



Base de Données – TP 1

Esigelec – Département TIC

Janvier 2021

Introduction

- Pourquoi faire un module de « base de données » en deuxième année ?
 - Suite logique de ce qui est fait en TP (plusieurs fichiers structurés à manipuler), même si nous ne verrons pas l'utilisation en C,
 - Les élèves de CPGE font maintenant des bases de données,
 - Le module d'informatique du premier semestre de la première année du cycle ingénieur prend en compte le fait que les élèves auront déjà fait des bases de données.
- Objectifs : À partir d'une base de données existante, être en mesure de réaliser des requêtes :
 - Simples,
 - Avec jointures,
 - Avec regroupement et fonctions de groupe.

Introduction

- Trois séances sont prévues, chacune comporte une partie de cours et une partie de TP,
- La partie TP consiste à réaliser la liste des requêtes qui vous sera proposée,
- À la fin de chaque TP, vous aurez **une interrogation** portant sur la séance qui vient de se dérouler (ainsi que sur les séances précédentes),
- La somme des trois interrogations formera la note de la partie base de données.

Généralités

- En TP de C vous avez été amenés à utiliser des structures ainsi que des fichiers où des données structurées étaient enregistrées,
- En pratique, on constate :
 - Qu'à partir de 3 ou 4 fichiers, il peut être difficile de gérer les données,
 - Qu'il est fastidieux de gérer les recherches, ajouts et suppressions dans les fichiers,
 - Qu'il peut y avoir des problèmes pour gérer l'accès aux données par plusieurs utilisateurs en même temps,
 - Etc...
- **La solution** : utiliser un *Système de Gestion de Bases de Données* (SGBD).

Généralités

- Il existe plusieurs SGBD connus :
 - Oracle
 - MySQL
 - PostgreSQL
 - ...
- Ils utilisent tous (avec des variantes) un même langage : le SQL (*Structured Query Language*).
- Ce langage va permettre :
 - De créer les bases de données,
 - D'ajouter, modifier, supprimer des données,
 - D'interroger une base de données pour obtenir les informations que l'on souhaite.

Généralités

- Exemple : Gérer des données relatives à des films.
- En C nous aurions les **structures** suivantes :
 - ☐ Acteur (nom, prenom, date de naissance,...)
 - ☐ Film (titre, duree, annee, ...)
 - ☐ Realisateur (nom, prénom, ...)
 - ☐ Studio (nom, adresse, ...)
- Problème : Comment matérialiser qu'un Film est réalisé par un Réalisateur, et produit par un Studio ?
 - ☐ Copier le Réalisateur et le Studio dans Film ?
- Solution :
 - ☐ On ajoute un identifiant (nombre entier unique) dans Réalisateur et dans Studio,
 - ☐ On ajoute ces 2 identifiants dans Film.

Généralités

- Nous obtenons :

- ☐ Acteur (nom, prenom, date de naissance,...)
- ☐ Film (titre, duree, annee, ..., [id_realisateur](#), [id_studio](#))
- ☐ Realisateur ([id_realisateur](#), nom, prenom, ...)
- ☐ Studio ([id_studio](#), nom, adresse, ...)

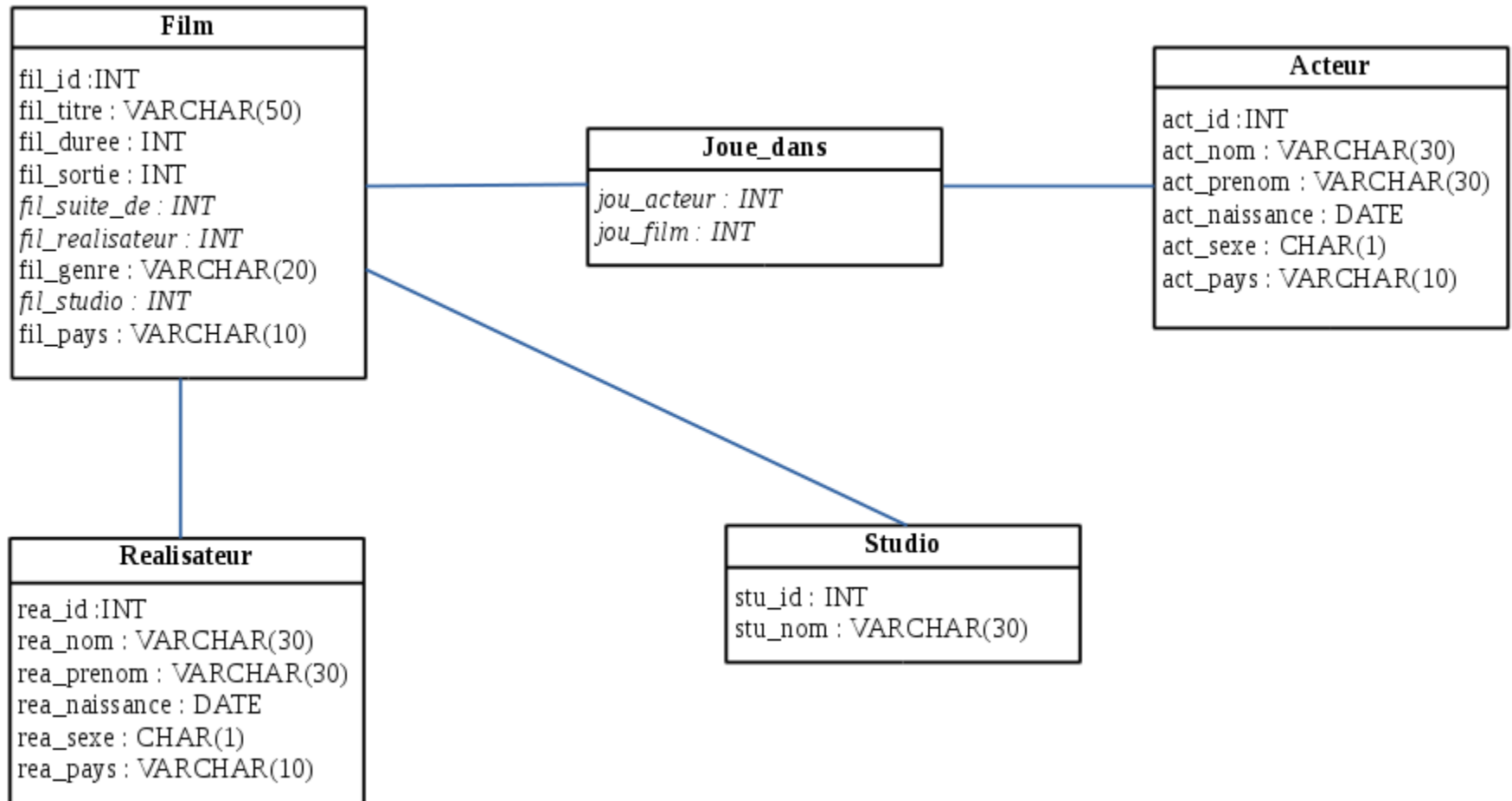
- Problème : Comment indiquer qu'un Acteur peut jouer dans plusieurs Films, et que dans un Film il peut y avoir plusieurs Acteurs ?

- ☐ Utilisation de tableaux ? Très mauvaise solution !
- ☐ On ajoute un identifiant dans Acteur et un dans Film,
- ☐ On crée une nouvelle structure composée de ces 2 identifiants.

Généralités

- Nous obtenons finalement les structures :
 - Acteur (`id_acteur`, nom, prenom, date de naissance,...)
 - Film (`id_film`, titre, duree, annee, ..., id_realisateur, id_studio)
 - Realisateur (id_realisateur, nom, prénom, ...)
 - Studio (id_studio, nom, adresse, ...)
 - `Joue_dans(id_acteur, id_film)` // Nouvelle structure
- En base de données, nous allons avoir les mêmes structures,
- Nous utiliserons le terme **table**,
- Comme en C, chaque attribut va avoir un type en fonction des valeurs qu'il peut contenir.

Schéma de la base « cinema »



Lecture d'une table

- Lister le contenu de la table « Acteur »

- `SELECT *`
`FROM Acteur;`

- Majuscules / Minuscules ?

- De préférence des majuscules pour le SQL,
 - Et des minuscules pour le reste,

- **Attention : respectez le nom des tables !**

Lecture d'une table

■ Exemple :

```
mysql> SELECT *  
      -> FROM Acteur;
```

| act_id | act_nom | act_prenom | act_naissance | act_sexe |
|--------|-----------|------------|---------------|----------|
| 1 | FONDA | Henry | 1905-05-16 | M |
| 2 | LEIGH | Viven | 1913-11-05 | F |
| 3 | FORD | Harrison | 1942-07-13 | M |
| 4 | HAMIL | Mark | 1951-09-25 | M |
| 5 | WINSLET | Kate | 1975-10-05 | F |
| 6 | DI CAPRIO | Leonardo | 1974-11-11 | M |
| 7 | CONNERY | Sean | 1930-08-30 | M |

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



La valeur NULL

- Pas de valeur affectée
- Peut signifier :
 - Valeur inconnue
 - Valeur inapplicable
- Est différente de :
 - 0
 - Caractère espace
 - Chaîne vide

La valeur NULL

- Exemple de table contenant des valeurs à *NULL* :

```
mysql> SELECT *  
-> FROM Film;
```

| fil_id | fil_titre | fil_duree | fil_sortie | fil_suite_de | fil_realisateur | fil_genre | fil_studio |
|--------|---|-----------|------------|--------------|-----------------|-----------------|------------|
| 1 | 12 hommes en colère | 96 | 1957 | NULL | 3 | Drame | NULL |
| 2 | Autant en emporte le vent | 224 | 1939 | NULL | NULL | Romance | NULL |
| 3 | Star Wars | 121 | 1977 | NULL | 6 | Science-fiction | NULL |
| 4 | Titanic | 194 | 1997 | NULL | 1 | Romance | 1 |
| 5 | Les Aventuriers de l'arche perdue | 115 | 1981 | NULL | 4 | Aventure | 3 |
| 6 | Indiana Jones et le Temple maudit | 118 | 1984 | 5 | 4 | Aventure | 3 |
| 7 | Indiana Jones et la Dernière Croisade | 127 | 1989 | 6 | 4 | Aventure | 3 |
| 8 | King Kong | 188 | 2005 | NULL | NULL | Aventure | NULL |
| 9 | King Kong | 105 | 1933 | NULL | NULL | Aventure | NULL |
| 10 | L'Empire contre-attaque | 124 | 1980 | 3 | 6 | Science-fiction | NULL |
| 11 | Indiana Jones et le Royaume du crâne de cristal | 123 | 2008 | 7 | 4 | Aventure | 3 |

11 rows in set (0.00 sec)

Valeurs inapplicables

Valeurs inconnues

Projection

- Exemple : Afficher uniquement les colonnes *fil_titre* et *fil_genre* de tous les films :

- ```
SELECT fil_titre, fil_genre
FROM Film;
```

- Attention, la projection peut générer des doublons.

# Projection

```
mysql> SELECT fil_titre, fil_genre
-> FROM Film;
```

| fil_titre                                       | fil_genre       |
|-------------------------------------------------|-----------------|
| 12 hommes en colère                             | Drame           |
| Autant en emporte le vent                       | Romance         |
| Star Wars                                       | Science-fiction |
| Titanic                                         | Romance         |
| Les Aventuriers de l'arche perdue               | Aventure        |
| Indiana Jones et le Temple maudit               | Aventure        |
| Indiana Jones et la Dernière Croisade           | Aventure        |
| King Kong                                       | Aventure        |
| King Kong                                       | Aventure        |
| L'Empire contre-attaque                         | Science-fiction |
| Indiana Jones et le Royaume du crâne de cristal | Aventure        |

11 rows in set (0.00 sec)

# Suppression des doublons

- Le mot clef : *DISTINCT*

```
mysql> SELECT fil_genre
-> FROM Film;
+-----+
| fil_genre |
+-----+
| Drame |
| Romance |
| Science-fiction |
| Romance |
| Aventure |
| Aventure |
| Aventure |
| Aventure |
| Aventure |
| Science-fiction |
| Aventure |
+-----+
11 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT fil_genre
-> FROM Film;
+-----+
| fil_genre |
+-----+
| Drame |
| Romance |
| Science-fiction |
| Aventure |
+-----+
4 rows in set (0.00 sec)
```



# Sélection

- Afficher les films de type *Aventure* :
- ***SELECT \****  
***FROM Film***  
***WHERE fil\_genre = 'Aventure';***
- Remarque : La sélection n'est pas sensible à la casse pour les chaînes de caractères.

# Sélection

```
mysql> SELECT *
-> FROM Film
-> WHERE fil_genre = 'Aventure';
```

| fil_id | fil_titre                                       | fil_duree | fil_sortie | fil_suite_de | fil_realisateur | fil_genre | fil_studio |
|--------|-------------------------------------------------|-----------|------------|--------------|-----------------|-----------|------------|
| 5      | Les Aventuriers de l'arche perdue               | 115       | 1981       | NULL         | 4               | Aventure  | 3          |
| 6      | Indiana Jones et le Temple maudit               | 118       | 1984       | 5            | 4               | Aventure  | 3          |
| 7      | Indiana Jones et la Dernière Croisade           | 127       | 1989       | 6            | 4               | Aventure  | 3          |
| 8      | King Kong                                       | 188       | 2005       | NULL         | NULL            | Aventure  | NULL       |
| 9      | King Kong                                       | 105       | 1933       | NULL         | NULL            | Aventure  | NULL       |
| 11     | Indiana Jones et le Royaume du crâne de cristal | 123       | 2008       | 7            | 4               | Aventure  | 3          |

6 rows in set (0.00 sec)

# Sélection

- On retrouve les opérateurs de comparaison classiques :

| Opérateurs | Significations               |
|------------|------------------------------|
| =          | Égal                         |
| <, <=      | Inférieur, inférieur ou égal |
| >, >=      | Supérieur, supérieur ou égal |
| != ou <>   | Différent                    |

- Et les opérateurs logiques :

| Opérateurs | Significations |
|------------|----------------|
| AND ou &&  | Et             |
| OR ou      | Ou             |
| NOT ou !   | Non            |

# Sélection

- Exemple : « Afficher la liste des films sortis avant 1980 et dont la durée est au moins égale à 100 minutes. »
- ```
SELECT *  
FROM Film  
WHERE fil_sortie < 1980 AND fil_duree >= 100;
```

```
+-----+-----+-----+-----+-----+-----+-----+-----+  
| fil_id | fil_titre           | fil_duree | fil_sortie | fil_suite_de | fil_realisateur | fil_genre   | fil_studio |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
| 2 | Autant en emporte le vent | 224 | 1939 | NULL | NULL | Romance | NULL |  
| 3 | Star Wars              | 121 | 1977 | NULL | 6 | Science-fiction | NULL |  
| 9 | King Kong              | 105 | 1933 | NULL | NULL | Aventure | NULL |  
+-----+-----+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

Sélection

- Cas particulier pour la valeur « NULL »

| Opérateurs | Significations |
|-------------|------------------------|
| IS NULL | La valeur est NULL |
| IS NOT NULL | La valeur est non NULL |

- Exemple : liste des réalisateurs dont on ne connaît pas la date de naissance :
 - ```
SELECT *
FROM Realisateur
WHERE rea_naissance IS NULL;
```

# La forme standard d'une requête avec sélection :

- **SELECT** *col1, col2, ...*  
**FROM** *table*  
**WHERE** *condition;*
- Pour une meilleure lisibilité, nous vous demandons de **toujours** écrire vos requêtes sur plusieurs lignes.

# Trier les résultats

- ```
SELECT col1, col2, ... coln
FROM table
[WHERE condition]
ORDER BY col1 [ASC|DESC],
           col2 [ASC|DESC],
           ...
           coln [ASC|DESC];
```
- Remarque : Si rien n'est précisé, **ASC** est la valeur par défaut.

Trier les résultats

- Exemple : Liste des Acteurs par ordre alphabétique du nom :

```
SELECT *  
FROM Acteur  
ORDER BY act_nom;
```

- Le tri est bien dans l'ordre alphabétique car ASC est la valeur par défaut.

Trier les résultats

```
mysql> SELECT *  
-> FROM Acteur  
-> ORDER BY act_nom;  
+-----+-----+-----+-----+-----+  
| act_id | act_nom  | act_prenom | act_naissance | act_sexe |  
+-----+-----+-----+-----+-----+  
|      7 | CONNERY | Sean       | 1930-08-30    | M        |  
|      6 | DI CAPRIO | Leonardo   | 1974-11-11    | M        |  
|      1 | FONDA    | Henry      | 1905-05-16    | M        |  
|      3 | FORD     | Harrison   | 1942-07-13    | M        |  
|      4 | HAMIL    | Mark       | 1951-09-25    | M        |  
|      2 | LEIGH    | Viven     | 1913-11-05    | F        |  
|      5 | WINSLET  | Kate       | 1975-10-05    | F        |  
+-----+-----+-----+-----+-----+  
7 rows in set (0.00 sec)
```

Opérateur LIKE

- Permet la comparaison d'une colonne contenant des chaînes de caractères avec un motif :
 - <colonne> **LIKE** <motif>
 - <colonne> **NOT LIKE** <motif>

- Les motifs peuvent être constitués à l'aide des symboles % et _ ou tout autre caractère:
 - % : remplace n'importe quelle chaîne de caractères, éventuellement vide,
 - _ : remplace un et un seul caractère,
 - Les autres caractères sont interprétés tel quel (y compris l'espace).

Opérateur LIKE

- Exemple : Afficher le titre, la date de sortie et la durée des films dont le titre contient le mot « le » :

```
mysql> SELECT fil_titre, fil_sortie, fil_duree  
-> FROM Film  
-> WHERE fil_titre LIKE '% le %';
```

| fil_titre | fil_sortie | fil_duree |
|---|------------|-----------|
| Autant en emporte le vent | 1939 | 224 |
| Indiana Jones et le Temple maudit | 1984 | 118 |
| Indiana Jones et le Royaume du crâne de cristal | 2008 | 123 |

3 rows in set (0.00 sec)

Renommer les colonnes

- Pour afficher des noms de colonnes différents on utilise :

- AS <nouveau nom>

- Exemple :

```
SELECT fil_titre AS Titre, fil_sortie AS  
       "Année de sortie"  
FROM Film;
```

Renommer les colonnes

```
mysql> SELECT fil_titre AS Titre, fil_sortie AS "Année de sortie"  
-> FROM Film;
```

| Titre | Année de sortie |
|---|-----------------|
| 12 hommes en colère | 1957 |
| Autant en emporte le vent | 1939 |
| Star Wars | 1977 |
| Titanic | 1997 |
| 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 |
| L'Empire contre-attaque | 1980 |
| Indiana Jones et le Royaume du crâne de cristal | 2008 |

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

Expressions dans les requêtes

- Des expressions arithmétiques, des opérations sur les chaînes de caractères, etc. peuvent être utilisées dans la clause SELECT

| Opérateurs | Significations |
|----------------------|---|
| +, -, *, / | Opérateurs arithmétiques classiques |
| DIV et MOD | Division entière et Modulo |
| ROUND(nb, dec) | Arrondi de <i>nb</i> à <i>dec</i> décimales |
| CONCAT(ch1, ch2,...) | Concaténation de chaînes |
| UPPER, LOWER | Mise en majuscules ou minuscules |

Expressions dans les requêtes

- Exemple :

```
mysql> SELECT fil_titre AS titre, ROUND(fil_duree/60, 1) AS Durée  
-> FROM Film;
```

| titre | Durée |
|---|-------|
| 12 hommes en colère | 1.6 |
| Autant en emporte le vent | 3.7 |
| Star Wars | 2.0 |
| Titanic | 3.2 |
| Les Aventuriers de l'arche perdue | 1.9 |
| Indiana Jones et le Temple maudit | 2.0 |
| Indiana Jones et la Dernière Croisade | 2.1 |
| King Kong | 3.1 |
| King Kong | 1.8 |
| L'Empire contre-attaque | 2.1 |
| Indiana Jones et le Royaume du crâne de cristal | 2.1 |

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

Fonctions de groupe

- Fonctions qui s'appliquent sur une colonne entière :

| Opérateurs | Significations |
|------------|----------------------------|
| MIN | Minimum de la colonne |
| MAX | Maximum de la colonne |
| COUNT | Compte le nombre de lignes |
| AVG | Moyenne de la colonne |

Fonctions de groupe

- Exemple :

```
mysql> SELECT ROUND(AVG(fil_duree), 2) AS "Durée moyenne"
-> FROM Film;
+-----+
| Durée moyenne |
+-----+
|          139.55 |
+-----+
1 row in set (0.00 sec)
```

Union

- Union (au sens ensembliste) de 2 tables contenant le même nombre de colonnes :

■ T1

| <i>A</i> | <i>B</i> |
|----------|----------|
| 1 | 2 |
| 3 | 4 |

■ T2

| <i>A</i> | <i>B</i> |
|----------|----------|
| 3 | 4 |
| 5 | 6 |

■ T1 UNION T2

| <i>A</i> | <i>B</i> |
|----------|----------|
| 1 | 2 |
| 3 | 4 |
| 5 | 6 |

Union

■ Exemple :

```
mysql> SELECT act_nom AS "Nom", act_prenom AS "Prénom"  
-> FROM Acteur  
-> UNION  
-> SELECT rea_nom, rea_prenom  
-> FROM Realisateur;
```

| Nom | Prénom |
|-----------|----------|
| FONDA | Henry |
| LEIGH | Viven |
| FORD | Harrison |
| HAMIL | Mark |
| WINSLET | Kate |
| DI CAPRIO | Leonardo |
| CONNERY | Sean |
| CAMERON | James |
| LANDAU | Jon |
| LUMET | Sidney |
| SPIELBERG | Steven |
| DILLER | Barry |
| LUCAS | George |

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

Utilisation de MySQL en TP

- Sous Linux : Téléchargez *cinema.sql* (sur ENT)
- Dans un terminal :
 - ☐ Allez dans le dossier où est *cinema.sql* (cd ...)
 - ☐ Écrivez : `mysql -u esigelec -p`
 - ☐ Saisissez ensuite le 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.