

- **L3 MIAHS/Ingémath/METIS**
- **Université Paris Cité**
- Année 2024-2025
- [Course Homepage](#)
- [Moodle](#)



⚠ Toutes les questions portent sur les schémas **pagila** et **world** rappelés ci-dessous.
Pour chaque question, proposer une requête écrite en algèbre relationnelle *OU* en SQL.

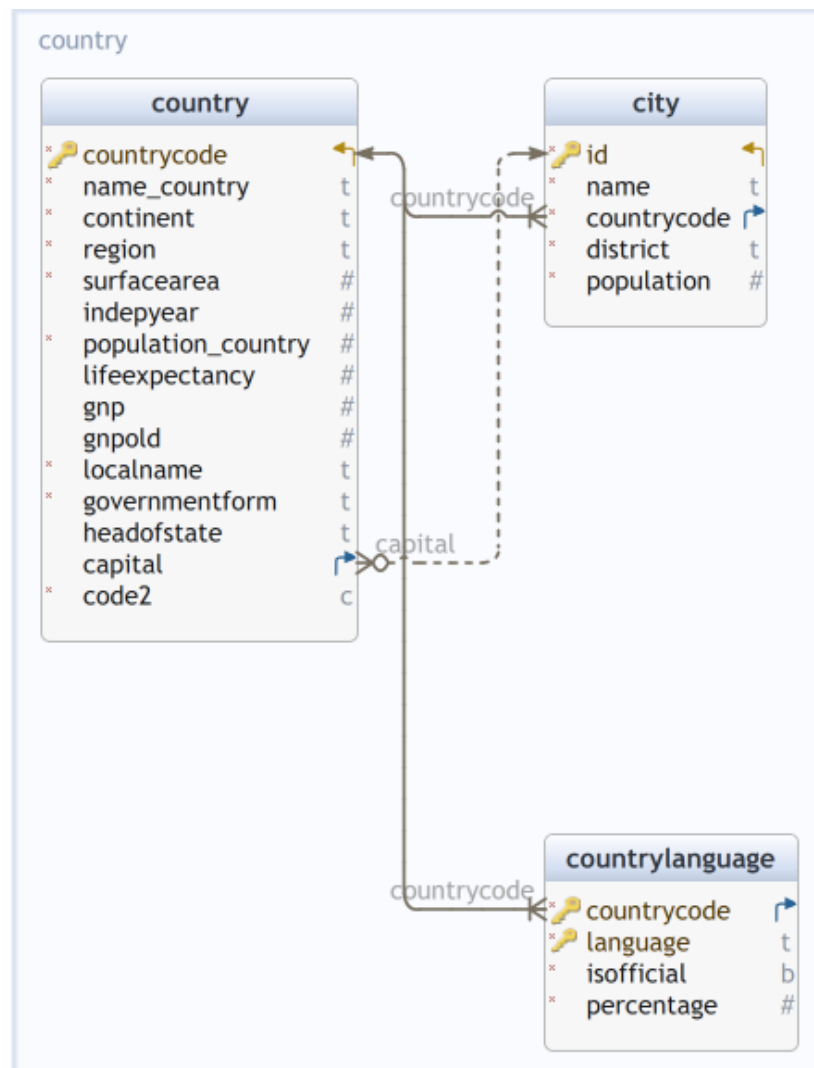


FIG. 1 : Schéma world

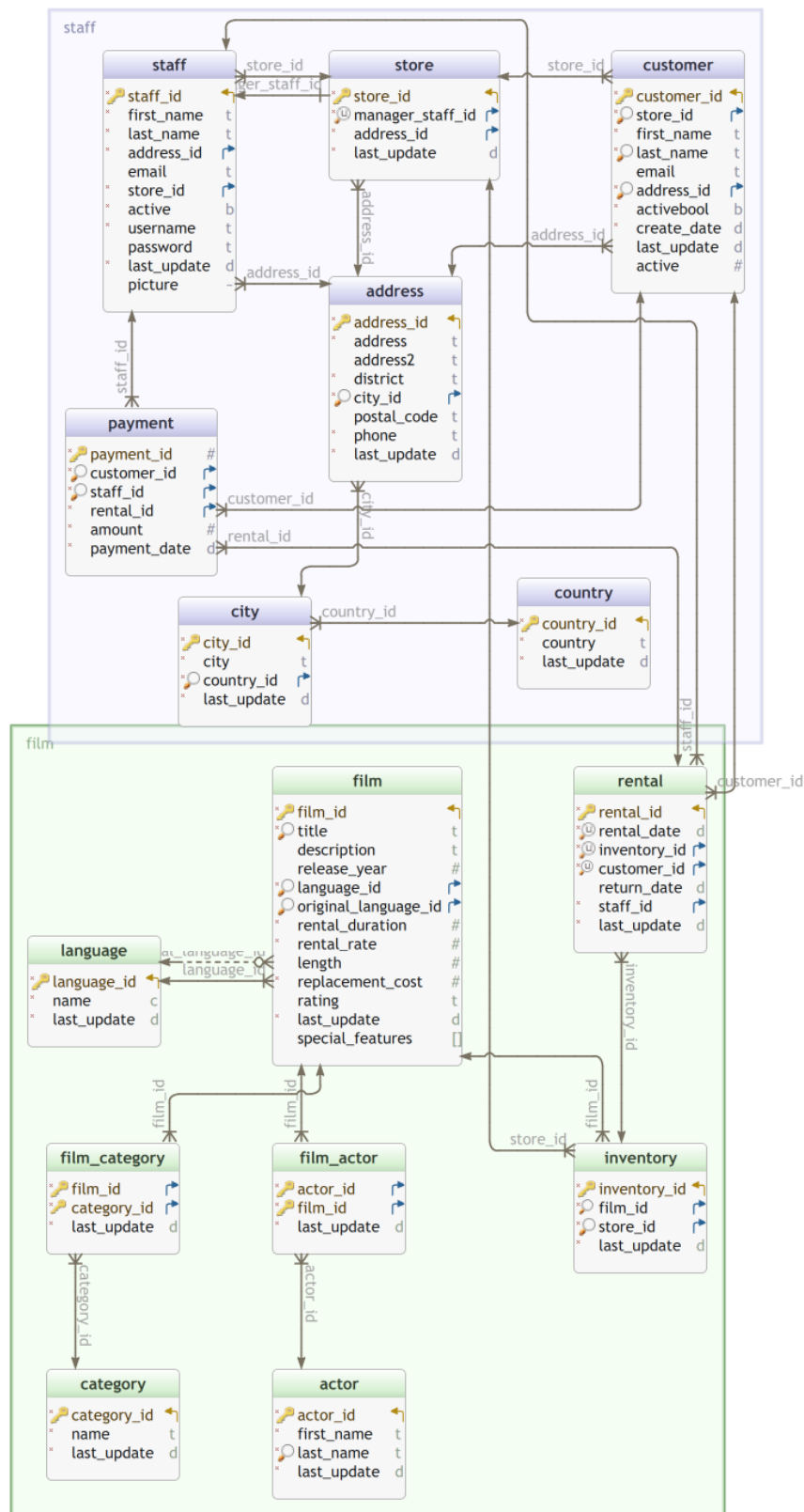


FIG. 2 : Schéma pagila, films

Schéma pagila

1. Lister la somme des paiements encaissés par année et par magasin.

💡 En PostgreSQL, on obtient le (numéro du) mois à partir d'une valeur de type `timestamp` avec la fonction `EXTRACT()`. On peut extraire les autres éléments de l'estampille de façon semblable, par exemple :

```
SELECT
    EXTRACT(YEAR FROM rental_date) AS annee
FROM
    pagila.rental;
```

💡 Solution

```
SELECT
    st.store_id,
    EXTRACT(YEAR FROM pay.payment_date) AS annee,
    SUM(amount) AS payments
FROM
    pagila.payment pay
JOIN
    pagila.staff st USING(staff_id)
GROUP BY
    st.store_id,
    EXTRACT(YEAR FROM pay.payment_date)
;
```

2. Lister l'identifiant, le nom, le prénom, de chaque employé ainsi que le total des paiements perçus par cet employé, et sa part dans le revenu de son magasin.

💡 Solution

```
SELECT
    st.store_id,
    st.staff_id,
    SUM(pay.amount) AS encaisse
FROM
    pagila.staff st
JOIN
    pagila.payment pay USING(staff_id)
GROUP BY ROLLUP(st.store_id, st.staff_id) ;
```

3. Lister par catégorie de film, les cinq titres qui ont engendré le plus gros chiffre d'affaire/revenu (somme des paiements perçus)

Solution

```
WITH film_payment AS (  
  SELECT  
    film_id,  
    fi.title,  
    SUM(amount) AS caffaires  
  FROM  
    pagila.payment pay  
  JOIN  
    pagila.rental re USING(rental_id)  
  JOIN  
    pagila.inventory inv USING(inventory_id)  
  JOIN  
    pagila.film fi USING(film_id)  
  GROUP BY  
    film_id, fi.title  
) , rank_per_cat AS (  
  SELECT  
    category_id,  
    ca.name,  
    film_id,  
    title,  
    caffaires,  
    RANK() OVER win AS rnk  
  FROM  
    film_payment fp  
  JOIN  
    pagila.film_category fc USING(film_id)  
  JOIN  
    pagila.category ca USING(category_id)  
  WINDOW win AS (PARTITION BY category_id ORDER BY caffaires DESC)  
)  
  
SELECT  
  name,  
  title,  
  caffaires,  
  rnk  
FROM rank_per_cat  
WHERE rnk <= 5  
ORDER BY name, rnk  
;
```

Schéma world

4. Lister pour chaque pays, la proportion de la population qui utilise l'une des deux langues les plus populaires du pays,

Solution

```
WITH ranking AS (  
  SELECT  
    countrycode, name_country, language, percentage,  
    RANK() OVER win AS rnk  
  FROM  
    world.country co  
  JOIN  
    world.countrylanguage cl USING(countrycode)  
  WINDOW  
    win AS (PARTITION BY countrycode ORDER BY percentage DESC)  
)  
  
SELECT  
  countrycode, name_country,  
  SUM(percentage) AS prop  
FROM  
  ranking  
WHERE  
  rnk <= 2  
GROUP BY countrycode, name_country  
ORDER BY prop DESC;
```

Solution

5. Lister pour chaque région (attribut **region** de **country**) les 10 villes les plus peuplées.

Solution

```
WITH R AS (  
  SELECT  
    co.countrycode, co.name_country,  
    ci.name,  
    RANK() OVER win AS rnk  
  FROM  
    world.country co  
  JOIN  
    world.city ci USING(countrycode)  
  WINDOW win AS (PARTITION BY co.countrycode ORDER BY ci.population DESC)  
)  
  
SELECT  
  name_country,  
  string_agg(name, ', ')  
FROM  
  R  
WHERE rnk <= 2  
GROUP BY countrycode, name_country  
;
```

6. Lister pour chaque langue, les deux pays où on trouve le plus grand nombre d'utilisateurs.

Solution

```
WITH R AS (  
  SELECT  
    language,  
    name_country,  
    percentage*population_country/100.0 AS pop_loc,  
    RANK() OVER win AS rnk  
  FROM  
    world.countrylanguage cl  
  NATURAL JOIN  
    world.country  
  WHERE  
    population_country IS NOT NULL  
  WINDOW win AS (PARTITION BY language ORDER BY percentage*population_country DESC)  
)  
  
SELECT  
  language,  
  string_agg(name_country, ', ') AS pays, SUM(pop_loc) AS loc  
FROM  
  R  
WHERE rnk <= 2  
GROUP BY language ;
```

7. Lister les pays en donnant leur rang par gnp (pnb) par habitant décroissant et leur rang par espérance de vie à la naissance (lifeexpectancy) décroissante.

Solution

```
SELECT  
  name_country,  
  lifeexpectancy ,  
  RANK() OVER win_gnp AS rnk_gnppercap,  
  RANK() OVER win_life AS rnk_lifeexp  
FROM  
  world.country  
WHERE  
  gnp IS NOT NULL AND  
  population_country IS NOT NULL AND  
  lifeexpectancy IS NOT NULL  
WINDOW  
  win_gnp AS (ORDER BY gnp/population_country DESC),  
  win_life AS (ORDER BY lifeexpectancy DESC)  
ORDER BY lifeexpectancy DESC ;
```