



Base de Données – TP 3

Esigelec – Département TIC

Février 2021



Contenu de la séance

- Réaliser des traitements sur des groupes de lignes,
- Réaliser des sous-requêtes.

Le partitionnement

- Créer des groupes de lignes, et appliquer une fonction sur ces groupes.
- Par exemple : Compter le nombre de films par genre,
 - Pour cela on regroupe les films par genre,
 - Et on compte le nombre de films par groupe.

Le partitionnement

- Pour faire un regroupement, on utilise **GROUP BY**

```
mysql> SELECT fil_genre AS "Genre", COUNT(*) AS "Nombre de films"  
-> FROM Film  
-> GROUP BY fil_genre;
```

Genre	Nombre de films
Aventure	5
Drame	1
Romance	2
Science-fiction	2

4 rows in set (0.00 sec)

Le partitionnement

- Sélection de groupes (**HAVING**) : permet de sélectionner les groupes vérifiant une condition
- Si nécessaire, revoir le TP 1 pour exprimer une condition,
- Attention :
 - On utilise **WHERE** pour faire une sélection **avant** regroupement,
 - On utilise **HAVING** pour une sélection après,
- Exemple : Calculer le nombre de films par genre pour les groupes comportant au moins 2 valeurs,

Le partitionnement

- Exemple : Calculer le nombre de films par genre pour les groupes comportant au moins 2 valeurs

```
mysql> SELECT fil_genre AS "Genre", COUNT(*) AS "Nombre de films"
-> FROM Film
-> GROUP BY fil_genre
-> HAVING COUNT(*) >= 2;
```

Genre	Nombre de films
Aventure	5
Romance	2
Science-fiction	2

3 rows in set (0.01 sec)

Le partitionnement

- Rappels sur les fonctions utilisables :
 - **COUNT** : Compte le nombre de lignes dans le regroupement
 - **AVG** : Calcule la moyenne pour chaque groupe,
 - **MIN** : Calcule le minimum dans le groupe,
 - **MAX** : Calcule le maximum dans le groupe.

Le partitionnement

- Une erreur à éviter : afficher une colonne différente de celle utilisée pour le regroupement,
- C'est possible de le faire en MySQL, mais le résultat n'a pas de sens.

Le partitionnement

■ Exemple d'erreur :

```
mysql> SELECT fil_sortie AS "Année de sortie", COUNT(*) AS "Nombre de films"
-> FROM Film
-> GROUP BY fil_genre;
```

Année de sortie	Nombre de films
1981	5
1957	1
1939	2
1977	2

4 rows in set (0.00 sec)

- L'année de sortie correspond ici à l'année d'une des lignes du groupe (la première).

La forme générale d'une requête utilisant une fonction de groupe

```
■ SELECT    colonne, fonction_g  
FROM      table  
[WHERE    condition]  
[GROUP BY colonne]  
[HAVING condition]  
[ORDER BY colonne];
```

Les sous-interrogations

- Exemple : Quels sont les films qui ont le même genre que « Les aventuriers de l'Arche perdue » ?
 - On commence par chercher le genre du film :

```
SELECT fil_genre
```

```
FROM Film
```

```
WHERE fil_titre = 'Les aventuriers de  
l''arche perdue';
```

Les sous-interrogations

- La requête donne le résultat :

```
mysql> SELECT fil_genre
-> FROM Film
-> WHERE fil_titre = 'Les aventuriers de l''arche perdue';
+-----+
| fil_genre |
+-----+
| Aventure  |
+-----+
1 row in set (0.00 sec)
```

- Le résultat va être utilisé dans la requête principale, la sous-requête est écrite entre parenthèses.

Les sous-interrogations

- La requête s'écrit :

```
SELECT fil_titre, fil_sortie
FROM Film
WHERE fil_genre =
    (SELECT fil_genre
     FROM Film
     WHERE fil_titre = 'Les aventuriers de
                        l''arche perdue');
```

Les sous-interrogations

```
mysql> SELECT fil_titre AS Titre, fil_sortie AS "Année de sortie"  
-> FROM Film  
-> WHERE fil_genre =  
->       (SELECT fil_genre  
->         FROM Film  
->         WHERE fil_titre = 'Les aventuriers de l''arche perdue');
```

Titre	Année de sortie
Les Aventuriers de l'arche perdue	1981
Indiana Jones et le Temple maudit	1984
Indiana Jones et la Dernière Croisade	1989
King Kong	2005
King Kong	1933

```
5 rows in set (0.00 sec)
```

Les sous-interrogations

- Les sous-interrogations sont entre parenthèses
- On peut placer une sous-interrogation dans les clauses :
 - **WHERE** (nous nous limiterons à ce cas),
 - **FROM**,
 - **HAVING**.

Les sous-interrogations

- Il y a 2 types de sous-interrogations :
 - Les sous interrogations qui donne **une seule** ligne (comme dans l'exemple précédent), elles sont utilisées avec des opérateurs mono-lignes,
 - Les sous interrogations qui donnent **plusieurs** lignes, utilisées avec des opérateurs multi-lignes.

Les sous interrogations mono-lignes

- Il s'agit des opérateurs pour faire une comparaison avec un élément :

☐ =

☐ < , <=

☐ > , >=

☐ <>

- Une erreur se produit si la sous-requête retourne plus d'une ligne

Les sous interrogations mono-lignes

■ Exemple d'erreur :

```
mysql> SELECT fil_genre
-> FROM Film
-> WHERE fil_titre = 'King Kong';
+-----+
| fil_genre |
+-----+
| Aventure  |
| Aventure  |
+-----+
2 rows in set (0.00 sec)

mysql> SELECT fil_titre AS Titre, fil_sortie AS "Année de sortie"
-> FROM Film
-> WHERE fil_genre =
->       (SELECT fil_genre
->         FROM Film
->         WHERE fil_titre = 'King Kong');
ERROR 1242 (21000): Subquery returns more than 1 row
```

Les sous-interrogations donnant plusieurs lignes

- Il s'agit ici d'opérateurs pour faire une comparaison sur un ensemble d'éléments :
 - **IN, NOT IN**
 - **ANY**
 - **ALL**
- Remarque : ANY et ALL sont donnés ici à titre indicatif, ils ne seront vus que l'année prochaine.

Les sous-interrogations donnant plusieurs lignes

- Exemple avec **IN** : afficher les films où joue l'acteur n°3 ?
 - Quels sont les identifiants des films où joue l'acteur 3 :

```
mysql> SELECT jou_film
-> FROM Joue_dans
-> WHERE jou_acteur = 3;
```

jou_film
3
5
6
8

Ensemble de 4 valeurs

4 rows in set (0.00 sec)

Les sous-interrogations donnant plusieurs lignes

- Exemple avec **IN** : afficher les films où joue l'acteur n°3 ?

```
mysql> SELECT fil_titre AS Titre, fil_sortie AS "Année de sortie"  
-> FROM Film  
-> WHERE fil_id IN  
->       (SELECT jou_film  
->         FROM Joue_dans  
->         WHERE jou_acteur = 3);
```

Titre	Année de sortie
Star Wars	1977
Les Aventuriers de l'arche perdue	1981
Indiana Jones et le Temple maudit	1984
King Kong	2005

4 rows in set (0.00 sec)

Utilisation de MySQL (rappel)

- Sous Linux : Téléchargez *cinema.sql* (sur ENT)
- Dans un terminal :
 - ☐ Allez dans le dossier où est *cinema.sql* (cd ...)
 - ☐ Écrivez : `mysql -u esigelec -p`
 - ☐ Mot de passe : `esigelec`
 - ☐ Écrivez : `source cinema.sql`
- Conseils pour faire les requêtes demandées :
 - ☐ Utilisez un éditeur (*gedit* par exemple),
 - ☐ Enregistrez le document avec l'extension `.sql`,
 - ☐ Écrivez vos requêtes une par une dans l'éditeur,
 - ☐ Et testez les en faisant un copier/coller,
 - ☐ Si nécessaire, corrigez dans l'éditeur et testez de nouveau.