

# TD 1 : Algèbre relationnelle

Logistique et Algèbre relationnelle

2025-09-26

---

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



## Introduction

### Paramétrage du client

En suivant le [guide dBeaver](#) :

- Paramétrez la connexion au serveur dans le client dBeaver.
- Utilisez l'onglet **Projet**.
- Créez un sous-répertoire **tp1** dans le répertoire par défaut du projet.
- Ouvrez un nouveau script dans votre projet et renommez-le en **ex1.sql**.

### Schémas

Les schémas sont une abstraction spécifique à PostgreSQL. Les schémas permettent de faire cohabiter dans une même base de données (un “catalogue” dans le jargon PostgreSQL) plusieurs ensembles d'informations de natures différentes.

Dans ce TP, nous allons nous concentrer sur le schéma **world** qui contient des informations concernant des villes et des pays/territoires du monde entier. Au cours de ce semestre, nous serons amenés à utiliser d'autres schémas : **pagila** qui contient des informations concernant des films, ...

Dans dBeaver, vous pouvez lister les schémas du catalogue **bd\_2023-24**, en développant la connexion **bd\_2023-24**.

Puis en développant le schéma **world**, vous pouvez lister les tables qu'il contient.

Vous pouvez consulter la [page de présentation du schéma world](#).

### Résolution des noms de tables et d'attributs

Pour résoudre les noms de tables et d'attributs, le serveur SQL utilise la syntaxe `nom_schéma.nom_table[.nom_attribut]` (rappel : les crochets signifient optionnel).

Par exemple, **world.city** désigne la table **city** du schéma **world**.

Pour alléger cette écriture, vous pouvez utiliser le *chemin par défaut*.

Afficher le chemin par défaut avec la requête SQL : `SHOW search_path ;`

Vous pouvez modifier ce chemin avec la requête `SET search_path TO nom_schéma1 [, nom_schéma2]`. Il peut y avoir plusieurs schémas dans le **search\_path**.

L'ordre est important. Avec `SET search_path TO world, public ;` la résolution des noms se fera d'abord dans le schéma `world` puis en cas d'échec, dans le schéma `public`.

Après cette instruction `city` désigne la table `city` du schéma `world` si elle existe, sinon cela désigne la table `city` du schéma `public`, sinon le serveur renverra une erreur.

On ajoute systématiquement `public` dans le `search_path`, pour permettre une bonne compatibilité car de nombreuses applications et outils s'attendent à ce que les objets (tables, vues, fonctions) soient accessibles via ce schéma.

Exécutez le script

```
SHOW search_path ;  
SET search_path TO world, public ;  
SHOW search_path ;
```

## Tables

Pour voir le schéma d'une table avec le client dBeaver, il suffit de dérouler le schéma `world` puis de double-cliquer sur la table.

Vous pouvez aussi visualiser toutes les tables du schéma par un clic droit sur le schéma puis **Voir le diagramme**.

Dans la deuxième partie du TP, on s'intéresse aux requêtes, c'est-à-dire, les moyens d'extraire une information pertinente d'une base de données.

## Écriture d'une requête

Pour extraire des informations d'une base de données, on utilise l'algèbre relationnelle (pour la théorie) et le langage SQL (pour la pratique).

A la fin du TP, vous devez rendre sur Moodle vos fichiers scripts (a priori un par exercice). Chaque fichier contient une liste de requêtes SQL. *Lorsque c'est demandé, il faut écrire en commentaire l'expression correspondante en algèbre relationnelle.*

Pour écrire un commentaire en SQL : `--` en début de ligne.

*NB* : pensez à documenter vos scripts en utilisant les commentaires SQL

## Requêtes monotables

On rappelle que la requête SQL

```
SELECT colonne1, colonne2, ...  
FROM table  
WHERE condition;
```

affiche les colonnes de `table` indiquées après le `SELECT` qui respectent la `condition`.

C'est la transcription de

$$\Pi_{\text{colonne1, colonne2}}(\sigma_{\text{condition}}(\text{table}))$$

Par exemple :

```
SELECT name_country, lifeexpectancy  
FROM world.country  
WHERE lifeexpectancy < 50 and continent = 'Asia';
```

affichera le nom et l'espérance de vie des pays d'Asie ayant une espérance de vie inférieure à 50 ans.

Remarquez l'utilisation des apostrophes simples pour délimiter les chaînes de caractères ('Asia').

## Exercice 1

Ecrire des requêtes en algèbre relationnelle et en SQL pour extraire les informations suivantes du schéma **world** :

1. Quelles sont les régions ? (25 lignes)
2. Quelles sont les régions situées en Europe ? (6 lignes)
3. Quels sont les pays situés en Europe du sud ('Southern Europe') ? (15 lignes)
4. Quelles sont les capitales des pays situés en Europe de l'Ouest ? (quel est le type de la colonne **capital** dans la table **country** ?) (9 lignes)
5. A partir de la table **countrylanguage**, quels sont les langues qui sont officielles dans au moins un pays ? (102 lignes)
6. Quels sont les codes des pays où le français est langue officielle ? (18 lignes) Même question pour les langues de votre choix ?
7. Quelle est la date d'indépendance de la France ?
8. Quelles sont les dates d'indépendance des pays d'Europe ? (46 lignes)
9. Quelles sont les villes françaises de plus de 200 000 habitants ? (10 lignes)
10. Pour chaque pays européen, calculer la densité, le GNP par habitant, et l'espérance de vie, ordonner par densité décroissante. (46 lignes)
11. Quels sont les pays où l'espérance de vie n'est pas inférieure à 77 ans et le pnb par habitant n'est pas supérieur à (0.010) ? (10 lignes)
12. Quels sont les pays tels que la condition (espérance de vie supérieure ou égale à 77 ans ou PNB par habitant inférieur à (0.01)) n'est pas vérifiée ? (16 lignes)
13. Quels sont les pays où une langue est officielle sans être parlée par au moins la moitié de la population ? (92 lignes)
14. Quels sont les pays qui ont au moins une langue officielle ? (190 lignes)
15. Quels sont les noms des pays qui comptent plus de 100 000 000 d'habitants ? (10 lignes)

## Requêtes multi-tables

On peut aussi combiner plusieurs tables. Pour ce TP, nous allons seulement présenter le produit cartésien de deux tables :  $T \times S$  est la table dont les colonnes sont les colonnes de  $S$  et les colonnes de  $T$  et dont les lignes sont tous les couples  $(l_1, l_2)$  où  $l_1$  est une ligne de  $T$  et  $l_2$  est une ligne de  $S$ . En SQL, on écrira :

```
SELECT col1,col2
FROM table1,table2
WHERE condition;
```

Par exemple,

```
SELECT language
FROM world.country as c, world.countrylanguage as l
WHERE c.countrycode = l.countrycode and c.continent = 'Europe';
```

affichera les langues parlées en Europe. Remarquez l'utilisation des **as** pour donner de nouveaux noms aux tables et l'utilisation de **c.countrycode** pour lever l'ambiguïté sur des noms de colonnes qui seraient éventuellement les mêmes.

## Exercice 2

Ecrire des requêtes pour les questions suivantes :

1. Quels sont les noms des capitales sud-américaines ? (14 lignes)
2. Quels sont les noms des pays où le français est langue officielle ? (18 lignes)

3. Quelles sont les pays où l'espagnol est langue officielle et la forme de gouvernement est 'Federal Republic' ? (3 lignes)
4. Quels sont les pays qui ont au moins deux langues officielles ? (38 lignes)
5. Quels sont les pays qui n'ont pas de langue officielle ? (49 lignes)
6. Quels sont les pays qui comportent au moins deux villes de plus de 1 000 000 habitants ? (32 lignes)
7. Quelles sont les régions qui ne comportent qu'une seule forme de gouvernement ? (3 lignes)
8. Quelles sont les régions où on ne trouve pas de monarchie ? (9 lignes)