Correction des exercices SQL

Exercice. Exercice 1 du cours

1. la liste des pays dont la population excède 60 000 000 d'habitants ;
2. la même liste triée par ordre alphabétique ;
3. la liste des pays et de leurs populations respectives, triée par ordre décroissant de population ;
4. le nom des dix pays ayant la plus petite superficie ;
5. le nom des dix suivants.

Correction.

Q1

```sql
SELECT name
FROM country
WHERE population >= 60000000
```

Q2

```sql
SELECT name
FROM country
WHERE population >= 60000000
ORDER BY name ASC
```

Q3

```sql
SELECT name, population
FROM country
ORDER BY population DESC
```

Q4

```
SELECT name
FROM country
ORDER BY area ASC
FETCH FIRST 10 ROWS ONLY
```

Q5

```
SELECT name
FROM country
ORDER BY area ASC
OFFSET 10 ROWS
FETCH FIRST 10 ROWS ONLY
```

Exercice. Exercice 1 du TD

1. la liste des continents et de leurs superficies, triés par superficie croissante ;
2. la liste des capitales mondiales, triées par ordre alphabétique ;
3. le nom et la ville de fondation des quatre plus anciennes organisations mondiales ;
4. le nom des volcans de plus de 6 000 mètres ;
5. les dix villes les plus peuplées au monde.

Correction.

Q1

```
SELECT *
FROM continent
ORDER BY area ASC
```

Q2

```
SELECT capital
FROM country
ORDER BY capital ASC
```

Q3

```
SELECT name, city
FROM organization
ORDER BY established ASC
FETCH FIRST 4 ROWS ONLY
```

Q4

```
SELECT name
FROM mountain
WHERE (type='volcanic' OR type='volcano') AND elevation >= 6000
```

Q5

```
SELECT name
FROM city
WHERE population IS NOT NULL
ORDER BY population DESC
FETCH FIRST 10 ROWS ONLY
```

Exercice. Exercice 2 du cours

1. le nom des pays qui sont à cheval sur plusieurs continents ;
2. les pays du contient américain qui comptent moins de 10 habitants par km2 .
3. Déterminer les capitales européennes situées à une latitude supérieure à 60 degrés.

Correction.

Q1

```
SELECT DISTINCT c.name
FROM country c
JOIN encompasses e
ON c.code = e.country
WHERE e.percentage < 100
```

Q2

```sql
SELECT c.name
FROM country c
JOIN encompasses e
ON c.code = e.country
WHERE e.continent = 'North America' AND c.population/c.area <= 10
```

Q3

```sql
SELECT c.capital
FROM country c
JOIN encompasses e
ON c.code = e.country
JOIN city ci
ON c.capital = ci.name AND c.code = ci.country
WHERE e.continent = 'Europe' AND ci.latitude >= 60
```

Exercice. Exercice 2 du TD

1. le nom des pays membres des Nations Unies, triés par ordre alphabétique ;
2. les monarchies constitutionnelles ;
3. les fleuves de France (c'est-à-dire les rivières de France qui se jettent dans la mer) ;
4. le nom des montagnes d'Alaska de plus de 5 000 mètres ;
5. les trois plus hautes montagnes africaines.

Correction.

Q1

```sql
SELECT c.name
FROM country c
JOIN ismember i
ON c.code = i.country
WHERE i.organization = 'UN' AND type='member'
ORDER BY c.name ASC
```

Q2

```sql
SELECT c.name
FROM country c
JOIN politics p
ON c.code = p.country
WHERE p.government = 'constitutional monarchy'
```

Q3

```sql
SELECT DISTINCT r.name
FROM river r
JOIN geo_river gr
ON r.name = gr.river
WHERE gr.country = 'F' AND r.sea IS NOT NULL
```

Q4

```sql
SELECT m.name
FROM mountain m
JOIN geo_mountain gm
ON m.name = gm.mountain
WHERE gm.province = 'Alaska' AND m.elevation >= 5000
```

Q5

```sql
SELECT DISTINCT m.name
FROM mountain m
JOIN geo_mountain gm
ON m.name = gm.mountain
JOIN encompasses e
ON gm.country = e.country
WHERE e.continent = 'Africa'
ORDER BY m.elevation DESC
FETCH FIRST 3 ROWS ONLY
```

Exercice. Exercice 3 du cours

1. Donner la liste ordonnée des dix langues parlées dans le plus de pays différents.
2. Quelles sont les langues parlées dans exactement six pays ? Et de quels pays s'agit-t-il ?
3. Quelles sont les langues parlées par moins de 30 000 personnes dans le monde ?
4. Quelles sont les cinq langues les plus parlées en Afrique ? On précisera pour chacune d'elles le nombre de personnes qui la parlent.

Correction.

Q1

```
SELECT language
FROM spoken
GROUP BY language
ORDER BY COUNT(*) DESC
FETCH FIRST 10 ROWS ONLY
```

Q2

```
SELECT language
FROM spoken
GROUP BY language
HAVING COUNT(*) = 6
```

Pour celui-ci, on doit passer par une sous-requète ... qu'on verra dans les exercices 4 !

Q2 - 2ème partie

```
SELECT c.name, s.language
FROM country c
JOIN spoken s
ON c.code = s.country
JOIN
    (SELECT language, COUNT(*)
    FROM spoken
    GROUP BY language
    HAVING COUNT(*) = 6) six
ON six.language = s.language
ORDER BY s.language ASC
```

Q3

```sql
SELECT s.language
FROM spoken s
JOIN country c
ON s.country = c.code
GROUP BY s.language
HAVING SUM(c.population*s.percentage/100) < 30000
```

Q4

```sql
SELECT s.language, SUM(c.population*s.percentage/100)
FROM spoken s
JOIN encompasses e
ON s.country = e.country
JOIN country c
ON c.code = e.country
WHERE e.continent = 'Africa'
GROUP BY s.language
HAVING SUM(c.population*s.percentage/100) IS NOT NULL
ORDER BY SUM(c.population*s.percentage/100) DESC
FETCH FIRST 5 ROWS ONLY
```

Exercice. Exercice 3 du TD

1. la longueur moyenne des fleuves qui se jettent dans la Mer Noire ;
2. le nombre de rivières françaises présentes dans la base de données ;
3. le nombre total de pays traversés par chacun des fleuves se jetant dans la mer Méditerranée ;
4. les organisations regroupant plus de 100 pays, avec la population totale de ceux-ci ;
5. la liste des pays des Amériques avec leur plus haute montagne.

Correction.

Q1

```sql
SELECT AVG(length)
FROM river
WHERE sea = 'Black Sea'
GROUP BY sea
```

Q2

```
SELECT COUNT(DISTINCT river)
FROM geo_river
WHERE country = 'F'
GROUP BY country
```

Q3

```
SELECT COUNT(DISTINCT r.name)
FROM river r
JOIN geo_river gr
ON gr.river = r.name
WHERE r.sea = 'Mediterranean Sea'
GROUP BY r.sea
```

Q4

```
SELECT o.name, SUM(c.population)
FROM organization o
JOIN ismember im
ON o.abbreviation = im.organization
JOIN country c
ON c.code = im.country
GROUP BY o.name
HAVING COUNT(im.country) >= 100
```

Q5

```
SELECT c.name, MAX(m.elevation)
FROM geo_mountain gm JOIN mountain m ON gm.mountain = m.name
JOIN country c ON gm.country = c.code
JOIN encompasses e ON c.code = e.country
WHERE e.continent = 'North America' OR e.continent = 'South America'
GROUP BY c.name
```

Avec le nom des montagnes :

Q5

```sql
SELECT mmax.nom, m.name, mmax.maxel
FROM mountain m
JOIN
    (SELECT c.name nom, MAX(m.elevation) maxel
     FROM geo_mountain gm JOIN mountain m ON gm.mountain = m.name
     JOIN country c ON gm.country = c.code
     JOIN encompasses e ON c.code = e.country
     WHERE e.continent = 'North America' OR e.continent = 'South America'
     GROUP BY c.name) mmax
ON m.elevation = mmax.maxel
```

Exercice. Exercice 4 du cours

1. Déterminer les pays majoritairement agricoles dont le taux de chômage est inférieur à la moyenne mondiale.
2. Déterminer pour chaque continent le pays au taux d'inflation le plus faible parmi les pays majoritairement industriels.

Correction.

Q1

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

Q2

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

Exercice. Exercice 4 du TD

1. le nom des pays non membres des Nations Unies ;
2. les affluents directs du Nil ainsi que les affluents de ses affluents ;
3. les pays frontaliers de la France, triés par population décroissante ;
4. la longueur de la frontière française ;
5. le nombre de voisins de chaque pays d'Europe ;
6. les pays qui partagent une même montagne.

Correction.

Q1

```
SELECT name
FROM country

EXCEPT

SELECT c.name
FROM ismember im
JOIN country c
ON c.code = im.country
WHERE im.organization = 'UN'
```

Q2

```
SELECT name
FROM river
WHERE river = 'Nile'

UNION

SELECT name
FROM river
WHERE river IN (SELECT name
                FROM river
                WHERE river = 'Nile')
```

Q3

```
SELECT name, population
FROM (SELECT c.name, c.population
      FROM country c
      JOIN borders b
      ON c.code = b.country1
      WHERE b.country2 = 'F'

      UNION

      SELECT c.name, c.population
      FROM country c JOIN borders b ON c.code = b.country2
      WHERE b.country1 = 'F')
ORDER BY population DESC
```

Q4

```sql
SELECT SUM(length)
FROM (SELECT b.length
      FROM country c
      JOIN borders b
      ON c.code = b.country1
      WHERE b.country2 = 'F'

     UNION

     SELECT b.length
     FROM country c
     JOIN borders b
     ON c.code = b.country2
     WHERE b.country1 = 'F')
```

Q5

```sql
SELECT name, SUM(nb_voisins)
FROM (SELECT c.name, COUNT(*) nb_voisins
     FROM country c
     JOIN encompasses e
     ON c.code = e.country
     JOIN borders b
     ON c.code = b.country1
     WHERE e.continent = 'Europe'
     GROUP BY c.name

     UNION

     SELECT c.name, COUNT(*) AS nb_voisins
     FROM country c
     JOIN encompasses e
     ON c.code = e.country
     JOIN borders b
     ON c.code = b.country2
     WHERE e.continent = 'Europe'
     GROUP BY c.name)
GROUP BY name
```

Q6 - réponse attendue

```
SELECT DISTINCT c.name
FROM mountain m
JOIN geo_mountain gm1
ON m.name = gm.mountain
JOIN country c
ON c.code = gm.country
WHERE m.name IN
                (SELECT mp.name
                FROM mountain mp
                JOIN geo_mountain gmp
                ON mp.name = gmp.mountain
                WHERE gm.country <> gmp.country)
```

Q6 - si on veut voir quelles sont les montagnes en question

```
SELECT DISTINCT m.name, c.name
FROM mountain m
JOIN geo_mountain gm1
ON m.name = gm.mountain
JOIN country c
ON c.code = gm.country
WHERE m.name IN
                (SELECT mp.name
                FROM mountain mp
                JOIN geo_mountain gmp
                ON mp.name = gmp.mountain
                WHERE gm.country <> gmp.country)
ORDER BY m.name ASC
```