

TD 2 - Bases de Données relationnelles

Informatique
Lycée Thiers

2014/2015

La base de données `geographie.sqlite` contient trois tables :

- La table `communes` contient toutes les communes de France. Ses attributs sont :
 - `num_departement` : le numéro du département,
 - `nom` : le nom de la commune,
 - `canton`,
 - `population_2010` : le nombre d'habitants lors du recensement de 2010,
 - `population_1999` : le nombre d'habitants lors du recensement de 1999,
 - `surface` : la surface de la commune en km^2 ,
 - `longitude` : la longitude du centre de la commune,
 - `latitude` : la latitude du centre de la commune,
 - `zmin` : l'altitude minimale de la commune,
 - `zmax` : l'altitude maximale de la commune,
- La table `departements` contient tous les départements de France. Ses attributs sont :
 - `num_departement` : le numéro du département,
 - `num_region` : le numéro de la région,
 - `nom` : le nom du département,
- La table `regions` contient toutes les régions de France. Ses attributs sont :
 - `num_region` : le numéro de la région,
 - `nom` : le nom de la région,

Exercice 1 : Enoncé

Exercice 1. *Requêtes simples.*

1. Obtenez la liste des noms de communes des Bouches-du-Rhône, classés par ordre alphabétique.
2. Combien Y-a-t-il d'habitants en France 2010 ? (en supposant que chaque habitant est rattaché à une commune).
3. Quel est le pourcentage d'accroissement de la population entre 1999 et 2010 ?
4. Quels sont les 10 communes de France les plus peuplées (en 2010) ?
5. Quels sont les 10 communes de France les plus densément peuplées (en 2010) ?
6. Quelle commune a l'altitude la plus élevée ?
7. Quelles sont les 10 communes dont la population a le plus augmenté : en augmentation absolue ? en augmentation relative ?
8. Obtenez la liste des numéros de départements, et pour chacun son nombre de communes.
9. Obtenez la liste des numéros de départements, et pour chacun sa population totale (en 2010), ordonnés par population décroissante.

Exercice 1 : Corrigé

1)

```
SELECT nom FROM communes WHERE num_departement = 13 ORDER BY nom;
```

Exercice 1 : Corrigé

1)

```
SELECT nom FROM communes WHERE num_departement = 13 ORDER BY nom;
```

2)

```
SELECT SUM(population_2010) FROM communes;
```

Exercice 1 : Corrigé

1)

```
SELECT nom FROM communes WHERE num_departement = 13 ORDER BY nom;
```

2)

```
SELECT SUM(population_2010) FROM communes;
```

3) Remarquer l'utilisation de 100.0 pour avoir un résultat non entier :

```
SELECT 100.0*(SUM(population_2010)-SUM(population_1999))/SUM(population_1999) FROM communes;
```

Exercice 1 : Corrigé

1)

```
SELECT nom FROM communes WHERE num_departement = 13 ORDER BY nom;
```

2)

```
SELECT SUM(population_2010) FROM communes;
```

3) Remarquer l'utilisation de 100.0 pour avoir un résultat non entier :

```
SELECT 100.0*(SUM(population_2010)-SUM(population_1999))/SUM(population_1999) FROM communes;
```

4)

```
SELECT nom FROM communes ORDER BY population_2010 DESC LIMIT 10;
```


Exercice 1 : Corrigé

1)

```
SELECT nom FROM communes WHERE num_departement = 13 ORDER BY nom;
```

2)

```
SELECT SUM(population_2010) FROM communes;
```

3) Remarquer l'utilisation de 100.0 pour avoir un résultat non entier :

```
SELECT 100.0*(SUM(population_2010)-SUM(population_1999))/SUM(population_1999) FROM communes;
```

4)

```
SELECT nom FROM communes ORDER BY population_2010 DESC LIMIT 10;
```

5)

```
SELECT nom, population_2010/surface AS densite FROM communes ORDER BY densite DESC LIMIT 10;
```

Exercice 1 : Corrigé

1)

```
SELECT nom FROM communes WHERE num_departement = 13 ORDER BY nom;
```

2)

```
SELECT SUM(population_2010) FROM communes;
```

3) Remarquer l'utilisation de 100.0 pour avoir un résultat non entier :

```
SELECT 100.0*(SUM(population_2010)-SUM(population_1999))/SUM(population_1999) FROM communes;
```

4)

```
SELECT nom FROM communes ORDER BY population_2010 DESC LIMIT 10;
```

5)

```
SELECT nom, population_2010/surface AS densite FROM communes ORDER BY densite DESC LIMIT 10;
```

6)

```
SELECT nom, MAX(zmax) FROM communes;
```

Exercice 1 : Corrigé

7) En population absolue :

```
SELECT nom, population_2010-population_1999 AS accroissement FROM communes ORDER BY  
accroissement DESC LIMIT 10;
```

Exercice 1 : Corrigé

7) En population absolue :

```
SELECT nom, population_2010-population_1999 AS accroissement FROM communes ORDER BY  
accroissement DESC LIMIT 10;
```

En population relative : (remarquer 1.0 pour avoir un résultat son entier) :

```
SELECT nom, 1.0*(population_2010-population_1999)/population_1999 AS accroissement  
FROM communes ORDER BY accroissement DESC LIMIT 10;
```

Exercice 1 : Corrigé

7) En population absolue :

```
SELECT nom, population_2010-population_1999 AS accroissement FROM communes ORDER BY  
accroissement DESC LIMIT 10;
```

En population relative : (remarquer 1.0 pour avoir un résultat son entier) :

```
SELECT nom, 1.0*(population_2010-population_1999)/population_1999 AS accroissement  
FROM communes ORDER BY accroissement DESC LIMIT 10;
```

9)

```
SELECT num_departement, COUNT(nom) FROM communes GROUP BY num_departement;
```

Exercice 1 : Corrigé

7) En population absolue :

```
SELECT nom, population_2010-population_1999 AS accroissement FROM communes ORDER BY  
accroissement DESC LIMIT 10;
```

En population relative : (remarquer 1.0 pour avoir un résultat son entier) :

```
SELECT nom, 1.0*(population_2010-population_1999)/population_1999 AS accroissement  
FROM communes ORDER BY accroissement DESC LIMIT 10;
```

9)

```
SELECT num_departement, COUNT(nom) FROM communes GROUP BY num_departement;
```

10)

```
SELECT num_departement, SUM(population_2010) AS population FROM communes GROUP BY  
num_departement ORDER BY population DESC;
```

Exercice 2 : Enoncé

Exercice 2. Jointures à deux tables

1. Obtenez la liste des noms de départements, et pour chacun son nombre de communes.
2. Obtenez la liste des noms de régions, et pour chacun son nombre de départements.
3. Obtenez la liste des noms de départements des régions Basse Normandie et Haute Normandie.
4. Quelle est la population (recensement 2010) de chaque département ; les résultats seront retournés par nom de département et par population décroissante.
5. Quelle est la densité de population de chaque département ; les résultats seront retournés par nom de département et par densité décroissante.
6. Donner la liste des (noms de) départements dont la population est supérieure à 1 000 000 d'habitants.
(Indication : soit une requête composée soit ... GROUP BY ... HAVING condition ... pour une agrégation sous condition).

Exercice 2 : Corrigé

(1)

```
SELECT departements.nom, COUNT(communes.nom)
FROM departements JOIN communes
ON communes.num_departement = departements.num_departement
GROUP BY communes.num_departement;
```


Exercice 2 : Corrigé

(1)

```
SELECT departements.nom, COUNT(communes.nom)
FROM departements JOIN communes
ON communes.num_departement = departements.num_departement
GROUP BY communes.num_departement;
```

2)

```
SELECT regions.nom, COUNT(departements.nom)
FROM regions JOIN departements
ON regions.num_region = departements.num_region
GROUP BY regions.nom;
```

Exercice 2 : Corrigé

(1)

```
SELECT departements.nom, COUNT(communes.nom)
FROM departements JOIN communes
ON communes.num_departement = departements.num_departement
GROUP BY communes.num_departement;
```

2)

```
SELECT regions.nom, COUNT(departements.nom)
FROM regions JOIN departements
ON regions.num_region = departements.num_region
GROUP BY regions.nom;
```

3)

```
SELECT departements.nom
FROM departements JOIN regions
ON departements.num_region = regions.num_region AND (regions.nom = "Basse
Normandie" OR regions.nom = "Haute Normandie");
```

Exercice 2 : Corrigé

(1)

```
SELECT departements.nom, COUNT(communes.nom)
FROM departements JOIN communes
ON communes.num_departement = departements.num_departement
GROUP BY communes.num_departement;
```

2)

```
SELECT regions.nom, COUNT(departements.nom)
FROM regions JOIN departements
ON regions.num_region = departements.num_region
GROUP BY regions.nom;
```

3)

```
SELECT departements.nom
FROM departements JOIN regions
ON departements.num_region = regions.num_region AND (regions.nom = "Basse
Normandie" OR regions.nom = "Haute Normandie");
```

4)

```
SELECT departements.nom, SUM(communes.population_2010)
FROM departements JOIN communes
ON communes.num_departement = departements.num_departement
GROUP BY departements.nom
ORDER BY departements.nom DESC;
```

Exercice 2 : Corrigé

(5)

```
SELECT departements.nom, SUM(communes.population_2010)/SUM(communes.surface) AS  
densite  
FROM departements JOIN communes  
ON communes.num_departement=departements.num_departement  
GROUP BY departements.nom  
ORDER BY densite DESC;
```

Exercice 2 : Corrigé

(5)

```
SELECT departements.nom, SUM(communes.population_2010)/SUM(communes.surface) AS  
densite  
FROM departements JOIN communes  
ON communes.num_departement=departements.num_departement  
GROUP BY departements.nom  
ORDER BY densite DESC;
```

6) Avec ORDER BY ... HAVING ... :

```
SELECT departements.nom, SUM(population_2010) AS pop  
FROM departements JOIN communes  
ON departements.num_departement=communes.num_departement  
GROUP BY departements.nom HAVING pop >= 1000000;
```

Exercice 2 : Corrigé

(5)

```
SELECT departements.nom, SUM(communes.population_2010)/SUM(communes.surface) AS
densite
FROM departements JOIN communes
ON communes.num_departement=departements.num_departement
GROUP BY departements.nom
ORDER BY densite DESC;
```

6) Avec ORDER BY ... HAVING ... :

```
SELECT departements.nom, SUM(population_2010) AS pop
FROM departements JOIN communes
ON departements.num_departement=communes.num_departement
GROUP BY departements.nom HAVING pop >= 1000000;
```

Avec une requête composée :

```
SELECT nom, pop
FROM (SELECT departements.nom, SUM(population_2010) AS pop
FROM departements JOIN communes
ON departements.num_departement= communes.num_departement
GROUP BY departements)
WHERE pop >= 1000000;
```

Exercice 3 : Enoncé

Exercice 3. *Jointures à trois tables*

1. Quelles sont les 10 communes de France d'altitudes les plus élevées, et pour chacune son nom de département, et son nom de région ?
2. Obtenez la liste des noms de régions par ordre alphabétique, et pour chacune son nombre de communes.
3. Obtenez la liste des noms des 10 régions les plus peuplées, et pour chacune son nombre d'habitants, ordonnée par population décroissante.

Exercice 3 : Corrigé

1)

```
SELECT communes.nom, departements.nom, regions.nom, communes.zmax  
FROM communes JOIN departements JOIN regions  
ON communes.num_departement=departements.num_departement AND  
departements.num_region=regions.num_region  
ORDER BY communes.zmax DESC LIMIT 10;
```


Exercice 3 : Corrigé

1)

```
SELECT communes.nom, departements.nom, regions.nom, communes.zmax  
FROM communes JOIN departements JOIN regions  
ON communes.num_departement=departements.num_departement AND  
departements.num_region=regions.num_region  
ORDER BY communes.zmax DESC LIMIT 10;
```

2)

```
SELECT regions.nom, COUNT(communes.nom)  
FROM regions JOIN departements JOIN communes  
ON communes.num_departement=departements.num_departement AND  
departements.num_region=regions.num_region  
GROUP BY regions.nom;
```

Exercice 3 : Corrigé

1)

```
SELECT communes.nom, departements.nom, regions.nom, communes.zmax
FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region
ORDER BY communes.zmax DESC LIMIT 10;
```

2)

```
SELECT regions.nom, COUNT(communes.nom)
FROM regions JOIN departements JOIN communes
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region
GROUP BY regions.nom;
```

3)

```
SELECT regions.nom, SUM(communes.population_2010) AS pop
FROM regions JOIN departements JOIN communes
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region
GROUP BY regions.nom
ORDER BY pop DESC LIMIT 10;
```

Exercice 4 : Enoncé

Exercice 4. Miscellaneous (*INTERSECT, EXISTS*)

1. *Quels communes d'auvergne et d'aquitaine sont homonymes ?*
2. *Déterminer la liste des régions dont un département a un nom commençant par "B".*
3. *Déterminer la liste des régions dont une commune a un nom commençant par "Marse".*
4. *Quelle communes de Haute-Garonne (31) ont une commune homonyme dans le département de la Dordogne (24) ?*
5. *Quelles communes des Bouches-du-Rhône ont une commune homonyme (dans un autre département) ?*

Exercice 4 : Corrigé

1)

```
SELECT communes.nom FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region AND regions.nom="Auvergne"
INTERSECT
SELECT communes.nom FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region AND regions.nom="Aquitaine";
```

Exercice 4 : Corrigé

1)

```
SELECT communes.nom FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region AND regions.nom="Auvergne"
INTERSECT
SELECT communes.nom FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region AND regions.nom="Aquitaine";
```

2)

```
SELECT nom FROM regions
WHERE EXISTS (SELECT * FROM departements WHERE departements.num_region =
regions.num_region AND departements.nom LIKE "B%");
```

Exercice 4 : Corrigé

1)

```
SELECT communes.nom FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region AND regions.nom="Auvergne"
INTERSECT
SELECT communes.nom FROM communes JOIN departements JOIN regions
ON communes.num_departement=departements.num_departement AND
departements.num_region=regions.num_region AND regions.nom="Aquitaine";
```

2)

```
SELECT nom FROM regions
WHERE EXISTS (SELECT * FROM departements WHERE departements.num_region =
regions.num_region AND departements.nom LIKE "B%");
```

3)

```
SELECT nom FROM regions
WHERE EXISTS (SELECT * FROM departements JOIN communes ON
communes.num_departement=departements.num_departement
AND departements.num_region=regions.num_region AND communes.nom LIKE 'Marse%');
```

Exercice 4 : Corrigé

4)

```
SELECT nom FROM communes WHERE num_departement = 31  
INTERSECT  
SELECT nom FROM communes WHERE num_departement = 24;
```

Exercice 4 : Corrigé

4)

```
SELECT nom FROM communes WHERE num_departement = 31  
INTERSECT  
SELECT nom FROM communes WHERE num_departement = 24;
```

5)

```
SELECT nom FROM communes WHERE num_departement = 13  
INTERSECT  
SELECT nom FROM communes WHERE NOT num_departement = 13;
```