

Khennous
Moad
1SIO - B

MISSION 10 : BASE DE DONNÉES

Objectif 1 :

On commence par créer une base de donnée qui se nomme ‘cinéma’ , ensuite nous pouvons commencer à créer les tables de cette bdd.

Pour créer une table il suffit d'aller dans la bdd créer et cliquer sur nouvelle table qui se trouve à gauche de l'écran sur l'interface phpMyAdmin, puis voila un screen de comment remplir créer une table avec son nom , ses attributs (avec le type et ses fonctions d'intégrations) :

| Name | Type | Length/Values | Default | Collation | Attributes | Null | Index | A_I | Comments | Virtuality | Move column |
|----------------|---------|---------------|---------|-----------|------------|---------|-------|-------------------------------------|----------|------------|-------------|
| IDENT_ACTEUR | INT | | None | | | PRIMARY | | <input checked="" type="checkbox"/> | | | |
| NOM | VARCHAR | 50 | None | | | | --- | <input type="checkbox"/> | | | |
| PRENOM | VARCHAR | 50 | None | | | | --- | <input type="checkbox"/> | | | |
| DATE_NAISSANCE | DATE | | None | | | | --- | <input type="checkbox"/> | | | |
| NB_FILM | INT | | None | | | | --- | <input type="checkbox"/> | | | |

Objectif 2 :

En observant la table film, j'ai constaté que le champ PAYS stockait directement le nom du pays, alors que la table pays contient déjà cette information via le champ LIBELLÉ. Cela créait une redondance et un risque d'incohérence (ex. écritures différentes du même pays).

Pour corriger cela, j'ai remplacé la valeur textuelle du pays par une clé étrangère pointant vers la table pays.

Dans phpMyAdmin, j'ai modifié la colonne PAYS dans l'onglet Structure : je l'ai renommée en IDENT_PAYS et changée en type INT.

Ensuite, dans Vue relationnelle, j'ai lié ce champ à pays(IDENT_PAYS) pour créer la contrainte référentielle.

Ainsi, chaque film est désormais associé à un pays existant sans duplication de données.

Objectif 3 :

Pour insérer des enregistrements dans l'une des tables il suffit d'aller dans la table ou l'on veut insérer les enregistrements et cliquer sur INSERT qui se trouve en haut de l'écran sur l'interface phpMyAdmin, puis voilà un screen de comment insérer des enregistrements dans une table :

The screenshot shows the 'Insert' interface in phpMyAdmin. At the top, there are tabs: Browse, Structure, SQL, Search, Insert (which is selected), Export, Import, Privileges, and More. Below the tabs is a table with columns: Column, Type, Function, Null, and Value. The rows represent fields for a new record:

| Column | Type | Function | Null | Value |
|----------------|-------------|----------|--------------------------|------------|
| IDENT_ACTEUR | int | | | 1 |
| NOM | varchar(50) | | | DURIS |
| PRENOM | varchar(50) | | | Romain |
| DATE_NAISSANCE | date | | | 1974-05-28 |
| NB_FILM | int | | <input type="checkbox"/> | 60 |

At the bottom right of the form is a 'Go' button.

Objectif 4 :

Pour interroger l'une des tables il suffit d'aller dans la table à interroger et cliquer sur SEARCH qui se trouve en haut de l'écran sur l'interface phpMyAdmin, puis voilà un screen de comment interroger une table ainsi que le résultat :

1. Afficher la liste des films (ID et Titre) qui ont un genre SF.

The screenshot shows the phpMyAdmin interface for the 'film' table. The top navigation bar includes tabs for Server, localhost, Database: cinema, Table: film, Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, and Triggers. Below the navigation is a toolbar with Table search, Zoom search, and Find and replace buttons.

The main area displays a search query builder titled "Do a 'query by example' (wildcard: "%")". It lists columns: IDENT_FILM (int), TITRE (varchar(100)), GENRE1 (varchar(50)), GENRE2 (varchar(50)), DATE_SORTIE (date), IDENT_PAYS (int), IDENT_REALISATEUR (int), DISTRIBUTEUR (varchar(50)), and DUREE (int). The GENRE2 field has a value of "SF" and an operator of "LIKE %...%".

Below the query builder, there are sections for "Extra options" (checkbox for DISTINCT), "Select columns (at least one)" (list of columns: IDENT_FILM, TITRE, GENRE1, GENRE2, DATE_SORTIE, IDENT_PAYS, IDENT_REALISATEUR, DISTRIBUTEUR, DUREE), and "Or Add search conditions (body of the "where" clause)" (empty input field).

Screenshot of the MySQL Workbench interface showing the results of a query on the 'film' table.

Query results:

```
SELECT DISTINCT `IDENT_FILM`, `TITRE` FROM `film` WHERE `GENRE2` LIKE "%SF%"
```

Table view:

| | IDENT_FILM | TITRE |
|--------------------------|------------|---------------------------------|
| <input type="checkbox"/> | 3 | STAR WARS 6 : LE RETOUR DU JEDI |
| <input type="checkbox"/> | 4 | AVATAR |

2. Afficher la liste des réalisateurs (Nom, Prénom) qui sont nés dans les années 50.

Screenshot of the MySQL Workbench interface showing the search dialog for finding directors born in the 1950s.

Search criteria:

- Column: IDENT_REALISATEUR Type: int Operator: = Value: (empty)
- Column: NOM Type: varchar(50) Collation: utf8mb4_general_ci Operator: LIKE %...% Value: (empty)
- Column: PRENOM Type: varchar(50) Collation: utf8mb4_general_ci Operator: LIKE %...% Value: (empty)
- Column: DATE_NAISSANCE Type: date Operator: BETWEEN Value: 1950-01-01, 1950-12-31
- Column: NB_FILM_ECRIT Type: int Operator: = Value: (empty)
- Column: NB_FILM_PRODUIT Type: int Operator: = Value: (empty)

Select columns (at least one):

- IDENT_REALISATEUR
- NOM
- PRENOM**
- DATE_NAISSANCE
- NB_FILM_ECRIT
- NB_FILM_PRODUIT

Show search criteria

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0004 seconds.)

```
SELECT `NOM`, `PRENOM` FROM `realisateur` WHERE `DATE_NAISSANCE`  
BETWEEN '1950-01-01' AND '1950-12-31'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]



3. Afficher le nom et le prénom des acteurs qui ont joué dans un film réalisé par Georges Lucas.

Do a "query by example" (wildcard: "%")

| Column | Type | Collation | Operator | Value |
|----------------|-------------|--------------------|------------|-------|
| IDENT_ACTEUR | int | | IN (...) | 6,7 |
| NOM | varchar(50) | utf8mb4_general_ci | LIKE %...% | |
| PRENOM | varchar(50) | utf8mb4_general_ci | LIKE %...% | |
| DATE_NAISSANCE | date | | = | |
| NB_FILM | int | | = | |

Select columns (at least one):

IDENT_ACTEUR
NOM
PRENOM
DATE_NAISSANCE
NB_FILM

The screenshot shows the phpMyAdmin interface. At the top, a green bar indicates "Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)". Below it, the SQL query "SELECT `NOM`, `PRENOM` FROM `acteur` WHERE `IDENT_ACTEUR` IN (6,7)" is displayed. To the right of the query are several buttons: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. Below the query, there are filters for Show all (unchecked), Number of rows (set to 25), Filter rows (Search this table), and Sort by key (set to None). A "Extra options" button is also present. The main area displays a table with columns NOM and PRENOM. The data rows are FORD Harrison and FISHER Carrie.

| NOM | PRENOM |
|--------|----------|
| FORD | Harrison |
| FISHER | Carrie |

Objectif 5 :

Pour interroger l'une des tables avec des requêtes SQL il suffit d'aller dans la table à interroger et cliquer sur SQL qui se trouve en haut de l'écran sur l'interface phpMyAdmin, puis voila un screen de comment interroger une table avec du SQL ainsi que le résultat :

4. Afficher la liste des films (ID et Titre) qui ont un genre SF.

```
SELECT TITRE, DUREE, DATE_SORTIE  
FROM film  
WHERE DUREE > 90  
AND DATE_SORTIE BETWEEN '2008-01-01' AND '2012-12-31';
```

Showing rows 0 - 0 (1 total, Query took 0.0004 seconds.)

```
SELECT TITRE, DUREE, DATE_SORTIE FROM film WHERE DUREE > 90 AND DATE_SORTIE BETWEEN '2008-01-01' AND '2012-12-31';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

| TITRE | DUREE | DATE_SORTIE |
|--------|-------|-------------|
| AVATAR | 170 | 2009-12-16 |

5. Afficher la liste des acteurs qui ont tourné entre 40 et 90 films

```
SELECT NOM, PRENOM, NB_FILM
FROM acteur
WHERE NB_FILM BETWEEN 40 AND 90;
```

Showing rows 0 - 4 (5 total, Query took 0.0003 seconds.)

```
SELECT NOM, PRENOM, NB_FILM FROM acteur WHERE NB_FILM BETWEEN 40 AND 90;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Extra options

| NOM | PRENOM | NB_FILM |
|--------|-----------|---------|
| DURIS | Romain | 60 |
| FORD | Harrison | 64 |
| FISHER | Carrie | 74 |
| WEAVER | Sigourney | 66 |
| RENO | Jean | 75 |

6. Un nouveau film est ajouté au catalogue

----- insérer ce réalisateur vu qu'il n'est pas dans la table -----

```
INSERT INTO realisateur (NOM, PRENOM, DATE_NAISSANCE,
NB_FILM_ECRIT, NB_FILM_PRODUIT)
```

```
VALUES ('BOON', 'DANY', '1966-06-26', 10, 5);
```

```
INSERT INTO film (TITRE, GENRE1, DATE_SORTIE, IDENT_PAYS,
IDENT_REALISATEUR, DISTRIBUTEUR, DUREE)
```

```
VALUES ('Bienvenue chez les Ch'tis', 'Comédie', '2008-02-27', 1, 4,
```

'Gaumont', 100);

| | | IDENT_FILM | TITRE | GENRE1 | GENRE2 | DATE_SORTIE | IDENT_PAYS | IDENT_REALISATEUR | DISTRIBUTEUR | DUREE |
|--------------------------|--|------------|---------------------------------|----------|------------|-------------|------------|-------------------|------------------|-------|
| <input type="checkbox"/> |  Edit  Copy  Delete | 1 | SUBWAY | POLICIER | DRAME | 1985-04-10 | 1 | 1 | GAUMONT | 104 |
| <input type="checkbox"/> |  Edit  Copy  Delete | 2 | NIKITA | DRAME | ROMANTIQUE | 1990-02-21 | 1 | 1 | GAUMONT | 118 |
| <input type="checkbox"/> |  Edit  Copy  Delete | 3 | STAR WARS 6 : LE RETOUR DU JEDI | ACTION | SF | 1983-10-19 | 2 | 2 | 20th Century Fox | 133 |
| <input type="checkbox"/> |  Edit  Copy  Delete | 4 | AVATAR | ACTION | SF | 2009-12-16 | 2 | 3 | 20th Century Fox | 170 |
| <input type="checkbox"/> |  Edit  Copy  Delete | 5 | Bienvenue chez les Ch'tis | Comédie | NULL | 2008-02-27 | 1 | 4 | Gaumont | 100 |