- **L3 MIASHS/Ingémath**
- **Université Paris Cité**
- Année 2023-2024
- Course Homepage

- Moodle

Toutes les questions portent sur le schéma `world` rappelé ci-dessous.

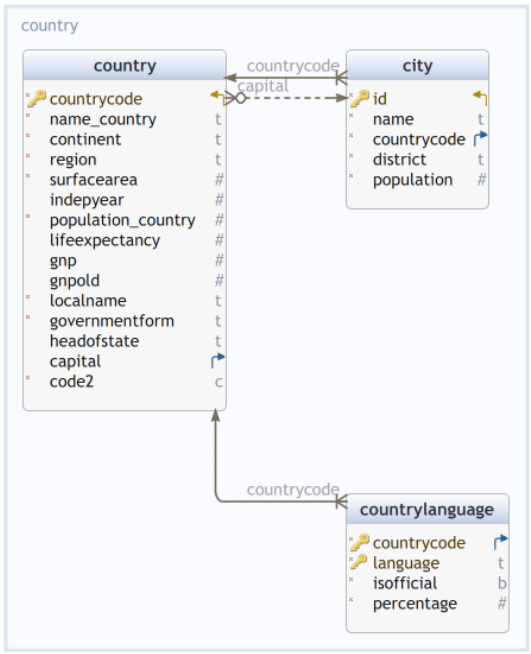

FIG. 1 : Schéma `world`

⚠ Les territoires qui sont inscrits dans la table `country` ne sont pas tous des pays, et pour certaines lignes, `population_country`, `gnp`, `governmentform`, … ne sont pas renseignés.

Pour chaque question, proposer une requête écrite en algèbre relationnelle ou en `SQL`.

1. Donner pour chaque pays, chaque district, la population totale qui réside dans des villes répertoriées dans `city`.

Solution

Un `district` est une unité territoriale incluse dans un territoire. Pour la France, les `district` correspondent aux *régions*, pour l'Allemagne aux *Laender*, ...

```
SELECT DISTINCT ci.district
FROM world.city ci
WHERE ci.countrycode='FRA' ;
```

Comme on doit sommer les populations des villes situées dans un district (et donc dans un pays), il faut partitionner `city` par district et pays.

```
SELECT ci.countrycode, ci.district,                    ②
       SUM(ci.population) as pop                       ③
FROM world.city ci
GROUP BY ci.countrycode, ci.district                   ①
```

① Deux colonnes pour le partitionnement (même si `countrycode` est peut-être déterminé par `district`)
② On ne peut projeter que sur les colonnes utilisées pour partitionner et ...
③ Sur des colonnes agrégées.
Après partition/aggrégation, la table résultat contient une ligne pour chaque combinaison des attributs/colonnes servant au partitionnement. Sur chaque ligne du résultat, on peut faire figurer des aggrégations.
Si on veut disposer des totaux par pays en plus, on peut utiliser la construction `GROUP BY ROLLUP(...)`

```
SELECT ci.countrycode, ci.district,
       SUM(ci.population) as pop
FROM world.city ci
GROUP BY ROLLUP(ci.countrycode, ci.district)
ORDER BY ci.countrycode, ci.district
LIMIT 20 ;
```

On peut chercher à imiter le résultat de `GROUP BY ROLLUP` avec les constructions vues en cours.

```
WITH r AS (
  SELECT ci.countrycode, ci.district,
       SUM(ci.population) as pop
  FROM world.city ci
  GROUP BY ci.countrycode, ci.district
), s AS (
  SELECT r.countrycode, NULL AS district, sum(r.pop_district) AS pop
  FROM r
  GROUP BY r.countrycode
)

(SELECT * FROM s)
UNION
(SELECT * FROM r)
ORDER BY countrycode, district ;
```

La construction `GROUP BY ROLLUP (...)` ne permet pas définir des requêtes qui seraient hors de portée de `GROUP BY ...`, elle permet une écriture plus concise et plus claire.

2. Donner pour chaque pays le nombre d'habitants qui ne parlent pas une langue officielle. On suppose ici que chaque habitant ne parle qu'une seule langue.

Solution

```
WITH cc_pop_not_official AS (                                                    ①
  SELECT countrycode, SUM(percentage) as percentage
  FROM world.countrylanguage
  WHERE NOT(isofficial)
  GROUP BY countrycode
)

SELECT name_country, (population_country * percentage / 100)::INTEGER
FROM world.country NATURAL JOIN cc_pop_not_official;                             ②
```

① La requête intermédiaire `cc_pop_not_official` nous donne pour chaque pays le pourcentage de locuteurs des langues non-officielles du pays

② Grâce à une jounture avec `country`, on peut combiner ce pourcentage avec la population du pays.

On peut chercher à vérifier la convention avancée dans l'énoncé :

```
SELECT cl.countrycode, SUM(cl.percentage) AS total_percent
FROM world.countrylanguage cl
GROUP BY cl.countrycode
HAVING SUM(cl.percentage) > 100
ORDER BY total_percent DESC
LIMIT 10 ;
```

3. Donner, pour chaque région ayant au moins 50 millions d'habitants, le ou les pays les moins peuplés de la région.

Solution

```
WITH r_minpop_g50M AS (
  SELECT region, MIN(population_country) AS population_country
  FROM world.country
  GROUP BY region
  HAVING SUM(population_country) >= 50000000)

SELECT co.region, co.name_country
  FROM world.country co NATURAL JOIN r_minpop_g50M ;
```

Warning

Ne pas confondre les régions ayant au moins 50 millions d'habitants et les régions où au moins un pays a plus de 50 millions d'habitants.

4. Liste des pays où la langue la plus parlée n'est pas officielle

Solution

```
WITH r AS (
  SELECT cl.*,
    co.name_country,
    rank() OVER w AS rang
    FROM world.countrylanguage cl NATURAL JOIN
         world.country co
  WINDOW w AS (PARTITION BY cl.countrycode ORDER BY cl.percentage DESC)
)

SELECT DISTINCT r.name_country
FROM r
WHERE r.rang=1 AND NOT r.isofficial ;
```

L'usage d'une fonction fenêtre est commode, mais c'est un marteau-pilon pour casser un œuf.

Solution

Alternatives sans fonction fenêtre :

```
WITH langmax AS (
  SELECT countrycode,
         MAX(percentage) AS percentage
  FROM world.countrylanguage
  GROUP BY countrycode)

SELECT DISTINCT name_country
FROM world.country NATURAL JOIN
     world.countrylanguage NATURAL JOIN
     langmax
WHERE NOT(isofficial);
```

```
SELECT DISTINCT cl.countrycode
FROM world.countrylanguage cl
WHERE NOT cl.isofficial AND
      NOT EXISTS (
        SELECT *
        FROM world.countrylanguage cl1
        WHERE cl1.isofficial AND
              cl1.countrycode = cl.countrycode
              cl1.percentage > cl.percentage
) ;
```

Une erreur fréquente consiste à lister les pays où il existe une langue officielle qui est moins parlée qu'une langue non-officelle. Ce n'est pas la même chose que la liste des pays où toutes les langues officielles sont moins parlées qu'une certaine langue non-officiele

Solution en calcul des tuples

$$\{t : \texttt{country}(t) \quad \wedge \exists s, s \in \texttt{countrylanguage} \wedge \neg s.\texttt{isofficial} \wedge \\ \forall u \neg (\texttt{countrylanguage}(u) \wedge u.\texttt{isofficial} \wedge u.\texttt{percentage} > s.\texttt{percentage})\}$$

Une autre erreur fréquente consiste à ne lister que les pays où il existe une langue non-officielle parlée par au moins la moitié de la population :

```
SELECT cl.countrycode
FROM world.countrylanguage cl
WHERE cl.percentage > 50 AND NOT c.isofficial ;
```

Il est très possible que dans un pays, aucune langue ne soit parlée par la moitié de la population, que toutes les langues soient minoritaires, et que la langue la plus parlée ne soit pas officielle.

1. Liste des pays où aucune ville ne compte plus 3 millions d'habitants.

Solution

```
WITH r AS (
    SELECT ci.countrycode,
           SUM(CAST (ci.population > 3000000 AS INTEGER)) AS nb
    FROM world.city ci
    GROUP BY ci.countrycode
)                                                              ①

SELECT co.name_country, co.continent
FROM world.country co LEFT JOIN
     r ON (co.countrycode = r.countrycode)
WHERE r.nb IS NULL OR r.nb = 0
ORDER BY co.continent, co.name_country;                        ②
```

① La requête `r` collecte les codes de pays et pour chaque pays le nombre de villes de plus de 3000000 d'habitants.

② En effectuant la jointure externe entre `country` et `r`, on récupère les territoires sans villes et les territoires où toutes les villes comptent moins de 3000000 d'habitants.

On a ajouté `continent` parmi les colonnes de projection pour rendre plus lisibles les résultats.

Solution

Alternative utilisant `EXCEPT` :

```
WITH cc_no_big_city AS (
  (SELECT countrycode
   FROM world.country)
   EXCEPT
  (SELECT countrycode
   FROM world.city
   WHERE population > 3000000))

SELECT name_country, continent
FROM world.country NATURAL JOIN cc_no_big_city
ORDER BY continent, name_country;
```

Cette requête donne le même résultat que la précédente.

Solution

Alternative utilisant `NOT IN` :

```
SELECT name_country, continent
  FROM world.country
 WHERE countrycode NOT IN (
     SELECT countrycode
       FROM world.city
      WHERE population > 3000000)
 ORDER BY continent, name_country;
```

Cette requête donne encore le même résultat que la précédente.

Solution en calcul des tuples

Traduction quasi-litérale de la question en formule du calcul des tuples :

$$\{t : \texttt{country}(t) \wedge (\nexists s \quad \texttt{city}(s) \wedge s.\texttt{countrycode} = t.\texttt{countrycode} \wedge s.\texttt{population} > 3000000)\}$$

. Cette formule se traduit presque mécaniquement avec une requête imbriquée

```
SELECT co.*
FROM world.country co
WHERE NOT EXISTS (
     SELECT *
     FROM world.city ci
     WHERE ci.countrycode=co.countrycode AND ci.population > 3000000) ;
```

on peut aussi réécrire la formule en

$$\texttt{country} \backslash \{t : \texttt{country}(t) \wedge (\exists s \quad \texttt{city}(s) \wedge s.\texttt{countrycode} = t.\texttt{countrycode} \wedge s.\texttt{population} > 3000000)\}$$

6. Liste des *formes de gouvernement* (*governmentform*) pour lesquelles dans tous les pays possédant cette forme de gouvernement, aucune langue n'est officielle.

Solution

```
WITH r AS (
    SELECT DISTINCT cl.countrycode
    FROM world.countrylanguage cl
    WHERE cl.isofficial
), s AS (

    SELECT DISTINCT co.governmentform
    FROM world.country co
    WHERE co.countrycode IN (
    SELECT r.countrycode FROM r)
)

SELECT DISTINCT co.governmentform
FROM world.country co

EXCEPT

SELECT s.governmentform
FROM s  ;
```

Solution

Alternative utilisant `EXCEPT` :

```
(SELECT governmentform
   FROM world.country)
EXCEPT
(SELECT governmentform
   FROM world.country NATURAL JOIN world.countrylanguage
  WHERE isofficial);
```

Solution

Alternative avec partition, agrégation

```
SELECT co.governmentform
FROM world.country  co LEFT join
     world.countrylanguage cl ON (co.countrycode=cl.countrycode)
GROUP BY co.governmentform
HAVING SUM(CAST(cl.isofficial AS INTEGER))=0 OR
       SUM(CAST(cl.isofficial AS INTEGER)) IS NULL;
```

Si on oublie la condition `SUM(CAST(cl.isofficial AS INTEGER)) IS NULL`, on ne retrouve pas `Co-administrated` qui concerne des territoires qui ne sont pas mentionnés dans la table `countrylanguage`.

7. Donner pour chaque région, le minimum du PIB par habitant (c'est-à-dire le résultat de la division `1000000 * gnp / population_country`, puisque le PIB est donné en millions) dans la région, un des pays (son `countrycode`) de la région où ce minimum est réalisé, le maximum du PIB par habitant et un des pays (son `countrycode`) où ce maximum est réalisé.

Solution

```
WITH r AS (
    SELECT co.region, co.countrycode, co.name_country,
        1000000*co.gnp/co.population_country AS gnp_per_cap,
        row_number() OVER w_max AS rang_max,
        row_number() OVER w_min AS rang_min
    FROM world.country co
    WHERE co.population_country IS NOT NULL AND co.population_country >0 AND co.gnp IS NOT NUL
    WINDOW
        w_max AS (PARTITION BY co.region ORDER BY co.gnp/co.population_country DESC),
        w_min AS (PARTITION BY co.region ORDER BY co.gnp/co.population_country)
), r1 AS (
    SELECT r.*
    FROM r
    WHERE r.rang_max=1
), r2 AS (
    SELECT r.*
    FROM r
    WHERE r.rang_min=1
)

SELECT r1.region, r1.countrycode, r1.gnp_per_cap,
       r2.countrycode, r2.gnp_per_cap
FROM r1 JOIN r2 ON (r1.region=r2.region)  ;
```

Solution

Alternative sans fonction fenêtre :

```
WITH cc_gpc AS (
  SELECT countrycode, 1000000 * gnp/population_country AS gnp_per_cap
    FROM world.country
   WHERE population_country > 0),
region_mingpc_maxgpc AS (
  SELECT region, MIN(gnp_per_cap) AS mingpc, MAX(gnp_per_cap) AS maxgpc
    FROM world.country NATURAL JOIN cc_gpc
   GROUP BY region),
region_cc_mingpc AS (
  SELECT region, MIN(c1.countrycode) AS countrycode, c1.gnp_per_cap
    FROM world.country c NATURAL JOIN region_mingpc_maxgpc r
         JOIN cc_gpc c1 ON c.countrycode = c1.countrycode
                           AND r.mingpc = c1.gnp_per_cap
   GROUP BY region, c1.gnp_per_cap),
region_cc_maxgpc AS (
  SELECT region, MIN(c1.countrycode) AS countrycode, c1.gnp_per_cap
    FROM world.country c NATURAL JOIN region_mingpc_maxgpc r
         JOIN cc_gpc c1 ON c.countrycode = c1.countrycode
                           AND r.maxgpc = c1.gnp_per_cap
   GROUP BY region, c1.gnp_per_cap)

SELECT *
  FROM region_cc_mingpc JOIN region_cc_maxgpc
       USING (region);
```

Erreur commune

Avec une table `foo` de schéma `(col1, col2, col3, col4)`, la requête suivante n'est pas correcte :

```
SELECT col1, col2, SUM(col4) as bar
FROM foo
GROUP BY col1, col3 ;
```

Les colonnes qui figurent dans la clause de projection `SELECT ...` doivent
- figurer dans la clause de partitionnement `GROUP BY ...`, ici `col1` et `col3`
- représenter des agrégations, comme ici `SUM(col4)`

La colonne `col2` ne peut pas figurer dans la clause de projection !

Erreur commune

Dans une clause `WHERE` ou une condition de jointure, lorsqu'on compare deux colonnes, les types des deux colonnes doivent être compatibles (en général identiques).
Des conditions comme `language=isofficial` ou `governmentform=countrycode` n'ont pas de sens.