

NOM : PRENOM : n° ETU :

La base de données utilisée contient 2 tables, `city` et `country`.

country			city		
Column	Type		Column	Type	
countrycode	character(3)		id	integer	
name_country	text		name_city	text	
continent	text		countrycode	character(3)	
region	text		district	text	
population_country	integer		population_city	integer	
lifeexpectancy	real				
capital	integer				
governmentform	text				

La base de données satisfait également les contraintes de clefs étrangères suivantes :

- `city.countrycode` \subseteq `country.countrycode`,
- `country.capital` \subseteq `city.id`.

1. Écrire en SQL une requête dont le résultat contient l'espérance de vie moyenne par continent, par ordre alphabétique sur `continent` (tableau résultat : `continent`, `esp_vie_moyenne`).

```
SELECT continent, avg(lifeexpectancy) as esp_vie_moyenne
FROM country
GROUP BY continent
ORDER BY continent
```

2. Écrire en SQL une requête dont le résultat contient le nom des pays dont la population (`population_country`) est strictement supérieure à la somme de la population des villes (`population_city`) .

```
SELECT C1.name_country
FROM Country C1
WHERE C1.population_country > (SELECT SUM(C2.population_city)
                               FROM City C2
                               WHERE C1.countrycode=C2.countrycode);
```

ATTENTION, la requête suivante (vu à maintes reprises sur les copies) est incorrecte :

```
incorrect **** SELECT name_country **** incorrect
incorrect **** FROM Country NATURAL JOIN City **** incorrect
incorrect **** HAVING population_country > SUM(population_city); **** incorrect
```

Il ne faut pas confondre le niveau des tuples et celui des groupes. L'attribut `population_country` ne peut pas aller dans le `HAVING`, car il ne figure ni dans le `GROUP BY` ni dans la portée d'un agrégat. Il concerne donc les tuples et non les groupes.

Veillez par ailleurs à toujours faire suivre un `HAVING` ou un `WHERE` par une expression Booléenne. Ces mots clefs servent en effet à construire des conditions Booléennes filtrant respectivement les groupes et les tuples. La condition `HAVING MAX(attribut)` n'a par exemple aucun sens telle quelle. En revanche, la condition `HAVING MAX(attribut)=valeur` est correctement définie. L'expression `valeur` doit être une expression du même type que `attribut`, par exemple une constante ou une sous requête scalaire, i.e., une sous requête renvoyant un unique tuple sur un unique attribut (donc une table contenant une unique valeur).

3. Écrire en SQL une requête dont le résultat contient pour chaque pays (`countrycode`), le district qui contient le plus de villes.

```
WITH dis_city as (Select countrycode, district, count(*) as nb
                  FROM City
                  GROUP BY countrycode, district)
SELECT D1.countrycode, D1.district
FROM   dis_city D1
GROUP BY D1.countrycode, D1.district
HAVING count(*) = (Select max(D2.nb)
                  FROM dis_city D2
                  WHERE D1.countrycode=D2.countrycode
                  AND D1.countrycode=D2.countycode);
```

ou

```
Select C1.countrycode, C1.district, count(*) as nb
FROM City C1
GROUP BY C1.countrycode, C1.district
HAVING count(*) >= ALL (SELECT count(*)
                       FROM City C2
                       WHERE C1.countrycode=C2.countrycode
                       GROUP BY C2.countrycode, C2.district);
```

4. Écrire en SQL une requête dont le résultat contient les régions dans lesquelles tous les pays ont la même forme de gouvernement. Donnez deux versions : une utilisant l'agrégation et l'autre pas.

```
SELECT C1.region
FROM Country C1
WHERE NOT EXISTS (SELECT *
                  FROM Country C2
                  WHERE C1.region=C2.region
                  AND C1.governmentform<>C2.governmentform);
```

ou

```
SELECT C1.region
FROM Country C1
WHERE C1.region NOT IN (SELECT C2.region
                       FROM Country C2
                       WHERE C1.region=C2.region
                       AND C1.governmentform<>C2.governmentform);
```

ou

```
SELECT C1.region
FROM Country C1
WHERE C1.governmentform = ALL (SELECT C2.governmentform
                                FROM Country C2
                                WHERE C1.region=C2.region);
```

ou en utilisant l'agrégation :

```
SELECT region
FROM Country
GROUP BY region
HAVING COUNT(DISTINCT governmentform)=1;
```