

Fonctions d'agrégation

1 Count

Les fonctions d'agrégation permettent d'effectuer des opérations avancées sur les solutions d'une requête comme : compter, sélectionner le maximum etc. Une des opérations les plus courantes est de compter. `COUNT(col)` permet de compter les résultats d'une requête. Pour compter les pays en Europe, on écrira :

```
SELECT COUNT(countrycode)
FROM world.country
WHERE continent='Europe';
```

Cette requête renvoie une table ayant une ligne et une colonne contenant le nombre de réponse à la requête. Admettons qu'on veuille compter les pays par continent. On serait tenté d'écrire :

```
SELECT continent ,COUNT(countrycode)
FROM world.country;
```

Cependant, cette requête ne fonctionnera pas en SQL. SQL est incapable de deviner comment regrouper les différentes lignes pour compter. On doit lui spécifier clairement cela avec la clause `GROUP BY` :

```
SELECT continent ,COUNT(countrycode)
FROM world.country
GROUP BY continent;
```

Cette requête regroupe les lignes de la table `country` par continent et pour chaque groupe compte le nombre de `countrycode` y apparaissant. Lorsque plusieurs lignes sont susceptibles d'avoir la même valeur, on peut compter seulement le nombre d'occurrence distinctes avec `COUNT(DISTINCT col)`.

1. Écrire une requête qui compte le nombre de langues parlées dans chaque pays.

Solution:

```
SELECT countrycode , COUNT(language)
FROM world.countrylanguage
GROUP BY countrycode;
```

2. Écrire une requête qui compte le nombre de langues parlées dans le monde.

Solution:

```
SELECT COUNT(DISTINCT language)
FROM world.countrylanguage;
```

3. Écrire une requête qui compte le nombre de langues officielles par pays.

Solution: Une solution presque bonne :

```
SELECT countrycode , COUNT(language)
FROM world.countrylanguage
WHERE isofficial;
```

Cependant, on rate les pays qui n'ont pas de langue officielle. On va utiliser une superbe jointure extérieure :

```
SELECT countrycode , COUNT(language)
FROM world.country as P LEFT JOIN world.countrylanguage as L
ON (P.countrycode = L.countrycode and L.isofficial)
```

2 Sum, Max, Min, Avg

Une autre fonction importante est la fonction `SUM(col)` qui effectue la somme des valeurs (numériques) d'une colonne :

```
SELECT SUM(population_country)
FROM world.country;
```

renvoie la population mondiale. On peut de même utiliser `GROUP BY` pour faire des paquets :

```
SELECT continent ,SUM(population_country)
FROM world.country
GROUP BY continent;
```

renvoie la population de chaque continent. On peut même faire des opérations sur la colonne à l'intérieur de `SUM`. Par exemple : `SUM(percentage/100)`.

1. Écrire une requête qui renvoie la surface de chaque région.

Solution:

```
SELECT region , SUM(surfacearea)
FROM world.country
GROUP BY region;
```

2. Écrire une requête qui compte le nombre de francophones dans le monde.

Solution:

```
SELECT SUM((percentage/100)*population_country)
FROM world.country NATURAL JOIN world.countrylanguage
WHERE language = 'French';
```

On peut utiliser de la même façon la fonction `MIN` (resp. `MAX`) qui renvoie la plus petite (resp. grande) valeur ou `AVG` qui renvoie la moyenne.

1. Combien de personnes vivent dans une capitale européenne ?

Solution:

```
SELECT SUM(population_city)
FROM world.country JOIN world.city ON capital=id
WHERE continent='Europe';
```

2. Quel est la capitale européenne la moins peuplée ?

Solution:

```
WITH r as (select id ,name_city ,sum(population_city) as s
FROM world.city JOIN world.country ON capital = id
WHERE continent='Europe' GROUP BY id ,name_city)

SELECT name_city FROM r WHERE s = (SELECT min(s) FROM r);
```

3. Quelle est la langue la plus parlée dans le monde ?

Solution:

```
WITH R AS
(SELECT language ,SUM((percentage/100)*population_country) AS s
FROM world.country NATURAL JOIN world.countrylanguage
GROUP BY language)

SELECT language FROM R WHERE s = (SELECT MAX(s) FROM R);
```

3 Having

Parfois, on veut filtrer les requêtes en fonction du résultat d'une fonction d'agrégation. Par exemple, pour connaître les langues officielles dans plus de 10 pays, on serait tenté d'écrire :

```
SELECT language FROM world.countrylanguage
WHERE COUNT(countrycode) > 10 and isofficial GROUP BY language;
```

Cependant, cela ne fonctionne pas. **WHERE** applique une condition sur chaque ligne de la table pour les filtrer, par exemple, garder seulement les langues officielles. Ici, on veut *ensuite* sélectionner les lignes après avoir regroupé par langue et compté. On utilisera alors **HAVING**, après le **GROUP BY** :

```
SELECT language FROM world.countrylanguage WHERE isofficial
GROUP BY language HAVING COUNT(countrycode) > 10;
```

4 TP

1. Écrire une requête qui renvoie le nombre de pays par régime.

Solution:

```
SELECT governmentform AS regime , COUNT(countrycode) AS nombre
FROM world.country
GROUP BY governmentform
ORDER BY governmentform ;
```

2. Écrire une requête calculant le nombre de personnes vivant dans des villes de plus d'un million d'habitant.

Solution:

```
SELECT SUM(population_city) AS pop
FROM world.city
WHERE population_city >= 1000000 ;
```

3. Écrire une requête qui calcule le nombre total de personnes vivants dans des villes qui ne sont pas listées dans la table `city`. (Indice : comparer la population du pays avec la somme sur les villes).

Solution:

```
WITH pop_villes AS (SELECT SUM(population_city) AS pop
FROM world.city) ,
pop_totale AS (SELECT SUM(population_country) AS pop
FROM world.country)
SELECT pop_totale.pop - pop_villes.pop
```

```
FROM pop_totale , pop_villes ;  
Réponse : (5 503 612 232 non)
```

4. Écrire une requête qui compte le nombre moyen de langue parlée par pays dans chaque région.

Solution: Dans chaque pays, calculer le nombre de langues parlées

```
WITH nb_langues_parlees AS (SELECT countrycode , COUNT(language) nb  
FROM world.countrylanguage  
GROUP BY countrycode )  
SELECT region , SUM(nb)/COUNT(DISTINCT countrycode)  
FROM world.country co NATURAL JOIN nb_langues_parlees  
GROUP BY region ;
```

Avec avg

```
WITH nb_langues_parlees AS (SELECT countrycode , COUNT(language) nb  
FROM world.countrylanguage  
GROUP BY countrycode )  
SELECT region , avg(nb)  
FROM world.country co NATURAL JOIN nb_langues_parlees  
GROUP BY region ;
```

5. Écrire une requête qui donne la liste des pays ayant deux langues officielles parlées par plus de la moitié de la population.

Solution: Pas besoin d'agrégation à cet endroit là ..

Devoir maison

1. Écrire une fonction `city(countryname character varying(49))` qui étant donné le nom d'un pays, renvoie le nom de la ville la plus peuplée de ce pays. (schéma : world)
2. Écrire une fonction `langue_region(c text)` qui étant donné le nom d'un continent, renvoie le nombre moyen de langue parlée par pays dans chaque région. (schéma : world). L'entête de cette fonction doit être :
`FUNCTION langue_region(c TEXT) RETURNS TABLE(region TEXT, nbmoy NUMERIC)`
3. Écrire une fonction `actor_category(nom character varying(49), prenom character varying(49))` qui prend en argument le nom et le prénom d'un acteur (d'une actrice) et renvoie la liste des noms des catégories de films dans lesquels il/elle a joué (schéma : sakila).
4. Écrire une fonction `co_actors(nom character varying(49), prenom character varying(49))` qui renvoie les noms et prénoms des acteurs qui jouent dans un film où apparaît un acteur ou une actrice dont le nom et le prénom sont donnés en argument (schéma : sakila).