TD 1 : Algèbre relationnelle

Logistique et Algèbre relationnelle

2025-09-26

Avec solutions

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 commantaire SQL

Requêtes monotables

On rappele 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

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

Par exemple :

```
SELECT name_country, lifeexpectancy
FROM 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').

Éxercice 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)

Solution

$\Pi($`country, region`$)$

```
SELECT DISTINCT region
FROM country ;
```

2. Quelles sont les régions situées en Europe ? (6 lignes)

Solution

$\Pi(\sigma($`country, continent = 'Europe'`$), $ `region`$)$

```
SELECT DISTINCT region
FROM country
WHERE continent = 'Europe' ;
```

3. Quels sont les pays situés en Europe du sud (‘Southern Europe’) ? (15 lignes)

Solution

$\Pi(\sigma($`country, region = 'Southern Europe'`$), $ `name_country`$)$

```
SELECT name_country
FROM country
WHERE region = 'Southern Europe' ;
```

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)

Solution

$\Pi(\sigma($`country, region = 'Western Europe'`$), $ `capital`$)$

```
SELECT capital
FROM country
WHERE region = 'Western Europe' ;
```

5. A partir de la table `countrylanguage`, quels sont les langues qui sont officielles dans au moins un pays ? (102 lignes)

Solution

$\Pi(\sigma($` countrylanguage, isofficial`$), $ `language`$)$

```
SELECT DISTINCT language
FROM countrylanguage
WHERE isofficial ;
```

La variante de l’algèbre relationnelle vue en cours opérant sur les ensembles, `DISTINCT` est toujours implicitement implémenté par la requête algébrique. Ce n’est pas le cas en SQL (variante multi-ensembliste de l’algèbre relationnelle), il existe donc un opérateur d’élimination des doublons : `DISTINCT`.

6. Quels sont les codes des pays où le français est langue officielle ? (18 lignes) Même question pour les langues de votre choix ?

Solution

$\Pi(\sigma($ countrylanguage, language = 'French' $\wedge$ isofficial$)$, countrycode$)$

```
SELECT countrycode
FROM countrylanguage
WHERE language = 'French' AND isofficial ;
```

7. Quelle est la date d'indépendance de la France ?

Solution

$\Pi(\sigma($country, countrycode = 'FRA'$)$, indepyear$)$

```
SELECT indepyear
FROM country
WHERE countrycode = 'FRA' ;
```

8. Quelles sont les dates d'indépendance des pays d'Europe ? (46 lignes)

Solution

$\Pi(\sigma($country, continent = 'Europe'$)$, name_country , indepyear$)$

```
SELECT name_country , indepyear
FROM country
WHERE continent = 'Europe' ;
```

9. Quelles sont les villes françaises de plus de 200 000 habitants ? (10 lignes)

Solution

$\Pi(\sigma($ city, countrycode = 'FRA' $\wedge$ population > 200000$)$, name , population$)$

```
SELECT name , population
FROM city
WHERE countrycode = 'FRA' AND population > 200000 ;
```

10. Pour chaque pays européen, calculer la densité, le GNP par habitant, et l'espérance de vie, ordonner par densité décroissante. (46 lignes)

Solution

R1 = $\Pi(\sigma($country, continent='Europe'$)$, name_country, population_country/surfacearea, gnp/population_country, lifeexpectancy$)$
R = $\rho($R1, population_country/surfacearea $\mapsto$ density, gnp/population_country $\mapsto$ gnp_per_hab$)$

```
SELECT name_country , population_country / surfacearea as density ,
       gnp / population_country as gnp_per_hab , lifeexpectancy
FROM country
WHERE continent = 'Europe'
ORDER BY density DESC;
```

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)

Solution

$\Pi(\sigma(\texttt{country}, \neg(\texttt{lifeexpectancy<77}) \wedge \neg(\texttt{gnp/population\_country>0,01})), \texttt{name\_country})$
ou, de manière équivalente
$\Pi(\sigma(\texttt{country}, (\texttt{lifeexpectancy>=77}) \wedge (\texttt{gnp/population\_country<=0.01})), \texttt{name\_country})$

```
SELECT name_country
FROM country
WHERE NOT (lifeexpectancy < 77) AND
      NOT (gnp / population_country > 0.01) ;
```

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 vérifée ? (16 lignes)

Solution

$\Pi(\sigma(\texttt{country}, \neg((\texttt{lifeexpectancy>=77}) \vee (\texttt{gnp/population\_country<0,01}))), \texttt{name\_country})$
ou, de manière équivalente
$\Pi(\sigma(\texttt{country}, (\texttt{lifeexpectancy<77}) \wedge (\texttt{gnp/population\_country >= 0.01})), \texttt{name\_country})$

```
SELECT name_country
FROM country
WHERE NOT (lifeexpectancy >= 77 OR (gnp / population_country < 0.01) );
```

13. Quels sont les pays où une langue est officielle sans être parlée par au moins la moitié de la population ? (92 lignes)

Solution

$\Pi(\sigma(\texttt{countrylanguage}, \texttt{isofficial} \wedge \texttt{percentage<50}), \texttt{countrycode})$

```
SELECT DISTINCT countrycode
FROM countrylanguage
WHERE isofficial AND percentage < 50 ;
```

14. Quels sont les codes des pays qui ont au moins une langue officielle ? (190 lignes)

Solution

$\Pi(\sigma(\texttt{countrylanguage}, \texttt{isofficial}), \texttt{countrycode})$

```
SELECT DISTINCT countrycode
FROM countrylanguage
WHERE isofficial ;
```

15. Quels sont les noms des pays qui comptent plus de 100 000 000 d'habitants ? (10 lignes)

Solution

$\Pi(\sigma(\texttt{country}, \texttt{population\_country} > 100000000), \texttt{name\_country})$

```
SELECT name_country
FROM country
WHERE population_country > 100000000 ;
```

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 country as co, countrylanguage as l
WHERE co.countrycode = l.countrycode and co.continent = 'Europe';
```

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

Éxercice 2

Ecrire des requêtes pour les questions suivantes :

16. Quels sont les noms des capitales sud-américaines ? (14 lignes)

Solution

$\Pi(\sigma(\rho(\texttt{city}, \texttt{countrycode} \mapsto \texttt{city\_countrycode}) \bowtie_{\texttt{capital=id}} \texttt{country}, \texttt{region='South America'}), \texttt{name})$
En toute rigueur la relation qui résulte d'une opération ne peut pas contenir deux attributs de même nom, d'où le renommage utilisé dans la requête algébrique ci-dessus. En pratique, on suppose que les renommages `attribut`↦`table.attribut` sont faits implicitement chaque fois que nécessaire.
On peut alors écrire plus simplement :
$\Pi(\sigma(\texttt{city} \bowtie_{\texttt{capital=id}} \texttt{country},\ \texttt{region='South America'}),\ \texttt{name})$

```
SELECT name
FROM country , city
WHERE capital = id AND region = 'South America' ;
```

Il est préférable d'utiliser l'opération de jointure :

```
SELECT name
FROM country JOIN city ON capital = id
WHERE region = 'South America';
```

17. Quels sont les noms des pays où le français est langue officielle ? (18 lignes)

Solution

$\Pi(\sigma(\texttt{country} \bowtie_{\texttt{countrycode}} \texttt{countrylanguage},\ \texttt{isofficial} \wedge \texttt{language='French'}),\ \texttt{name\_country})$

```
SELECT co.name_country
FROM country co JOIN countrylanguage l ON co.countrycode = l.countrycode
WHERE l.isofficial AND l.language = 'French' ;
```

18. Quelles sont les pays où l'espagnol est langue officielle et la forme de gouvernement est 'Federal Republic' ? (3 lignes)

Solution

$\Pi(\sigma(\texttt{country} \bowtie_{\texttt{countrycode}} \texttt{countrylanguage},$
$\texttt{isofficial} \wedge \texttt{language='Spanish'} \wedge \texttt{governmentform='Federal Republic'}),\ \texttt{name\_country})$

```
SELECT co.name_country
FROM country co JOIN countrylanguage l ON co.countrycode = l.countrycode
WHERE l.isofficial AND
      l.language = 'Spanish' AND
      co.governmentform = 'Federal Republic' ;
```

19. Quels sont les pays qui ont au moins deux langues officielles ? (38 lignes)

Solution

$\Pi(\sigma(\texttt{country} \bowtie_{\texttt{countrycode}} \texttt{l1} \bowtie_{\texttt{countrycode}} \texttt{l2},$
$\texttt{l1.isofficial} \wedge \texttt{l2.isofficial} \wedge \texttt{l1.language} \neq \texttt{l2.language}),\ \texttt{name\_country})$

```
SELECT DISTINCT co.name_country
FROM
   country AS co
   JOIN countrylanguage AS l1 ON co.countrycode = l1.countrycode
   JOIN countrylanguage AS l2 ON co.countrycode = l2.countrycode
                              AND NOT (l1.language = l2.language)
WHERE l1.isofficial AND l2.isofficial ;
```

On dispose aussi de la syntaxe `JOIN … USING (a1, … ,ak)` pour faire des jointures naturelles qui ne considèrent que les attributs communs $\texttt{a}_1$ à $\texttt{a}_k$.

```
SELECT DISTINCT name_country
FROM country
   JOIN countrylanguage AS l1 USING (countrycode)
   JOIN countrylanguage AS l2 USING (countrycode)
WHERE l1.isofficial AND l2.isofficial AND l1.language <> l2.language ;
```

20. Quels sont les pays qui n'ont pas de langue officielle ? (49 lignes)

Solution

$\Pi(\texttt{country}, \texttt{name\_country}) - \Pi(\sigma(\texttt{country} \bowtie_{\texttt{countrycode}} \texttt{countrylanguage}, \texttt{isofficial}), \texttt{name\_country})$

```
(
   SELECT name_country
   FROM country
)
EXCEPT
(
   SELECT name_country
   FROM country JOIN countrylanguage USING (countrycode)
   WHERE isofficial
);
```

21. Quels sont les pays qui comportent au moins deux villes de plus de 1 000 000 habitants ? (32 lignes)

Solution

`c1 = city`
`c2 = city`
$\Pi(\sigma(\texttt{country} \bowtie_{\texttt{countrycode}} \texttt{c1} \bowtie_{\texttt{countrycode}} \texttt{c2},\ \texttt{c1.population\_city>1000000} \wedge$
$\texttt{c2.population\_city>1000000} \wedge \texttt{c1.id} \neq \texttt{c2.id}), \texttt{name\_country})$

```
SELECT DISTINCT name_country
FROM country AS co
   JOIN city AS c1 ON co.countrycode = c1.countrycode
   JOIN city AS c2 ON co.countrycode = c2.countrycode AND (NOT c1.id = c2.id)
WHERE c1.population > 1000000 AND c2.population > 1000000;
```

22. Quelles sont les régions qui ne comportent qu'une seule forme de gouvernement ? (3 lignes)

Solution

c1 = city
c2 = city
Π(country , region)
$-\ \Pi(\sigma(\texttt{c1} \bowtie_{\texttt{region}} \texttt{c2},\ \texttt{c1.governmentform<>c2.governmentform}))$

```
(
   SELECT DISTINCT region
   FROM country
)
EXCEPT
(
   SELECT DISTINCT c1.region
   FROM country as c1
      JOIN country as c2 ON c1.region = c2.region
                        AND c1.governmentform <> c2.governmentform
) ;
```

23. Quelles sont les régions où on ne trouve pas de monarchie ? (9 lignes)

Solution

La requête algébrique a la même forme que la précédente mais nécessite d'introduire une opération LIKE.

```
(
   SELECT region
   FROM country
)
EXCEPT
(
   SELECT region
   FROM country
   WHERE governmentform LIKE '%Monarchy%'
) ;
```