

TP 4 : S.Q.L. solutions des R1 à R10

- (R1) Afficher le nom et la longueur de toutes les rivières, classées par longueur décroissante (mot clef DESC pour décroissante.).

Solution 1 `SELECT Name, Length FROM River ORDER BY Length DESC`

- (R2) Afficher le nom de toutes les montagnes dont l'altitude est supérieure ou égale à 8000 m.

Solution 2 `SELECT Name FROM Mountain WHERE Elevation>=8000`

(14 réponses).

- (R3) Afficher les codes Country1, Country2 de la table borders où l'un des deux pays a le code 'F' de la France.

Solution 3 `SELECT Country1, Country2 FROM borders WHERE Country1='F' OR Country2='F'`

(8 réponses).

- (R4) A l'aide de l'opérateur SUM déterminer la longueur totale des frontières terrestres françaises.

Solution 4 `SELECT SUM(length)
FROM borders
WHERE country1 = 'F' OR country2 = 'F'`

(On trouve 2892.4 km).

- (R5) Combien d'habitants compte la capitale du Lesotho ?

Consigne : *On demande de trouver le résultat par deux méthodes différentes. La première en faisant une jointure entre les deux tables¹ qui contiennent les informations nécessaires, la seconde avec une sous-requête.*

Solution 5 (M1) La jointure :

```
SELECT ci.Population  
FROM City ci  
JOIN Country c ON c.Capital = ci.Name  
WHERE c.Name = 'Lesotho'
```

(M2) Avec une sous-requête :

```
SELECT Population FROM City  
WHERE Name = (SELECT Capital FROM Country WHERE Name='Lesotho')}
```

(Réponse : 75000 habitants).

- (R6) Lister les noms et la population des pays dont la population est supérieure à celle de la France. (19 pays).

Solution 6 `SELECT Name, Population
FROM Country
WHERE Population > (SELECT population FROM country WHERE name='France')`

- (R7) Afficher le nom de la plus haute montagne du monde.

Solution 7
`SELECT Name
FROM Mountain
WHERE Elevation = (SELECT MAX(Elevation) FROM Mountain)`

1. Votre première jointure ah quelle émotion, en plus la question est passionnante.

- (R8) Afficher les *noms* de toutes les montagnes dont l'altitude est comprise strictement entre 3000 et 4000m, avec le code de Pays où elles se trouvent.

N.B. les code de Pays et les altitudes ne se trouvent pas dans la même table.

Solution 8 On doit utiliser les tables `Mountain` et `geo_Mountain`. La jointure se fait par le nom de la montagne, qui s'appelle `Name` dans `Mountain` mais `Mountain` dans `geo_Mountain`.

```
SELECT M.Name, G.Country FROM Mountain M, geo_Mountain G
WHERE M.Name=G.Mountain AND M.Elevation >3000 AND M.Elevation<4000
```

(48 réponses sans `SELECT DISTINCT`, mais 41 avec `SELECT DISTINCT`. Cette différence vient du fait que la même montagne du même pays apparaît dans plusieurs provinces.

- (R9) Modifier la requête précédente pour ne pas faire apparaître les code de pays mais les vrais noms de pays.

Première jointure entre trois tables !

Solution 9 Cette fois besoin d'une jointure entre trois tables, puisque le lien entre code de pays et nom de pays se trouve dans `Country`.

```
SELECT M.Name, C.Name FROM geo_Mountain G
JOIN Mountain M ON M.Name=G.Mountain
JOIN Country C ON C.Code=G.Country
WHERE M.Elevation >3000 AND M.Elevation<4000
```

- (R10) Afficher le nom des provinces suisses qui n'ont pas de lacs.

Solution 10 `SELECT Name as Province FROM Province WHERE country = 'CH' EXCEPT SELECT Province FROM geo_lake WHERE Country = 'CH'`