# TD 4 : SQL

## Aggrégations, Partitions, Fenêtres

#### 2025-10-17



#### Avec solutions

L3 MIASHS Université Paris Cité Année 2025 Course Homepage Moodle



#### Documentation Postgres

#### Documentation Postgres en Français

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

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

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

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

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

#### Solution

Une solution presque bonne:

```
SELECT countrycode, COUNT(language) AS nb_lg_official
FROM world.countrylanguage
WHERE isofficial
GROUP BY countrycode
ORDER BY nb_lg_official, countrycode;
```

Cependant, on perd les pays qui ne possèdent pas de langue officielle. On va utiliser une jointure extérieure pour les conserver :

```
SELECT c.countrycode, c.name_country, COUNT(cl.language) AS nb_lg_official
FROM world.country c LEFT JOIN world.countrylanguage cl
ON (c.countrycode = cl.countrycode AND cl.isofficial)
GROUP BY c.countrycode, c.name_country
ORDER BY nb_lg_official, c.countrycode;
```

NB: L.isofficial doit être dans la condition de jointure et pas dans un WHEREpour ne pas perdre les pays qui n'ont pas de langue officielle.

4. Écrire une requête qui renvoie le nombre de langues officielles par pays en utilisant la fonction SUM.

#### Solution

Presque correct :

```
WITH s AS(
    SELECT L.countrycode, sum(CAST (isofficial AS INTEGER)) AS nb_lg_official FROM world.countrylanguage as L
    GROUP BY L.countrycode)
SELECT s.*, c.name_country
FROM world.country c JOIN s USING (countrycode)
ORDER BY s.nb_lg_official, s.countrycode;
```

Mais on a perdu les pays qui n'apparaissent pas dans la table countrylanguage. Pour y remédier, il faut faire une jointure gauche et faire un CASE WHEN pour gérer les valeurs NULL. Ici, les valeurs NULL entrent dans le cas par défaut ELSE.

```
SELECT c.countrycode, c.name_country, SUM(CASE WHEN cl.isofficial THEN 1 ELSE 0 END) AS nb_lg_FROM world.country c LEFT JOIN world.countrylanguage cl USING (countrycode) GROUP BY c.countrycode, c.name_country ORDER BY nb_lg_official, c.countrycode;
```

Au final, compter en faisant une somme n'est pas la meilleure idée.

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

#### Solution

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

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

# Solution SELECT ROUND(SUM((percentage/100)\*population\_country)) FROM world.country JOIN world.countrylanguage USING (countrycode) WHERE language = 'French';

7. Combien de personnes vivent dans une capitale européenne?

```
Solution

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

8. Quelle est la capitale européenne la moins peuplée?

```
WITH r AS (
    SELECT min(population) AS min
    FROM world.country co
        JOIN world.city ci ON co.capital=ci.id
    WHERE co.continent='Europe'
)
SELECT ci.name, ci.population
FROM city ci join r ON ci.population = r.min;
```

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

#### Solution

```
WITH r AS
(
    SELECT language, ROUND(SUM((percentage/100)*population_country)) AS sum
    FROM world.country JOIN world.countrylanguage USING (countrycode)
    GROUP BY language
),
s AS
(
    SELECT MAX(sum) AS max
    FROM r
)
SELECT language, r.sum AS nb_speaker
FROM r JOIN s ON r.sum = s.max;
```

On peut aussi utiliser un ordre descendant sur le nombre de locuteurs en faisant attention à bien gérer les valeurs nulles possibles. Car celles-ci fausses l'ordre.

Ici, pas de souci car les valeurs nulles sont interdites pour les attributs country.population\_country et countrylanguage.percentage.

```
SELECT
cl.language, ROUND(SUM((percentage/100)*population_country)) as nb_speaker
FROM
world.country co JOIN world.countrylanguage cl USING(countrycode)
GROUP BY cl.language
ORDER BY nb_speaker DESC
LIMIT 1;
```

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

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

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

```
Solution

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

12. Écrire une requête qui calcule le nombre total de personnes dans le monde qui n'habitent pas une ville listée dans la table  $\mathtt{city}$ . Réponse : (4,649,189,566)

```
Solution
WITH
  pop_villes AS
    SELECT
      SUM(population) AS pop
    FROM
      world.city
  ),
  pop_totale AS
      SUM(population_country) AS pop
    FROM
      world.country
  )
SELECT
  (pop_totale.pop - pop_villes.pop) as non_urban_population
  pop_totale, pop_villes ;
```

13. Écrire une requête qui compte le nombre moyen de langues parlées par pays dans chaque région.

```
Solution
Avec la fonction AVG():
WITH nb_langues_parlees AS (
   SELECT
    countrycode, COUNT(language) nb
   FROM
    world.countrylanguage
   GROUP BY countrycode
SELECT
  co.region, AVG(nb)
FROM
  world.country co JOIN nb_langues_parlees USING(countrycode)
GROUP BY co.region
ORDER BY co.region;
Sinon on faisant le calcul de moyenne "à la main" :
WITH nb_langues_parlees AS (
    SELECT
    countrycode, COUNT(language) nb
    world.countrylanguage
    GROUP BY countrycode
SELECT
  region, SUM(nb)/COUNT(DISTINCT countrycode)
  world.country co JOIN nb_langues_parlees USING(countrycode)
GROUP BY region ;
Le DISTINCT dans COUNT(DISTINCT countrycode) est utile en général pour faire un calcul
correct de moyenne. Cependant, ici, il est inutile car nous avons fait une jointure sur
```

14. Écrire une requête qui donne la liste des pays ayant deux langues officielles parlées chacune par plus du quart de la population.

countrycode qui est à valeur unique dans les deux tables jointes.

♣ Pas besoin d'agrégation à cet endroit là.

```
SELECT
cl1.countrycode, cl1.language, cl2.language
FROM
countrylanguage AS cl1
JOIN
countrylanguage AS cl2 ON (
cl1.countrycode=cl2.countrycode AND
cl1.language < cl2.language AND
cl1.isofficial AND cl2.isofficial
)
WHERE cl1.percentage > 25 AND cl2.percentage > 25;
```

15. Écrire une fonction plus\_peuplee(p\_countrycode text) qui, étant donné le code d'un pays, renvoie le nom de la ville la plus peuplée de ce pays.

### Solution En remplaçant monschema par votre nom de schéma personnel : CREATE OR REPLACE monschema.plus\_peuplee(p\_countrycode text) RETURNS text LANGUAGE sql AS \$\$ WITH m AS ( SELECT MAX(population) AS m\_pop FROM world.city ci WHERE ci.countrycode = p\_countrycode **SELECT** ci.name FROM world.city ci, m WHERE ci.countrycode = p\_countrycode AND ci.population = m\_pop; -- Utilisation de la fonction SELECT monschema.plus\_peuplee('FRA'); SELECT monschema.plus\_peuplee('USA');

16. Écrire une fonction langues\_region(p\_continent text) qui étant donné le nom d'un continent, renvoie le nombre moyen de langues parlées par pays dans chaque région. L'entête de cette fonction doit être :

```
FUNCTION langues_region(p_continent TEXT)
RETURNS TABLE(region TEXT, nbmoy NUMERIC)
```

#### Solution

En remplaçant monschema par votre nom de schéma personnel:

```
CREATE OR REPLACE FUNCTION monschema.langues_region(p_continent TEXT)
RETURNS TABLE(region TEXT, nbmoy NUMERIC)
LANGUAGE SQL AS $$
WITH S AS (
  SELECT
    R.region, cl.countrycode, COUNT(cl.language) AS n_l
  FROM
    (
      SELECT
        co.region, co.countrycode
      FROM
        world.country co
      WHERE
        co.continent = p_continent
    ) AS R
  JOIN
    world.countrylanguage cl USING (countrycode)
  GROUP BY
    R.region, cl.countrycode
)
SELECT
  region, AVG(n_1) AS nbmoy
FROM
  S
GROUP BY
 region;
$$;
SELECT * FROM monschema.langues_region('Europe');
SELECT * FROM monschema.langues_region('Asia');
```

- 17. Écrire une requête qui liste trois attributs, pour chaque pays où on parle français, ordonné par population croissante :
  - name\_country,
  - cumul\_loc qui donne le nombre cumulé de locuteurs du français dans les pays où on parle français, dans l'ordre croissant des populations des pays,
  - cum\_pop qui donne la population cumulée des pays où on parle français, dans l'ordre croissant des populations des pays.
- Utilisez une fonction fenêtre (WINDOW) sans partition.
- Pour trouver les pays où on parle français, utilisez l'expression language like '%French%'. Vous remarquerez que dans certains pays, il existe plusieurs variétés de 'French'. Veillez à compter les locuteurs de toutes les formes du Français. Que se passe-t-il si un locuteur parle plusieurs formes de Français?

#### Solution

```
WITH f AS (
  SELECT
    cl.countrycode, SUM(cl.percentage) AS percentage
    world.countrylanguage cl
  WHERE
    cl.language LIKE '%French%'
   GROUP BY
    cl.countrycode
)
SELECT
  co.name_country,
  SUM(ROUND(f.percentage * co.population_country/100)) OVER w AS cumul_loc,
  SUM(co.population_country) OVER w AS cum_pop
FROM
  f JOIN world.country co USING(countrycode)
WINDOW w AS (
  ORDER BY co.population_country
```

**Problème** : si une même personne parle plusieurs formes de Français, elle est comptabilisée autant de fois comme locuteur du Français. Mais aucune information dans la table ne permet de distinguer ce cas de figure.