# Tables de données

tables = champs (colonnes)

chaque colonne a un type (number par exemple --> full off numbers)

chq ligne de données est unique (pas 2 lignes identiques)

chaque colonne ne contient qu'une valeur (une colonne pas + d'un e-mail par exemple, dans une même case)

Comme types de données on retrouve

\*\*particulier : colonne id = clé primaire (identifie chaque ligne des tables)

bdd relationelles --> reliées par clefs primaires & étrangères

# Types de données :

## Types numériques

* + INT : **Integer** – All numbers
  + NUMERIC(décimaux) – préciser P et S (maximum de chiffre acceptable, et le second = nombre de chiffres après la virgule)
  + SERIAL – propre à postGreSQL, il stock un integer. S’incrémente automatiquement de 1 (pratique pour id(clé primaire)) :
    - On crée sa table réalisateurs, le champ « realisateurs\_id » en type Serial et on aura les id de 1 à x…

## Types de string

* + CHAR(N) – fixes – (N) = paramètre max ex CHAR(2)
  + VARCHAR – varié (entre 1 et maximum)
  + TEXT – pas de max

## Types de date

* + TIME - Pas très utilisé
  + DATE –
  + TIMESTAMP –

## Autres types

* + Enum : on crée une liste de valeurs qui peuvent être choisis par l’utilisateur

# Clés primaires & étrangères

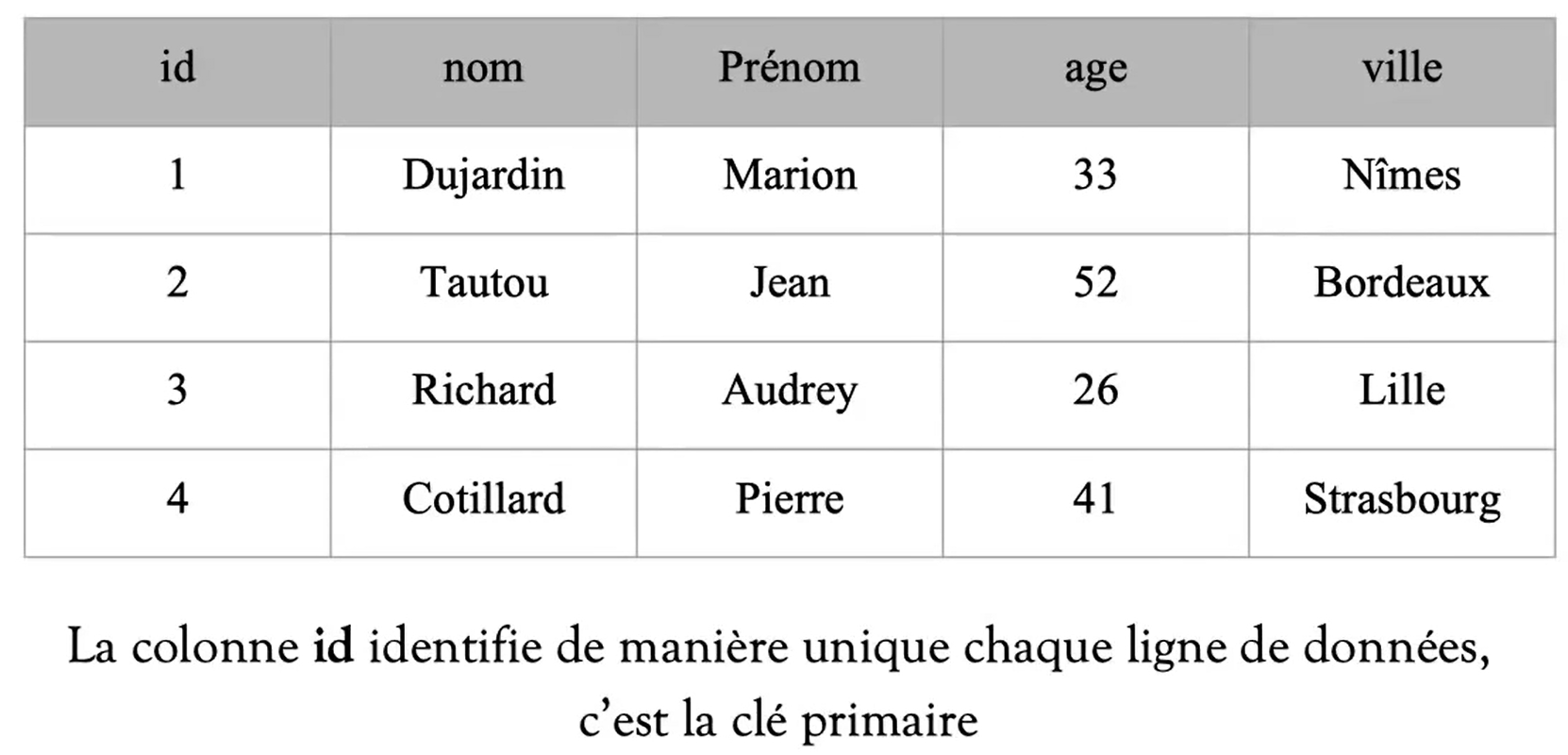
## Clé primaire

Colonne qui identifie une ligne de donnée dans une table

Unique et ne peut être nulle

1 seule par table

Pas obligatoire mais très recommandé



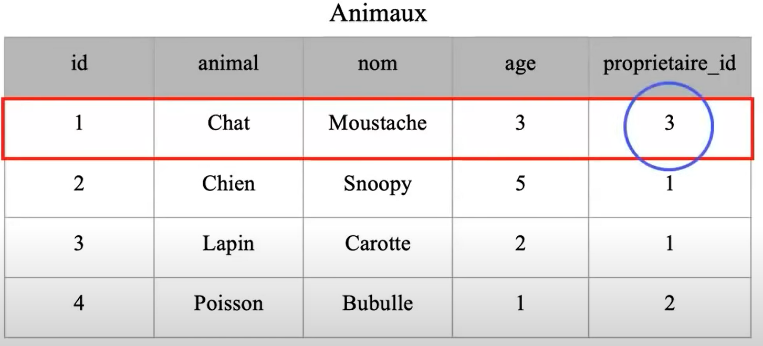
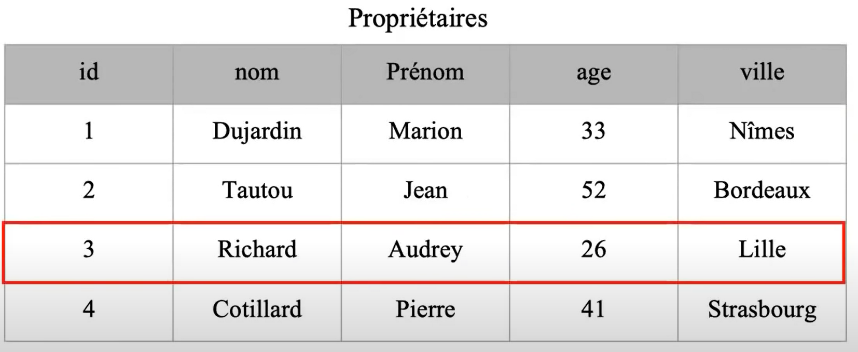
## Clé étrangère

Utilisée pour relier 2 tables ensembles

Ses valeurs correspondent à la colonne d’une clé primaire de la 2ème table

La table avec la clé primaire = référence // table parent

Celle avec clé étrangère = table enfant

Une table peut contenir plusieurs clés étrangères ou pas du tout.

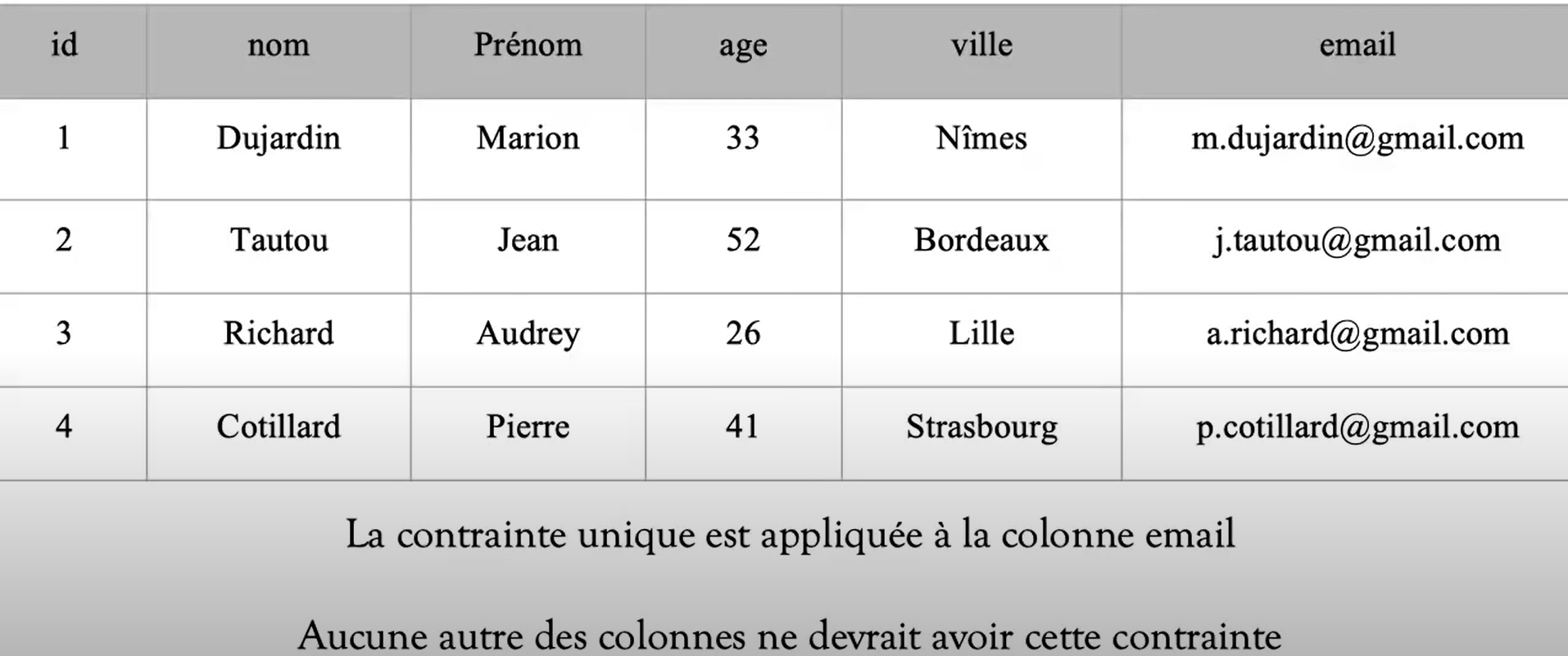
# Contraintes

## Unique

Permet de s’assurer qu’elles soient toutes uniques, aucune en double

Envoie une erreur si une valeur est dupliquée

On l’indique lors de la création



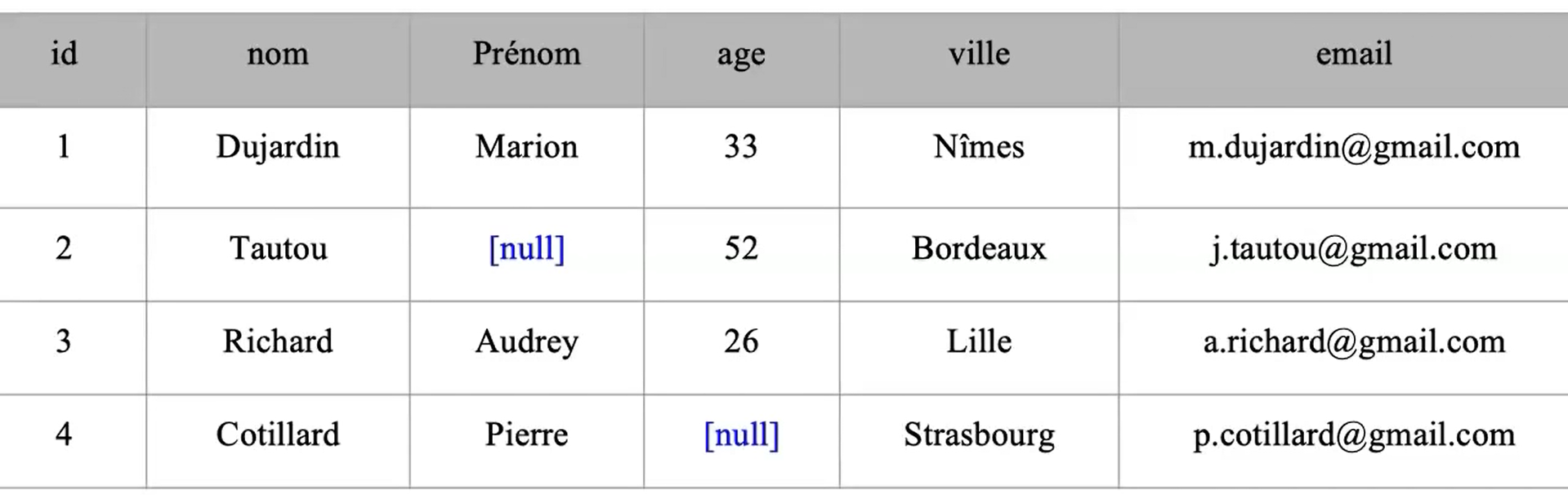
Aucun ne doit avoir le même e-mail donc contrainte Unique

Parcontre nom prénom ville etc. peuvent être partagés par plusieurs lignes

Id = unique par défaut

## Non-null

S’assurer qu’aucun champ ne soit null exemple on veut absolument un e-mail, on met la contrainte not-null à ce champ



Ça ne gêne pas qu’ils ne donnent pas prénom ou âge, mais on veut obligatoirement nom ou e-mail, not-null est appliqué à ces deux

## Check

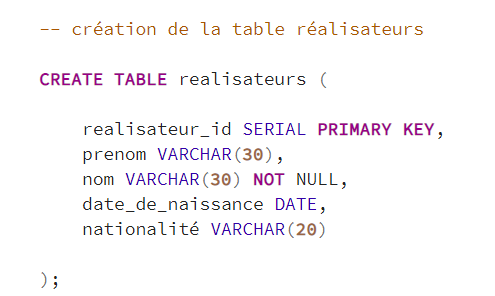
Vérifie si la valeur répond à boolean true ou false

Exemple une colonne âge doit contenir val > à 0

Ou encore s’assurer qu’on laisse une note entre 1 et 5 (étoiles pour noter)

# Création base de données

## 1ère table



create table (nom à donner) (

nom du champ + type + propriété s’il y a

);

## Modification des tables

### COLUMN

#### Add Column :

ALTER TABLE nom\_table ADD COLUMN nom\_column type

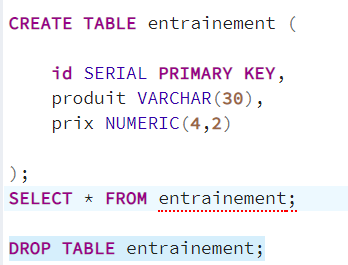
#### Rename Column :

ALTER TABLE nom\_table RENAME COLUMN ancien\_nom \_colonne TO nouveau\_nom\_colonne

### TABLE

#### Remove table

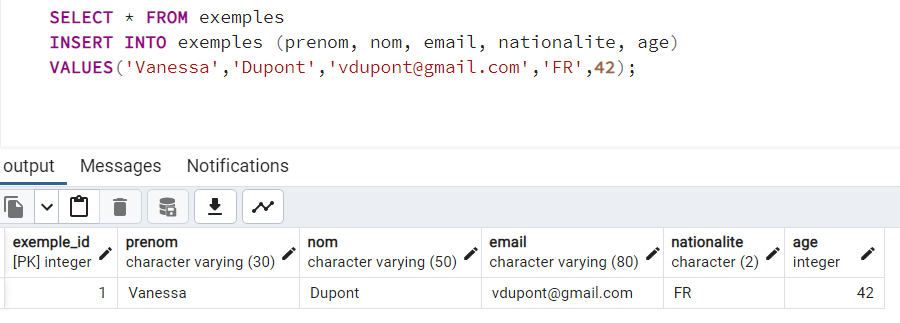
DROP TABLE nom\_table



### INSERTION DE DONNEES

INSERT INTO nom\_de\_la\_table (colonne1, col2…)

VALUES (‘valeur1’,’valeur2’…)



Pour les INTEGER, pas de signe typographique " ou ‘

#### Modifier données dans la table

UPDATE nom\_table

SET nom\_colonne = ‘nouvelle\_valeur’

WHERE nom\_colonne = ‘valeur’

### SUPPRESSION DE DONNEES

DELETE FROM nom\_table

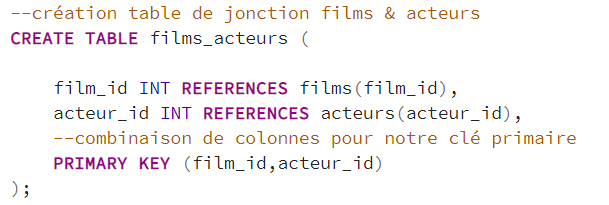
WHERE nom\_de la ligne avec id = ‘valeur’



## Table de jonction

Relie les tables entre elles

Ne peut avoir que 2 colonnes – les clefs étrangères



## Résumé introduction

Docu

LIMIT 🡪 limit number of results

Exo SQL

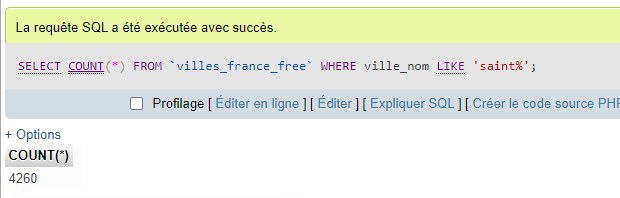
Enoncé

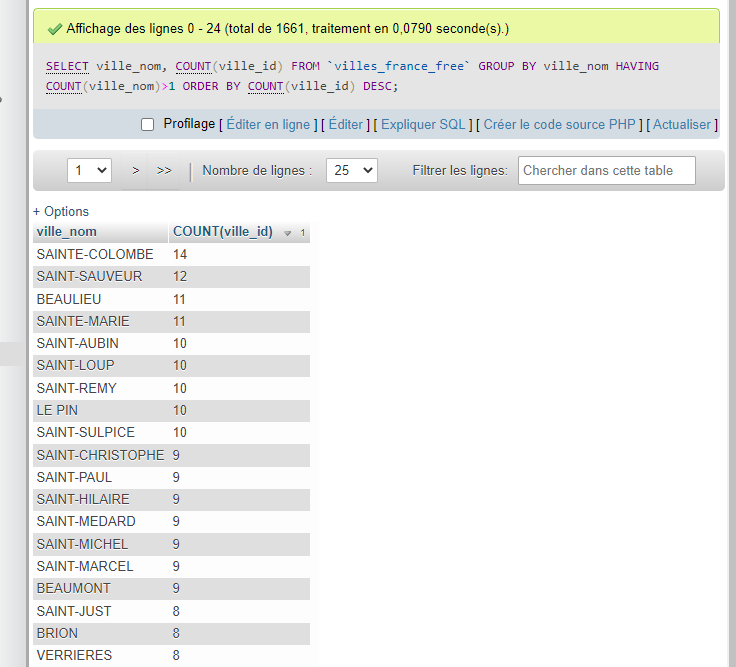
1. Obtenir la liste des 10 villes les plus peuplées en 2012  
2. Obtenir la liste des 50 villes ayant la plus faible superficie  
3. Obtenir la liste des départements d’outres-mer, c’est-à-dire ceux dont le numéro de département commencent par “97”  
4. Compter le nombre de villes dont le nom commence par “Saint”  
5. Obtenir la liste des villes qui ont un nom existants plusieurs fois, et trier afin d’obtenir en premier celles dont le nom est le plus souvent utilisé par plusieurs communes  
6. Obtenir en une seule requête SQL la liste des villes dont la superficie est supérieur à la superficie moyenne  
7. Obtenir la liste des départements qui possèdent plus de 2 millions d’habitants  
8. Remplacez les tirets par un espace vide, pour toutes les villes commençant par “SAINT-” (dans la colonne qui contient les noms en majuscule)  
9. Obtenir le nom des 10 villes les plus peuplées en 2012, ainsi que le nom du département associé  
10. Obtenir la liste du nom de chaque département, associé à son code et du nombre de commune au sein de ces département, en triant afin d’obtenir en priorité les départements qui possèdent le plus de communes  
11. Obtenir la liste des 10 plus grands départements, en terme de superficie

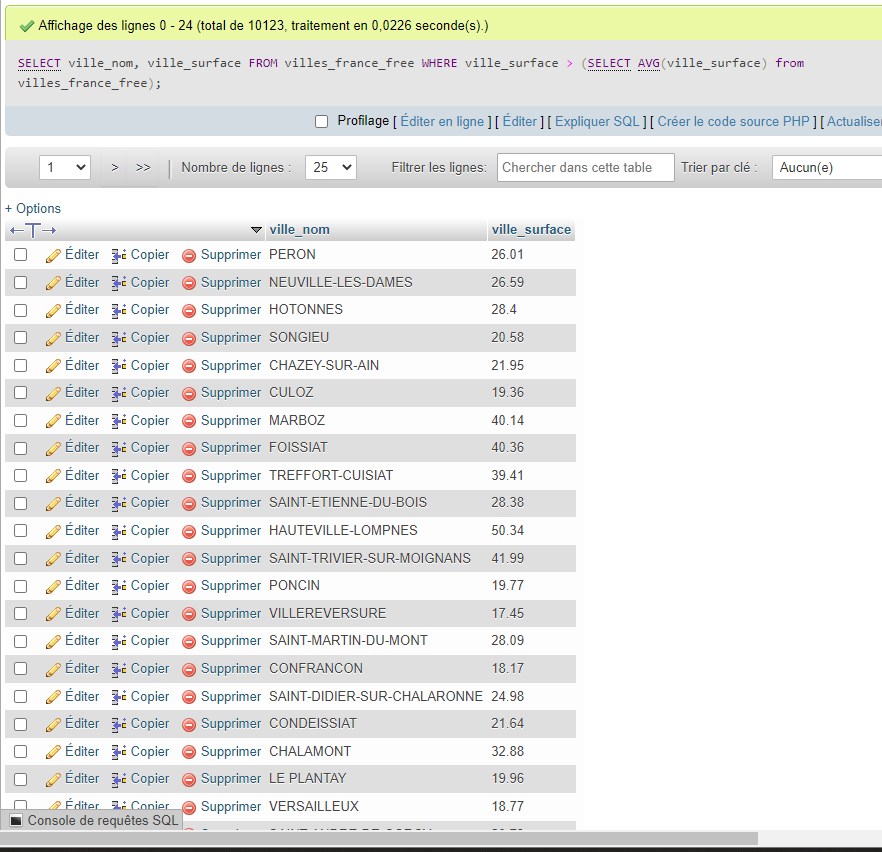
1 - [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) \* FROM `villes\_france\_free` ORDER BY ville\_population\_2012 DESC LIMIT 10;

2 – [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) \* FROM `villes\_france\_free` ORDER BY ville\_surface ASC LIMIT 50;

3 - [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/select.html) \* FROM `villes\_france\_free` WHERE ville\_code\_postal [LIKE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.7/en/string-comparison-functions.html#operator_like) '97%';

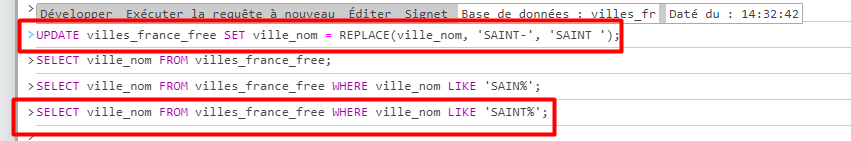
4 - 

5 - 

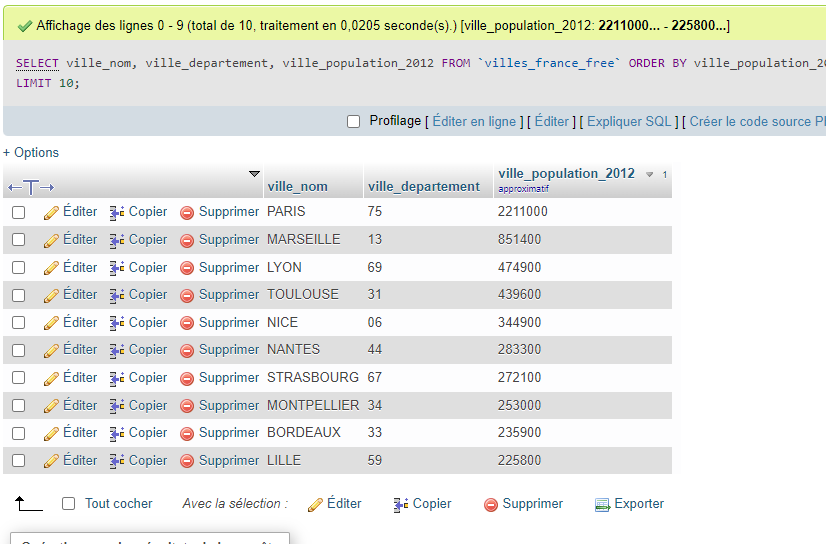
6 - 

7 

8



9



10

Obtenir NOM DEPARTEMENT, VILLE CODE COMUNE, NOMBRE DE COMMUNES DANS CES DEPARTEMENTS et trier descendant

🡺 obtenir nom département, code commune, commune ayant même département

COMMUNES GROUP BY NOM DEPARTEMENT

10. Obtenir la liste du nom de chaque département, associé à son code et du nombre de commune au sein de ces département, en triant afin d’obtenir en priorité les départements qui possèdent le plus de communes

CONCATENATION

