Exercice 2 : Exécution répartie de requêtes — 4 pts

On considère le schéma. Chaque table est un dataset.

Film (nF, titre, annéeF)

GenreFilm (nF, genre) le genre est un code de genre de 1 à 20 (1='SF', 2='comédie', …)

La répartition initiale d'un dataset est aléatoire (*ie.*, ne dépend pas d'un attribut) sur **3** machines, avec **une** partition par machine. Chaque partition a le même nombre d'éléments.

Pour un dataframe contenant des éléments qui sont des couples (*k, v*), la fonction *rdd.partitionBy(n, f)* partitionne des données en *n* partitions. La fonction *f* appliquée sur la clé *k* d'un élément retourne son numéro de partition.

Question 1 : **Regroupement** (2pts). Soit la requête G :

G = GenreFilm.groupBy('genre').agg(count('nF').alias('nb')). Le schéma de G est (genre, nb)

On traite cette requête sans utiliser la méthode groupBy mais avec les instructions suivantes :

G1 = GenreFilm.rdd.mapPartition(regroupementPartiel).toDF(['genre', 'L'])
G2 = G1.rdd.partitionBy(3, lambda genre: genre%n).toDF(['genre', 'L']) → transfert de données entre partitions
G3 = G2.rdd.mapPartition(regroupementFinal).toDF(['genre', 'n'])

a) Décrire en une phrase ce que fait *regroupementPartiel*. Donner son code.

Description : compter le nb de films pour chaque genre

```
def regroupementPartiel(iterateur) :
    D = {}
    for nF, genre in iterateur
        if genre not in D
            D[genre] = 1
        else:
            D[genre] += 1
    for genre, nb in D :
        yield (genre, nb)
```

b) Décrire en une phrase ce que fait *regroupementFinal*.

Description : somme des nb pour chaque genre

```
def regroupementFinal(iterateur) :
    D = {}
    for genre, nb in iterateur:
```

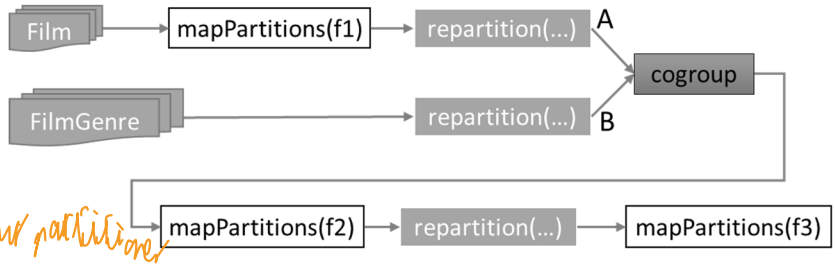
```
if genre not in D:
    D[genre] = nb
else:
    D[genre] += nb
```

→

```
for genre, n in D:
    yield(genre, n)
```

Question 2 (2pts): Soit M la requête qui affiche pour chaque genre, l'année min et max des films de ce genre.

Son schéma est M(genre, minA, maxA). On étudie l'évaluation distribuée de M selon ce plan :



Répondre en français **sans** code python.

a) On applique une fonction *f1* sur les partitions de Film. Décrire *f1*.

f1 : projeter sur (nF, année)

Puis on repartitionne le résultat de f1 ainsi que FilmGenre. Quel est l'attribut de partitionnement ?

repartitionnement par nF

Cogroup associe 2 à 2 les partitions obtenues précédemment (notées A et B sur la figure): cela forme les paires (i^{ème} partition de A, i^{ème} partition de B) sur lesquelles on applique une fonction *f2*. Décrire *f2*.

cogroup puis f2

toutes les tuples de Ai (nF, année) et Bi (nF, genre)

jointure entre Ai et Bi sur nF → (nF, année, genre)

regrouper par genre → (genre, min A, max A)

On repartitionne le résultat de *f2*. Quel est l'attribut de partitionnement ?

Repartitionner f2 par genre

On applique une fonction *f3*, la décrire.

f3 : itérer sur les (genre, minA, maxA)

→ regrouper par genre, et on cherche min des minA, max des maxA

→ (genre, min minA, max maxA)