TD 2 : SQL

SQL Interrogation de données

2024-09-27

Avec solutions

- **L3 MIASHS/Ingémath**
- **Université Paris Cité**
- Année 2024-2025
- Course Homepage

- Moodle

Schéma `world` (rappel)

Utiliser les commandes suivantes pour retrouver les schémas de tables `country`, `countrylanguage` et `city`.

```
bd_2023> \d world."nom de table"
```

Vue d'ensemble du schéma `world`

Écriture de requêtes

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

Opération	Algèbre rel.	SQL
Projection	$\Pi_{\text{liste d'attributs}}(R)$	`SELECT attributs FROM R`
Sélection	$\sigma_{\text{condition}}R$	`SELECT * FROM R WHERE condition`
Renommage	$\rho_{\text{nom1}\to\text{nom2}}(R)$	`SELECT nom1 AS nom2 FROM table1 AS table2`
Union	$R \cup S$	`(SELECT * FROM R) UNION (SELECT * FROM S)`
Différence	$R - S$	`(SELECT * FROM R) EXCEPT (SELECT * FROM S)`
Produit	$R \times S$	`SELECT * FROM R,S`
Intersection	$R \cap S$	`(SELECT * FROM R) INTERSECT (SELECT * FROM S)`

Opérations avancées

Jointures

La jointure est une opération qui permet de recoller deux relations ensemble. On la note avec le signe $\bowtie_C$ où C est une condition. Formellement, ce n'est qu'une sélection selon la condition C d'un produit cartésien, c'est-à-dire, par définition :

$$R \bowtie_C S = \sigma_C(R \times S).$$

Cette opération est très courante dans les requêtes. Cela vient du fait que les bases de données sont conçues pour ne pas avoir d'information redondante.

Prenons l'exemple du schéma `world` et des tables `world.country` et `world.city`. La table `world.country` possède une colonne `capital` qui contient le numéro d'identification d'une ville de la table `world.city`. Par exemple, pour la France, la colonne `capital` vaut 2974 et la ville numéro 2974 est Paris :

```
bd_2023-24> SELECT capital
FROM world.country
WHERE countrycode='FRA';
bd_2023-24> SELECT name
FROM world.city
WHERE id = 2974;
```

Pour lister les noms des capitales mondiales, on va avoir besoin de recoller les deux tables en prenant soin de respecter la condition d'égalité entre la capitale et le numéro d'identification de la ville. Cela peut s'écrire en SQL comme suit :

```
SELECT name
FROM world.city AS V JOIN
     world.country AS P ON
     V.id = P.capital;
```

Parfois, on veut recoller deux tables avec la condition que les colonnes qui ont le même nom doivent être égales. Cela arrive souvent si la base de données a bien été conçue. On appelle cette jointure une jointure *naturelle*, qu'on peut écrire avec `NATURAL JOIN`. La requête

```
SELECT name
FROM world.city NATURAL JOIN world.country
WHERE continent = 'Oceania';
```

renvoie la liste des villes d'Océanie. La jointure est faite implicitement sur la condition

`world.city.countrycode = world.country.countrycode`

la table résultante n'aura qu'une seule colonne `countrycode`. Si l'on souhaite ne faire la jointure naturelle que sur un sous-ensemble `a1,...,ak` des attributs communs, on peut utiliser `JOIN ... USING (a1,...,ak)`.

Requêtes nommées : utilisation de `WITH`

Lorsque les requêtes se compliquent, il est parfois commode de la découper en plusieurs requêtes qu'on manipulera par la suite. Par exemple, trouver le nom des capitales européennes ayant plus d'un million d'habitants peut se découper ainsi : on résout la requête `CAP` qui contient le nom, le continent et la population des capitales. Puis on résout la requête finale en filtrant `CAP`. Cela peut s'écrire avec l'opérateur `WITH` :

```
WITH CAP AS (
  SELECT name, continent, population
  FROM world.city AS V JOIN
    world.country AS P ON  V.id=P.capital
)

SELECT name
FROM CAP
WHERE continent = 'Europe' AND population > 1000000;
```

Requêtes

- Quels sont les noms des capitales Sud-Américaines ? (14 lignes)

Solution

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

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

Solution

```
SELECT name_country
FROM world.country NATURAL JOIN world.countrylanguage
WHERE isofficial AND language='French';
```

- Quels sont les pays dont le gouvernement est monarchique (constitutionnelle ou pas) ? (43 lignes)

Solution

```
SELECT name_country
FROM world.country
WHERE governmentform LIKE '%Monarchy%';
```

- Quelles sont les monarchies d'Europe dont la capitale compte moins d'un million d'habitants ? (8 lignes)

Solution

```
WITH T1 AS (
  SELECT *
  FROM world.country
  WHERE governmentform LIKE '%Monarchy%' AND continent='Europe' )

SELECT name_country
FROM T1 JOIN world.city ON capital=id
WHERE population < 1000000;
```

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

Solution

```
WITH monarchy AS (
  SELECT region
  FROM world.country
  WHERE governmentform LIKE '%Monarchy%')

(SELECT region
 FROM world.country)
EXCEPT
(SELECT *
 FROM monarchy);
```

- Quelles sont les langues qui ne sont langues officielles dans aucun pays ? (355 lignes)

Solution

```
WITH R AS (
  SELECT language
  FROM world.countrylanguage
  WHERE isofficial
)

(SELECT DISTINCT language
FROM world.countrylanguage)
EXCEPT
(SELECT *
FROM R);
```

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

Solution

```
WITH R AS (
  SELECT countrycode
  FROM world.countrylanguage
  WHERE isofficial
),
S AS (
  SELECT countrycode
  FROM world.country
  EXCEPT
  SELECT * FROM R
)

SELECT name_country
FROM world.country NATURAL JOIN S;
```

- Quels sont les pays dont la seule langue officielle est le français ? (9 lignes)

Solution

```
WITH R AS(
  SELECT DISTINCT T1.countrycode
  FROM world.countrylanguage as T1 JOIN
      world.countrylanguage as T2 ON
      T1.countrycode = T2.countrycode
  WHERE T1.language <> T2.language AND
      T1.isofficial AND
      T2.isofficial),
S AS (
  SELECT countrycode
  FROM world.countrylanguage
  WHERE language='French' AND isofficial
  EXCEPT
  (SELECT * FROM R)
)

SELECT name_country
FROM world.country NATURAL JOIN S;
```

En utilisant `EXCEPT` :

```
WITH OnlyFrench AS (
  (SELECT countrycode
  FROM countrylanguage
  WHERE isofficial AND language = 'French')
  EXCEPT
  (SELECT countrycode
  FROM countrylanguage
  WHERE isofficial AND language <> 'French')
)

SELECT name_country
FROM world.country NATURAL JOIN OnlyFrench;
```

- Les noms des pays où le français n'est pas la seule langue officielle. (9 lignes)

Solution

```
WITH R AS (
SELECT DISTINCT T1.countrycode FROM world.countrylanguage as T1
                    JOIN world.countrylanguage as T2
                    ON T1.countrycode = T2.countrycode
WHERE T1.language <> T2.language AND
     T1.language = 'French' AND
     T1.isofficial AND
     T2.isofficial)

SELECT name_country FROM R NATURAL JOIN world.country;
```

En utilisant `INTERSECT` :

```
WITH AlsoFrench AS (
  (SELECT countrycode
  FROM world.countrylanguage
  WHERE isofficial AND language = 'French')
  INTERSECT
  (SELECT countrycode
  FROM world.countrylanguage
  WHERE isofficial AND language <> 'French')
)

SELECT name_country
FROM world.country NATURAL JOIN
     AlsoFrench;
```

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

Solution

```
WITH R AS (
  SELECT T1.region
  FROM world.country AS T1 JOIN
       world.country AS T2 ON
       T1.region = T2.region
  WHERE T1.governmentform <> T2.governmentform)

SELECT region
FROM world.country
EXCEPT
SELECT *
FROM R;
```

- Quelles sont les langues officielles des pays dont la capitale compte plus de 5 000 000 d'habitants ? (12 lignes)

Solution

```
SELECT DISTINCT language
FROM world.countrylanguage NATURAL JOIN
     world.city JOIN world.country ON capital=id
WHERE population > 5000000 and isofficial;
```

- Quels sont les pays où au moins trois langues sont parlées chacune par strictement plus de 10% de la population ? (35 lignes)

Solution

```
WITH R AS (
  SELECT T1.countrycode
  FROM  world.countrylanguage AS T1 JOIN
        world.countrylanguage AS T2 ON T1.countrycode = T2.countrycode JOIN
        world.countrylanguage AS T3 ON T1.countrycode = T3.countrycode
  WHERE T1.language <> T2.language AND
        T2.language <> T3.language AND
        T1.language <> T3.language AND
        T1.percentage > 10 AND
        T2.percentage > 10 AND
        T3.percentage > 10
)

SELECT DISTINCT name_country
FROM world.country NATURAL JOIN R;
```

ou

```
WITH tpc AS (
  SELECT *
  FROM world.countrylanguage
  WHERE percentage > 10),
three AS (
    SELECT DISTINCT t1.countrycode
    FROM tpc AS t1 JOIN
         tpc AS t2 ON (t1.countrycode=t2.countrycode AND t1.language < t2.language) JOIN
         tpc AS t3 ON (t1.countrycode=t3.countrycode AND t3.language > t2.language)
)

SELECT co.*
FROM world.country co NATURAL JOIN three ;
```

- Quelles sont les régions où il existe deux pays dont les espérances de vie diffèrent par au moins 10 ans ? (16 lignes)

Solution

```
SELECT DISTINCT T1.region
FROM world.country AS T1 JOIN
     world.country AS T2 ON T1.region = T2.region
WHERE T1.countrycode <> T2.countrycode AND
      (T1.lifeexpectancy - T2.lifeexpectancy >= 10 OR
       T2.lifeexpectancy - T1.lifeexpectancy >= 10);
```

- Quels sont les pays où l'anglais et le français sont des *langues officielles* ? (3 lignes)

Solution

```
WITH R AS (
  SELECT T1.countrycode
  FROM world.countrylanguage AS T1 JOIN
       world.countrylanguage AS T2 USING (countrycode)
  WHERE T1.language='French' AND
        T2.language='English' AND
        T1.isofficial AND
        T2.isofficial)

SELECT name_country
FROM R NATURAL JOIN
     world.country;
```

- Montrer comment calculer l'*intersection* de deux tables avec une *jointure* (sous certaines conditions).

Solution

Si `R` et `S` sont deux tables de même schéma, une jointure naturelle des deux tables devrait (en première approximation) calculer l'intersection des deux tables.
Pourtant

```
(SELECT * FROM R)
INTERSECT
(SELECT * FROM S)   ;
```

ne donne pas toujours le même résultat que

```
SELECT *
FROM    R NATURAL JOIN S ;
```

Essayez par exemple pour l'intersection de la table `country` avec elle-même :

```
(SELECT * FROM country)
INTERSECT
(SELECT * FROM country);
```

retourne la table `country`, soit 239 lignes.
La jointure naturelle

```
SELECT *
FROM country AS c1 NATURAL JOIN
     country AS c2;
```

ne donne que 167 lignes. On peut réconcilier les deux résultats en ne conservant que les lignes pour lesquelles aucune colonne ne prend la valeur `NULL`.
En vérifiant quels attributs peuvent prendre la valeur `NULL` dans le schéma, on peut tester cela par la requête

```
SELECT *
FROM world.country
WHERE indepyear IS NOT NULL AND
      lifeexpectancy IS NOT NUL AND
      gnp IS NOT NULL AND
      gnpold IS NOT NULL AND
      headofstate IS NOT NULL AND
      capital IS NOT NULL;
```

qui retourne bien 167 lignes.