

## 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 ) :

The screenshot shows the phpMyAdmin interface for a database named 'cinéma'. The 'Structure' tab is selected, and the table 'acteurs' is being edited. The table has 1 column(s). The columns are defined as follows:

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

## Objectif 2 :

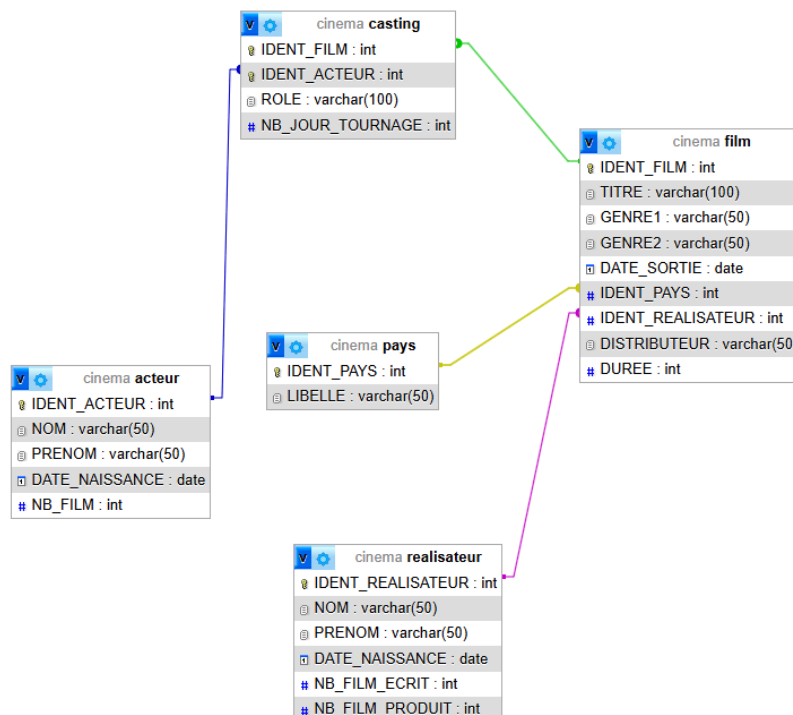
En observant la table film, j'ai constaté que le champ PAYS stockait directement le nom du pays, alors que la table pays contient déjà cette information via le champ LIBELLÉ. Cela créait une redondance et un risque d'incohérence (ex. écritures différentes du même pays).

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 r 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 voila un screen de comment insérer des enregistrements dans une table :

	Browse		Structure		SQL		Search		Insert		Export		Import		Privileges		More
Column	Type	Function	Null	Value													
IDENT_ACTEUR	int	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="1"/>													
NOM	varchar(50)	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="DURIS"/>													
PRENOM	varchar(50)	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="Romain"/>													
DATE_NAISSANCE	date	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="1974-05-28"/>													
NB_FILM	int	<input type="text"/>	<input type="checkbox"/>	<input type="text" value="60"/>													

Go

## 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 voila 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.

Server: localhost » Database: cinema » Table: film

Table search   Zoom search   Find and replace

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

Column	Type	Collation	Operator	Value
IDENT_FILM	int		=	
TITRE	varchar(100)	utf8mb4_general_ci	LIKE %...%	
GENRE1	varchar(50)	utf8mb4_general_ci	LIKE %...%	
GENRE2	varchar(50)	utf8mb4_general_ci	LIKE %...%	SF
DATE_SORTIE	date		=	
IDENT_PAYS	int		=	
IDENT_REALISATEUR	int		=	
DISTRIBUTEUR	varchar(50)	utf8mb4_general_ci	LIKE %...%	
DUREE	int		=	

Extra options

Select columns (at least one):

- IDENT\_FILM
- TITRE
- GENRE1
- GENRE2
- DATE\_SORTIE
- IDENT\_PAYS
- IDENT\_REALISATEUR
- DISTRIBUTEUR
- DUREE

☒ DISTINCT

Or Add search conditions (body of the "where" clause):

Server: localhost Database: cinema Table: film

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Table search Zoom search Find and replace

Show search criteria

Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

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

☐ 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

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

☐ Check all With selected: ☐ Edit ☐ Copy ☐ Delete ☐ Export

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

Table search Zoom search Find and replace

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

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

Extra options

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		=	

Extra options

Select columns (at least one):

IDENT\_ACTEUR  
NOM  
PRENOM  
DATE\_NAISSANCE  
NB\_FILM

✓ Showing rows 0 - 1 (2 total, Query took 0.0004 seconds.)

```
SELECT `NOM`, `PRENOM` FROM `acteur` WHERE `IDENT_ACTEUR` IN (6,7)
```

☐ Profiling [ [Edit inline](#) ] [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP code](#) ] [ [Refresh](#) ]

☐ Show all | Number of rows: 25 ▼ | Filter rows:  | Sort by key: None ▼

Extra options

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\u00e2tis', 'Comédie', '2008-02-27', 1, 4,
```



'Gaumont', 100);

← T →		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	GAUMONT	104
<input type="checkbox"/>	 Edit	 Copy	 Delete	2	NIKITA	DRAME	ROMANTIQUE	1990-02-21	1	GAUMONT	118
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	STAR WARS 6 : LE RETOUR DU JEDI	ACTION	SF	1983-10-19	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