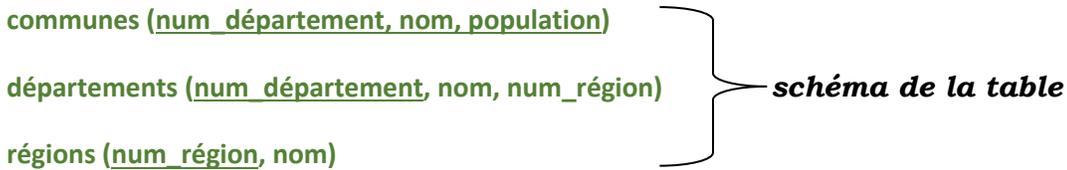


Vous avez à votre disposition la base de données *géographie2017*, qui est composée de 3 tables :



La clé primaire de chaque table est soulignée.

Dans la table **communes**, l'attribut **num_département** est une clé étrangère faisant référence à la clé primaire **num_département** de la table **départements**.

Question : Y a-t-il d'autres clés étrangères dans cette base de données ? Si oui, quelles sont-elles, et à quelle(s) clé(s) primaire(s) font-elles références ?

Dans la table **départements**, l'attribut **num_région** est une clé étrangère faisant référence à la clé primaire **num_région** de la table **régions**.

Précisons qu'à part l'attribut **population** de la table **communes**, qui est de type INTEGER, tous les attributs des trois tables sont de type CHAR (chaîne de caractère)

N.B. : Les villes de Paris, Lyon et Marseille sont répertoriées en étant découpées par arrondissement, ne vous étonnez donc pas si vous ne les voyez pas apparaître quand vous cherchez les communes les plus peuplées

Écrivez les requêtes SQL suivantes et testez-les.

I) Requêtes sans jointure (i.e. portant sur une seule table)

1) Faire afficher le nom de toutes les communes du département '63'.

```
SELECT nom FROM communes WHERE num_département='63'
```

2) Faire afficher le nombre de communes du département '63'. *Vous devez en trouver 467*

```
SELECT COUNT(*) FROM communes WHERE num_département='63'
```

3) Donner la liste des communes du département '63' de moins de 100 habitants. *Dans la liste obtenue doit figurer la commune d'Aix-la-Fayette par exemple.*

```
SELECT nom FROM communes WHERE num_département='63' AND population<=100
```

4) Déterminer la population moyenne des communes françaises. *Vous devez trouver environ 1876 habitants.*

```
SELECT AVG(population) FROM communes
```

5) Classer les communes du département '63' selon leur population (de la plus peuplée à la moins peuplée). On fera apparaître le nom de la commune et sa population. *La liste obtenue doit commencer par Clermont-Ferrand et finir avec une commune de 14 habitants.*

```
SELECT nom,population FROM communes WHERE num_département='63' ORDER BY population DESC
```

6) Combien y a-t-il d'habitants en France? *Vous devez trouver un peu plus de 67 millions d'habitants.*

```
SELECT SUM(population) FROM communes ;
```

7) Donner la liste des numéros de département, avec pour chaque numéro le nombre de communes du département. Afficher un titre explicite pour la colonne des nombres de communes.

```
SELECT num_département,COUNT(nom) AS nb_communes FROM communes  
GROUP BY num_département
```

8) Donner la liste des numéros de département, avec pour chaque numéro le nombre de communes du département, en ne faisant afficher que les départements possédant plus de 800 communes.

```
SELECT num_département,COUNT(nom) AS nb_communes FROM communes  
GROUP BY num_département HAVING nb_communes >=800
```

9) Donner la liste des numéros de département, avec pour chaque numéro la population totale du département. Trier par population totale croissante.

```
SELECT num_département,SUM(population)AS population_totale FROM communes  
GROUP BY num_département ORDER BY population_totale
```

10) Donner la liste des numéros de département, avec pour chaque numéro la population totale du département, en ne faisant afficher que les 5 départements les plus peuplés.

```
SELECT num_département,SUM(population)AS population_totale FROM communes  
GROUP BY num_département ORDER BY population_totale DESC LIMIT 5
```

11) Utilisation d'une sous-requête: Lorsqu'un SELECT renvoie un simple scalaire (par exemple via une fonction d'agrégation), on peut réutiliser ladite requête SELECT. . . (entre parenthèses) comme ledit scalaire.

Donner la liste des noms de communes dont la population excède 100 fois la population moyenne des communes françaises.

```
SELECT nom FROM communes WHERE population>=100*(SELECT AVG(population) FROM communes)
```

II)Utilisation d'une opération ensembliste sur deux requêtes

12) Donner la liste des numéros de départements qui font partie de la région numéro 53, ou bien qui possèdent une commune de moins de 5 habitants.

```
SELECT num_département FROM départements WHERE num_région=53  
UNION  
SELECT num_département FROM communes WHERE population<=5
```

II) Requêtes avec jointures (i.e. portant sur deux tables ou plus)

13) Donner la liste des noms des départements français et le nom de la région à laquelle ils appartiennent.

```
SELECT départements.nom, régions.nom FROM départements JOIN régions  
ON départements.num_région=régions.num_région
```

14) Donner la liste des noms des départements de la région 'Auvergne-Rhône-Alpes'. *Vous n'avez pas le droit de tricher en cherchant avec vos yeux le numéro de la région 'Auvergne-Rhône-Alpes'. Vous devez d'abord faire une jointure des deux tables dont vous avez besoin pour répondre à cette question, à savoir la table « départements » et la table « régions ».*

```
SELECT départements.nom FROM départements JOIN régions  
ON départements.num_région=régions.num_région  
WHERE régions.nom='Auvergne-Rhône-Alpes'
```

15) Donner la liste des noms de tous les départements, avec pour chaque département le nombre de communes. Afficher un titre explicite pour la colonne des nombres de communes.

```
SELECT départements.nom, Count(communes.nom) AS nb_communes  
FROM départements JOIN communes ON départements.num_département=communes.num_département  
GROUP BY départements.nom
```

16) Donner la liste des noms de département, avec pour chaque département la population totale du département. Trier par population totale croissante. *Votre liste doit commencer par la Lozère puis la Creuse.*

```
SELECT départements.nom, SUM(population)AS pop_totale  
FROM départements JOIN communes  
ON départements.num_département=communes.num_département  
GROUP BY départements.nom  
ORDER BY pop_totale
```

17) Donner la liste des noms de département qui comptent moins de 200 000 habitants.

```
SELECT départements.nom FROM départements JOIN communes  
ON départements.num_département=communes.num_département  
GROUP BY départements.nom HAVING SUM(population)<200000
```

18) Donner sans doublon les noms des départements possédant une commune où la population dépasse 100 000 habitants.

```
SELECT DISTINCT départements.nom FROM départements JOIN communes  
ON départements.num_département=communes.num_département  
WHERE population>100000
```

19) Combien de départements possèdent une commune où la population dépasse 100 000 habitants ? *Vous devez trouver 36 départements.*

```
SELECT COUNT(DISTINCT départements.nom) FROM départements JOIN communes  
ON départements.num_département=communes.num_département  
WHERE population>100000
```

20) Donner sans doublons la liste des noms des départements ne contenant aucune commune de moins de 1000 habitants. *Vous devez obtenir des départements d'outre-mer et de la région parisienne.*

```
SELECT départements.nom FROM départements
EXCEPT
SELECT DISTINCT(départements.nom) FROM départements JOIN communes
ON départements.num_département=communes.num_département
WHERE population<=1000
```

Les questions 21) , 22) et 23) utilisent les trois tables à la fois:

21) Donner la liste des noms de départements de la région 'Auvergne-Rhône-Alpes', avec pour chaque nom la population totale du département. Trier par population totale décroissante.

```
SELECT départements.nom,SUM(population) AS pop_totale FROM départements
      JOIN régions ON départements.num_région=régions.num_région
      JOIN communes ON départements.num_département=communes.num_département
WHERE régions.nom='Auvergne-Rhône-Alpes'
GROUP BY départements.nom
ORDER BY pop_totale DESC
```

22) Donner sans doublons la liste des noms des régions ne contenant aucune commune de moins de 1000 habitants. *Vous devez obtenir seulement deux régions : Guadeloupe et La Réunion.*

```
SELECT régions.nom FROM régions
EXCEPT
SELECT régions.nom FROM départements
      JOIN régions ON départements.num_région=régions.num_région
      JOIN communes ON départements.num_département=communes.num_département
WHERE population<=1000
```

23) Donner le nom de la région contenant la (les) commune(s) la(les) moins peuplée(s) de France. On fera aussi apparaître le nom et la population de cette(ces) commune(s).Interprétation historique?

```
SELECT régions.nom, communes.nom,population FROM départements
      JOIN régions ON départements.num_région=régions.num_région
      JOIN communes ON départements.num_département=communes.num_département
WHERE population=
(SELECT MIN(population) FROM départements
      JOIN régions ON départements.num_région=régions.num_région
      JOIN communes ON départements.num_département=communes.num_département)
```

Ce sont des villages entièrement détruits pendant la bataille de Verdun qui n'ont jamais été reconstruits.

24) Donner sans doublons la liste des noms de commune possédant un ou des homonymes.

```
SELECT nom,COUNT(nom) FROM communes GROUP BY nom
HAVING COUNT(nom)>1
```

24) Donner la proportion de communes françaises possédant un ou des homonymes.

```
SELECT 1.0*SUM(homonymes)/(SELECT COUNT(nom) FROM communes) FROM
(SELECT nom,COUNT(nom) AS homonymes FROM communes GROUP BY nom HAVING COUNT(nom)>1 )
```