

1 Introduction

Une base de données est un regroupement d'informations classée dans des **tables**. Une table est représentée par un tableau dont les colonnes sont les **attributs**.

On n'aborde pas ici les questions de création et de modification d'une base de données : on se concentre sur l'aspect consultation : étant donnée une base de données, comment en extraire les informations que l'on souhaite ?

Un **langage de requête** est un langage informatique utilisé pour accéder aux données d'une base de données : il est ici question du langage le plus utilisé : le langage **SQL**.

2 Requêtes élémentaires

Dans ce TP, on interroge une base de données geo.sqlite à télécharger sur le cahier de texte d'informatique. Ouvrez la base de données dans le logiciel DB Browser (à télécharger également à partir du lien sur le cahier de texte).

- Dans l'onglet "**Structure de la Base de données**", vous pouvez observer les différentes tables et leurs attributs. Chaque attribut a un type de donnée : essayez de deviner à quoi correspondent les différents types de données annoncés.
- Dans l'onglet "**Parcourir les données**", vous pouvez directement lire le contenu des tables. Comme vous pourrez le constater dans la table commune, étant donné le nombre d'entrées de la table, cette façon de consulter les données n'est pas facilement exploitable.
- L'onglet "**Exécuter le SQL**" sert à rédiger des commandes en langage SQL et à les exécuter (en cliquant sur la flèche "lecture" ou en tapant F5)

Voici les tables et attributs de la base :

commune
num_departement
nom
canton
population_2010
population_1999
surface
longitude
zmin
zmax

region
num_region
nom

departement
num_departement
num_region
nom

Le langage SQL n'est pas sensible à la casse (majuscules et minuscules), et à l'indentation (tout le contraire de Python!). Cependant, de façon à ce que le code soit facilement lisible, il est d'usage de taper **en majuscule les mots-clé du langage SQL** et de passer régulièrement à la ligne.

Testez la commande suivante (exécutez-la avec F5) :

```
SELECT nom, num_departement
FROM commune
```

Testez maintenant

```
SELECT *  
FROM region
```

Quelle est la signification du symbole *?

Testez

```
SELECT nom, num_departement  
FROM commune  
ORDER BY population_2010
```

puis

```
SELECT nom, num_departement  
FROM commune  
ORDER BY population_2010 DESC
```

À quoi sert la commande ORDER BY? Qu'est-ce qui change avec la commande DESC?

Testez

```
SELECT nom Nom, canton Nom_Canton  
FROM commune
```

À quoi sert un deuxième paramètre derrière un nom d'attribut ?

Testez

```
SELECT nom, population_2010  
FROM commune  
WHERE population_2010 >100000
```

puis

```
SELECT nom, population_2010  
FROM commune  
WHERE population_2010 >100000  
AND population_2010 < population_1999
```

En déduire à quoi sert la commande WHERE ... éventuellement suivie de un ou plusieurs AND ...

Variantes :

WHERE ... AND NOT ! "intersection du complémentaire"

WHERE ... OR ! "réunion"

Exercice 1

Écrire des requêtes SQL permettant de répondre aux questions suivantes :

1. Quelles sont les villes du Finistère de plus de 10000 habitants ?
2. Est-ce qu'il existe des villes du Finistère qui ont perdu plus de 1000 habitants entre 1999 et 2010?
3. Que fait la commande suivante ?

```
SELECT DISTINCT nom  
FROM commune
```

Est-ce qu'il existe des villes françaises qui portent le même nom?

4. Faites s'afficher la liste des numéros de départements (distincts) ayant au moins une ville de plus de 100 000 habitants.

3 Fonctions arithmétiques et commande GROUP BY

3.1 Fonctions arithmétiques

Le langage SQL dispose de fonctions arithmétiques dont les plus utiles sont :

SUM	renvoie la somme des valeurs
AVG	renvoie la valeur moyenne (average)
MIN, MAX	renvoient respectivement les valeurs minimales et maximales
COUNT	renvoie le nombre de résultats

Par exemple, pour obtenir la somme des populations de toutes les villes, on exécute

```
SELECT SUM(population_2010)
FROM commune
```

Exercice 2

1. Quelle est la population moyenne des villes du Finistère en 2010 ?
2. Combien y a-t-il de villes de plus de 10000 habitants en Bretagne ? Écrire une requête qui vous renvoie ce nombre.

3.2 Commande GROUP BY

Important : lorsqu'elles sont suivies de la commande

```
GROUP BY nom_de_colonne
```

les fonctions précédentes sont exécutées séparément sur les groupes d'entrées pour lesquelles on a égalité de nom_de_colonne.

Par exemple, pour obtenir une table à deux colonnes dont la première est le numéro de département et la deuxième la population totale du département correspondant, on peut écrire

```
SELECT num_departement, SUM(population_2010)
FROM commune
GROUP BY num_departement
```

3.3 Commande GROUP BY...HAVING

Après avoir entré une commande GROUP BY, on peut imposer des conditions supplémentaires avec la commande HAVING. Par exemple, pour sélectionner les départements dont la population moyenne des villes en 2010 est supérieure à 8000, on peut écrire

```
SELECT num_departement, AVG(population_2010)
FROM commune
GROUP BY num_departement
HAVING AVG(population_2010) > 8000
```

La commande HAVING se comporte comme la commande WHERE, mais doit obligatoirement être placée après une commande GROUP BY. En particulier, on peut y ajouter des conditions AND, AND NOT et OR.

Exercice 3

Exécutez des commandes SQL pour obtenir :

1. La liste des noms des villes de France avec pour chaque nom, le nombre de villes qui portent ce nom. La liste sera triée par ordre décroissant du nombre d'occurrences des noms de ville. Quelle est le nom de ville le plus utilisé en France et combien de villes portent ce nom?
2. La liste des numéros de département avec, pour chaque département, le nom de la ville ayant le plus d'habitants et ce nombre d'habitants.

4 Produit cartésien, jointures

4.1 Produit cartésien

Testez la commande suivante :

```
SELECT *  
FROM departement, region
```

Cette commande rajoute l'ensemble des lignes de `region` à la fin de chaque ligne de `departement`, ce qui n'a que peu d'intérêt. On voudrait plutôt ne rajouter à chaque ligne de `departement` la ligne de la région correspondante, ce qui s'obtient en écrivant

```
SELECT *  
FROM departement, region  
WHERE departement.num_region = region.num_region
```

Notez que les tables `departement` et `region` ont toutes les deux des attributs `num_region`. Pour lever l'ambiguïté, on utilise la syntaxe

```
nom_de_la_table.nom_de_l'attribut
```

Comme il arrive très souvent qu'on souhaite "croiser" ainsi les informations de plusieurs tables, il existe une syntaxe spécifique à cette opération en SQL : la commande de jointure.

4.2 Commande de jointure JOIN...ON

Testez

```
SELECT *  
FROM departement  
JOIN region ON departement.num_region = region.num_region
```

Il est important d'avoir bien compris le fonctionnement de la commande `JOIN...ON` sur cet exemple : concrètement, on demande de rajouter à la fin de chaque entrée de la table `departement` l'entrée de la table `region` correspondante.

L'utilisation d'alias ("raccourcis") permet de simplifier un peu les écritures :

```
SELECT *  
FROM departement D  
JOIN region R ON D.num_region = R.num_region
```

On est d'ailleurs parfois obligés d'utiliser des alias, notamment lorsqu'on veut faire une jointure d'une table sur elle-même : dans ce cas, on prend deux alias différents pour chacune des deux tables.

Par exemple, si on souhaite connaître les couples de villes (de noms distincts) qui ont une même longitude, on peut exécuter

```
SELECT C1.nom, C2.nom  
FROM commune C1  
JOIN commune C2 ON C1.longitude=C2.longitude  
WHERE C1.nom != C2.nom
```

Exercice 4

1. Écrire une commande qui renvoie la liste des communes de France avec le nom du département auquel elles appartiennent. Faites ensuite en sorte que la première colonne s'appelle "Commune" et la deuxième "Département".
2. Même question que la précédente, mais on souhaite aussi que figure sur l'entrée de chaque commune le nom de la région à laquelle elle appartient. Ordonnez le tout par ordre alphabétique sur les régions.
3. Écrire une commande qui renvoie la liste des régions avec leur population totale.

4. Écrire une commande qui renvoie la liste des régions dont la population totale a décré strictement entre 1999 et 2010.

5. Pour chacune des commandes suivantes, décrire ce qu'elle renvoie par une phrase.

Vérifier **ensuite** votre prédiction sur machine.

```
SELECT C.nom AS Commune, D.nom AS Département
FROM commune C
JOIN departement D ON C.num_departement = D.num_departement
WHERE population_2010 > population_1999
AND population_2010 > 100000
```

```
SELECT SUM(C.population_2010), D.nom
FROM commune C
JOIN departement D ON C.num_departement = D.num_departement
GROUP BY D.nom
HAVING SUM(C.population_2010) > 1500000
```

```
SELECT R.nom, SUM(C.population_2010)
FROM commune C
JOIN departement D ON C.num_departement = D.num_departement
JOIN region R ON R.num_region=D.num_region
GROUP BY R.nom
```

5 Requêtes composées

Il est possible (et souvent utile) de composer les requêtes comme on compose les fonctions en mathématiques : on effectue une requête sur le résultat d'une requête précédente.

Par exemple, pour trouver les villes plus peuplées que la ville la plus peuplée du Finistère, on peut écrire

```
SELECT nom, population_2010
FROM commune
WHERE population_2010 >=
  (SELECT MAX(population_2010)
   FROM commune
   WHERE num_departement='29')
```

Il existe également une commande WHERE...IN, qui teste l'appartenance à une table. Par exemple, pour connaître les villes qui portent le même nom qu'au moins une ville du Finistère, on peut écrire

```
SELECT *
FROM commune
WHERE nom IN
  (SELECT nom
   FROM commune
   WHERE num_departement="29")
```

Exercice 5

Pour les questions suivantes, il vous est conseillé de mettre sur papier votre "stratégie" pour obtenir le résultat avant de commencer à coder : commencez par envisager la sousrequête.

1. Écrire une requête qui renvoie la liste des villes dont la population est supérieure de la moyenne des populations des villes françaises en 2010.
2. Écrire une requête qui renvoie la population de la région la plus peuplée de France, son numéro et son nom.

Questions supplémentaires

1. Écrire une requête qui renvoie les noms (distincts) des villes de France, classés par ordre alphabétique.
2. Écrire une requête qui renvoie le nombre de villes de plus de 100000 habitants en France en 2010.
3. (a) Écrire une requête qui affiche la liste des villes de population maximale pour chacun des départements, ainsi que le numéro du département et la population correspondante (en 2010).
(b) Modifier la requête précédente pour afficher, à la place de leurs numéros, les noms des départements.
(c) Modifier à nouveau la requête précédente pour afficher également les noms des régions correspondants aux villes, et classer les villes par ordre décroissant de population.
4. (a) Afficher (sans redondance) les numéros des départements dans lesquels il existe une ville de plus de 200000 habitants.
(b) Modifier la requête précédente pour afficher, à la place des numéros de départements, leurs noms.
5. Écrire une requête qui renvoie les noms des villes qui ont une superficie supérieure ou égale à celle de Quimper, ordonnés par ordre alphabétique.