

fiche 5  

# Langage SQL

Version avec preuves.

Pour ce TD nous utiliserons la base de données relationnelle **mondial** (gérée par l'université de Göttingen) . Il est possible d'interagir avec elle à l'adresse :

<http://www.semwebtech.org/sqlfrontend/>

Exercice 1: La base de données contient la table **country**.

Lancer les requêtes pour obtenir les tableaux suivants :

1. L'ensemble des données de la table **country**.

**Correction**

```
select * from country
```

2. La liste des noms des pays.

**Correction**

```
select country.name from country
```

3. La liste des pays ayant la capitale du même nom.

**Correction**

```
SELECT country.name FROM country  
where country.name = country.capital
```

4. La liste des pays,leur superficie et la population des pays ayant plus d'un million d'habitant sur une superficie inférieur à 1000  $km^2$ .

**Correction**

```
SELECT country.name , country.area , country.population  
FROM country  
where area < 1000 and population > 1000000
```

5. La liste des noms des pays terminant par la lettre a. ( on utilisera la commande **like** : <https://sql.sh/cours/where/like>)

**Correction**

```
select country.name from country  
where country.name like '%a'
```

6. La liste des pays dont la densité de population dépasse mille habitants au  $km^2$ .

**Correction**

```
select country.name from country
where country.population > 10000000
```

7. La liste des pays et de leurs populations respectives, triée par ordre décroissant de population. (on utilisera la commande `order by` : <https://sql.sh/cours/order-by>)

**Correction**

```
select country.name , country.population from country
order by country.population desc
```

8. La liste des pays et leur population dont la population dépasse les dix million d'habitants et triée par ordre croissant de population.

**Correction**

```
select country.name , country.population from country
where country.population > 10000000
order by country.population asc
```

Exercice 2: Dans la base de données se trouve une table nommée `encompasses` et une table `continent`. (un pays appartient à un ou plusieurs continents)

Lancer les requêtes pour obtenir les tableaux suivants :

1. La liste des pays dont une fraction au moins est en Europe.

**Correction**

```
SELECT country.name
FROM country JOIN encompasses
ON country.code = encompasses.country
WHERE encompasses.continent = 'Europe'
```

2. La liste des pays du continent américain (Amérique du nord) qui comptent moins de 10 habitants par  $km^2$ .

**Correction**

```
SELECT country.name
FROM country
JOIN encompasses
ON country.code = encompasses.country
WHERE encompasses.continent = 'North_America' and country.population /
country.area < 10
```

3. La liste regroupant le nombre de pays de chaque continent.

**Correction**

```
SELECT encompasses.continent, count(*)  
FROM country  
JOIN encompasses  
ON country.code = encompasses.country  
group by encompasses.continent
```

4. La liste des continents et le nombre d'habitants dont la population totale dépasse le milliard d'habitants.

**Correction**

```
SELECT encompasses.continent , SUM( country.population )  
FROM country JOIN encompasses ON country.code = encompasses.country  
GROUP BY encompasses.continent  
HAVING SUM( country.population ) > 1000000000
```

Exercice 3 :

Dans la base de données se trouve une table nommée `language`.

Lancer les requêtes pour obtenir les tableaux suivants :

1. La liste des langages parlés dans le monde, rangée dans l'ordre croissante.

**Correction**

```
SELECT encompasses.continent, count(*)
FROM country
JOIN encompasses
ON country.code = encompasses.country
group by encompasses.continent
```

2. La liste dans l'ordre croissant des langages ainsi que le nombre de pays les utilisant.

**Correction**

```
SELECT language.name , count(language.name)
FROM language
group by language.name
order by language.name
```

3. La liste des langages dans l'ordre croissant parlés en Europe.

**Correction**

```
SELECT distinct language.name
FROM country JOIN encompasses
ON country.code = encompasses.country
JOIN language
ON country.code = language.country
WHERE encompasses.continent = 'Europe'
order by language.name
```

4. La liste des pays, leur nom et le continent, qui ont comme langage le français.

**Correction**

```
SELECT country.name , encompasses.continent
FROM country JOIN encompasses
ON country.code = encompasses.country
JOIN language
ON country.code = language.country
WHERE language.name = 'French'
```

5. La somme totale des personnes parlant français.

**Correction**

```
SELECT sum( floor( country.population * language.percentage /100 )) as nb
FROM country JOIN encompasses
ON country.code = encompasses.country
JOIN language
ON country.code = language.country
WHERE language.name = 'French'
```

Exercice 4: Dans la base de données se trouve aussi une table nommée `economy`. Lancer les requêtes pour obtenir les tableaux suivants :

1. La répartition économique des pays européens.

#### Correction

```
select *
from economy join encompasses
on economy.country = encompasses.country
where encompasses.continent = 'Europe'
```

2. La liste des pays, et la part de l'agriculture rangé dans l'ordre décroissant ( en retirant les pays dont on ne dispose pas de données).

#### Correction

```
select country.name , economy.agriculture
from economy join country
on economy.country = country.code
where (economy.agriculture > 0)
order by economy.agriculture desc
```

3. La répartition moyenne de l'économie sur les différents continents.

#### Correction

```
select encompasses.continent , floor(avg(economy.agriculture)) , floor(avg(
economy.service))
from economy join encompasses
on economy.country = encompasses.country
group by encompasses.continent
```

4. Déterminer les pays majoritairement agricoles dont le taux de chômage est inférieur à la moyenne mondiale.

#### Correction

```
SELECT country.name
FROM country JOIN economy ON
country.code = economy.country
WHERE
economy.agriculture > economy.service
AND economy.agriculture > economy.industry
AND economy.unemployment < (SELECT AVG( unemployment )FROM economy )
```

5. Déterminer pour chaque continent le pays au taux d'inflation le plus faible parmi les pays majoritairement industriels.

**Correction**

```
SELECT en.continent , c.name
FROM country c JOIN economy e
ON c.code = e.country
JOIN encompasses en ON en.country = c.code
WHERE e.industry > e.agriculture
AND e.industry > e.service
AND( en.continent , e.inflation )in
(SELECT en.continent ,MIN( e.inflation)
FROM economy e JOIN encompasses en
ON e.country = en.country
WHERE e.industry > e.agriculture
AND e.industry > e.service
GROUP BY en.continent )
```