1,a2,a3) de T et de T1..
- Supprimer les attributs **a1,a2,a3 de T**, car le lien est fait avec la clé étrangère(bien vérifier avant).

--

- 2) Répartissez et créez les tables. Attention les données doivent être partagées(Exemple OperateurLA de TD4-exertcice3). Chaque étudiant doit travailler sur ces propre données.

Après cette décomposition en plusieurs tables, chaque étudiant doit travailler sur ses propres données. Par exemple avoir toutes les communes de Loire Atlantique, les bornes dans ces communes, etc.

Si une table est identique quelque soit le département, alors cette table est sur un seul compte.

Des erreurs peuvent se produire lors de cette répartition.

Select .....where attribut = (select distinct attribut from .....) ; // Erreur de plusieurs lignes dans le deuxième select.

Solution : Rajouter un Fetch next 1 row only dans le deuxième select.

Select .....where attribut = (select distinct attribut from .....  
Fetch next 1 row only) ;

- 3) Vous devez maintenant créer les contraintes d'intégrité PK et FK des tables en locale.

Idem que le TD4

- 4) Créez maintenant les contraintes qui sont distants. Donnez les différents droits qui sont utilisés.

Idem TD4 exercice 3 ( Dans le cas où une table est sur un seul compte).

- 5) Donnez les requêtes suivantes : **Chaque étudiant du groupe travaille sur 2 fonctionnalités, avec accès aux données de son groupe. Vous indiquez dans le rapport les différents GRANT donnés ou reçus pour les tests.**
- Affichez pour chaque opérateur le nombre de prises(points de charge) par département.
  - Afficher pour chaque opérateur le nombre de prises avec la puissance nominale.
  - Affichez pour chaque département la répartition des prises. C'est à dire en fonction des valeurs de la puissance nominale(22, 50, 43, 60, 120,...). Dans ce cas vous prenez les 15 meilleurs prises qui sont installées.
  - Les bornes les plus puissantes de chaque commune
  - L'Aménageur qui a posé le plus de bornes
  - Proposez 2 autres types de requêtes.

**Pour le point 3 .**

Les requêtes de SUM sur un champ VARCHAR est impossible.

Traduire le VARCHAR en NUMBER d'un attribut.

SUM(TO\_NUMBER(replace(attribut,' ','')));

- 6) Vous générez aussi différents graphiques(en batons et/ou batons empilés).

7) Faites une synthèse de ces requêtes en passant par des vues.

**Remarque :** Idem que la Q5 , mais en passant par des vues de chaque département. Après, une vue globale qui fait la synthèse de toutes les vues à chaque fois.

**Exercice 2 :** Proposez un jeu ou plusieurs jeux de données des Pays de Loire ou de toutes les régions. Et refaire une extraction comme au début avec partage des tables.

Ici refaire les questions jusqu'à la répartition des tables. Pas besoin de requêtes ou de graphiques
--