

# TD 4 : SQL

## Aggrégations, Partitions, Fenêtres

2025-10-17

---

L3 MIASHS  
[Université Paris Cité](#)  
Année 2025  
[Course Homepage](#)  
[Moodle](#)



[Documentation Postgres](#)  
[Documentation Postgres en Français](#)

## Fonctions d'agrégation

Les fonctions d'*agrégation* permettent d'effectuer des opérations avancées sur les solutions d'une requête (sur une table) comme : compter les lignes, sélectionner le maximum dans une colonne, etc.

Une des opérations les plus courantes est de compter. `COUNT(col)` permet de compter les résultats d'une requête.

### Count

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 lignes dans le résultat de la requête.

## GROUP BY

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 modalité de la colonne `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'occurrences distinctes avec `COUNT(DISTINCT col)`.

**i Question**

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

**i Question**

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

**i Question**

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

Une solution presque bonne :

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

## 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)`.

**i Question**

Écrire une requête qui renvoie le nombre de langues officielles par pays

## Requêtes (I)

**i Question**

Écrire une requête qui renvoie la surface de chaque région.

**i Question**

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

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.

**i Question**

Combien de personnes vivent dans une capitale européenne ?

**i Question**

Quelle est la capitale européenne la moins peuplée ?

**i Question**

Quelle est la langue la plus parlée dans le monde ?

## 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;
```

🔥 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 la clause **GROUP BY** :

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

- ① La requête concerne la table **world.countrylanguage**
- ② On filtre les lignes qui correspondent à des langues officielles
- ③ On groupe/partitionne la table filtrée selon la langue
- ④ On ne garde que les groupes comportant au moins 10 tuples
- ⑤ On projette le résultat sur la colonne **language**

## Requêtes (II)

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

**i Question**

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

**i Question**

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).

Réponse : (4,649,189,566)

**i Question**

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

**i Question**

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

👉 Pas besoin d'agrégation à cet endroit là.

**i Question**

Écrire une fonction `plus_peuplee(p_name_country text)` qui, étant donné le nom d'un pays, renvoie le nom de la ville la plus peuplée de ce pays. (schéma : `world`)

**i Question**

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 (schéma : `world`). L'entête de cette fonction doit être :

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

**i Question**

Ecrire une fonction `actor_category(p_nom text, p_prenom text)` 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 : `pagila`).

**i Question**


Ecrire une fonction `co_actors(p_nom text, p_prenom text)` qui renvoie les noms et pré-noms 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 : `pagila`).

**i** Question

Écrire une vue qui contient une ligne pour chaque pays où on parle français, présente les pays par population croissante, et contient trois colonnes :

- `name_country` (même type que dans `world.country`) ;
- `cumul_loc` (de type `float4`) qui donne le nombre cumulé de locuteurs du français dans les pays où on parle français, pas plus peuplés que le pays courant ;
- `cum_pop` (de type `float4`) qui donne la population cumulée des pays où on parle français, pas plus peuplés que le pays courant.

 Utilisez une fenêtre (`WINDOW`) avec une clause `RANGE` ....

 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 tous les locuteurs, et à ne compter les habitants qu'une seule fois.