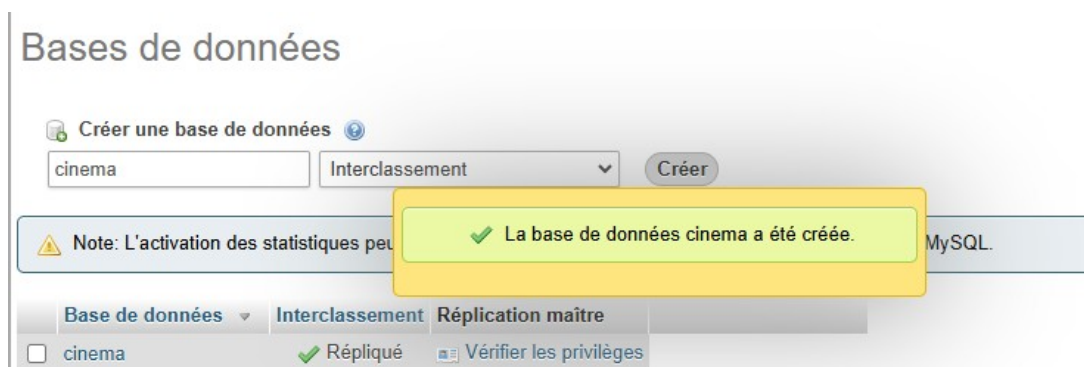


Mission 6 – donnée le 18/10/24
Réalisée du 10/11/2024 à 23h30 au 11/11/2024
Fin : 11/11/2024 3h27

Par Aliyah FALGUEYRETTES
1SIO-B, groupe 1
Responsable – M. Cafford

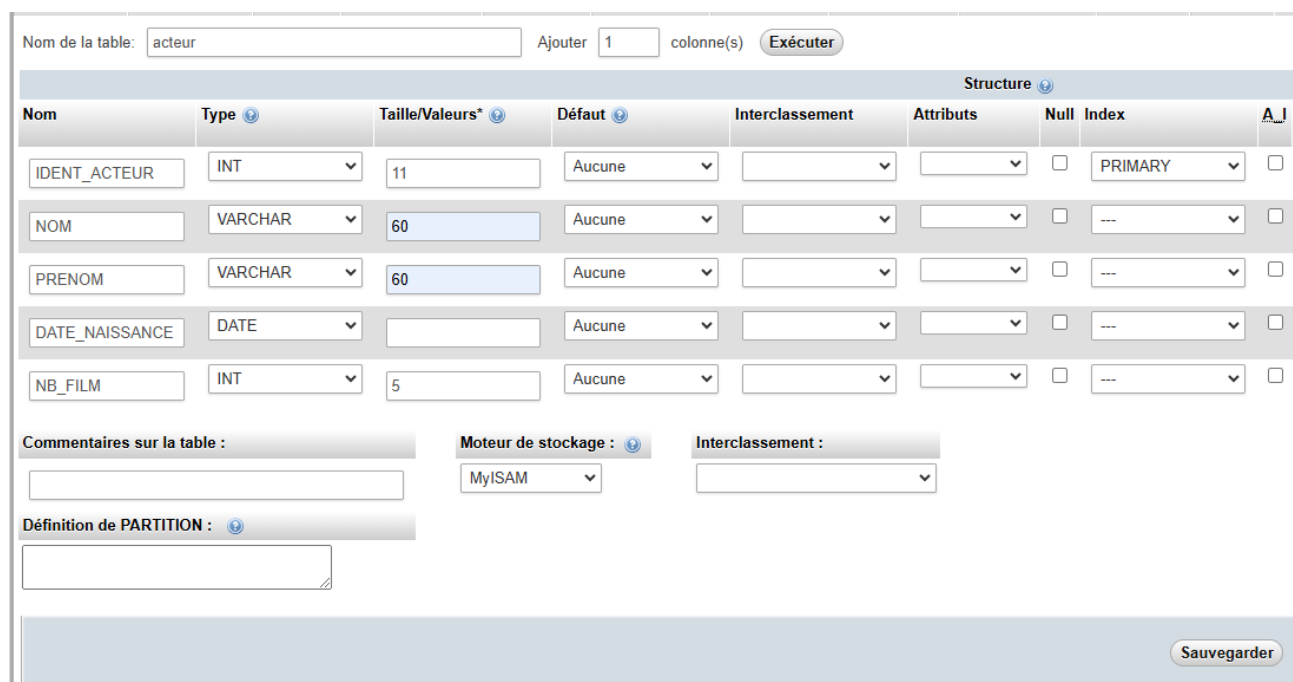
Objectif 1 : création de la base de données cinema

Tout d'abord, l'on crée la base de données qui contiendra les tables.



Puis l'on initialise chaque table, en lui associant ses champs, et en précisant celui qui servira de clef primaire.

Exemple de création de la table « acteur » :



Nom	Type	Taille/Valeurs*	Défaut	Interclassement	Attributs	Null	Index
IDENT_ACTEUR	INT	11	Aucune			<input type="checkbox"/>	PRIMARY
NOM	VARCHAR	60	Aucune			<input type="checkbox"/>	---
PRENOM	VARCHAR	60	Aucune			<input type="checkbox"/>	---
DATE_NAISSANCE	DATE		Aucune			<input type="checkbox"/>	---
NB_FILM	INT	5	Aucune			<input type="checkbox"/>	---

– initialisation de la table –

Ici, le champ qui a servi de clef primaire est le champ « IDENT_ACTEUR », réputé unique. La plupart du temps, les champs *identifiants* seront désignés comme étant clef primaire de leur table d'origine, de sorte à plus tard permettre les contraintes d'intégrité. Dans ce premier objectif, la clef primaire est à retrouver à l'aide des noms des champs dans certaines tables ; les autres champs *identifiants* serviront plus tard à faire respecter des contraintes d'intégrité.

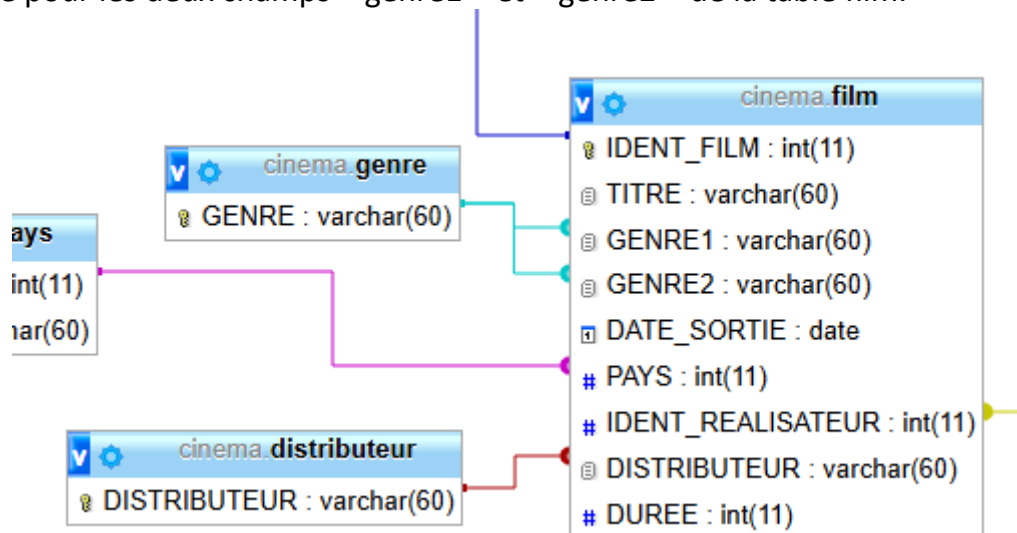
Objectif 2 – Contraintes d'intégrité

Contrainte de domaine – Un bon *typage* des champs assure la contrainte de domaine.

Contrainte d'unicité – Pour éviter les redondances de la table film, je crée deux tables : la table « distributeur » et la table « genre ». En effet, sans cela, une faute de frappe pourrait empêcher de retrouver, par exemple, tous les éléments dont le genre est la science-fiction ; ou dont le distributeur est Gaumont (cf annexe). Pour éviter cela, il convient de « fermer » le choix de l'utilisateur qui voudrait ajouter une entrée, à des genres et des distributeurs déjà prédéfinis.

Pour être certains que de telles relations entre les tables soient créées, il convient de vérifier les types de chaque champ (ces derniers devant correspondre, au risque de provoquer une erreur ou une modification involontaire).

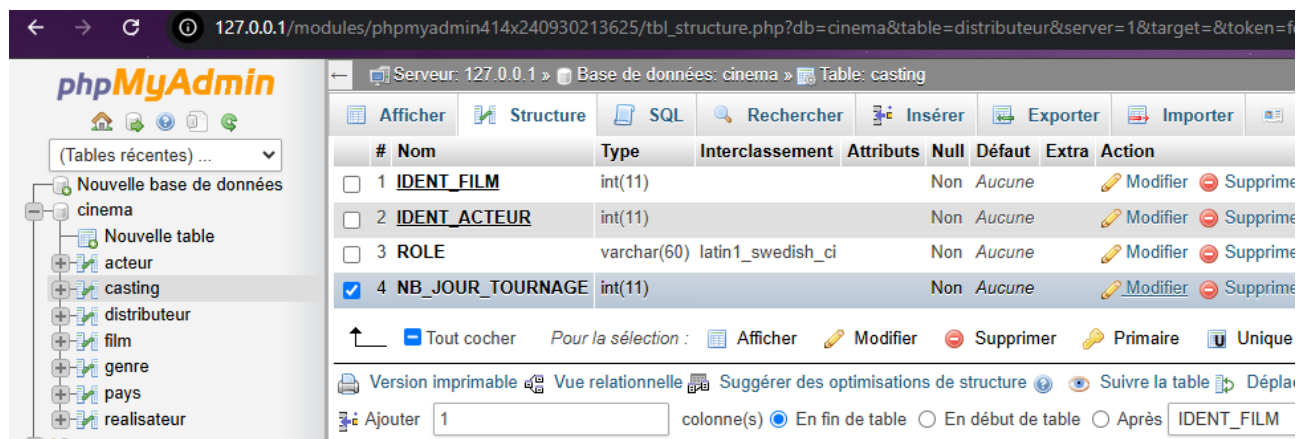
Dans le schéma ci-dessous, le champ « genre » de la table du même nom sert de clef étrangère pour les deux champs « genre1 » et « genre2 » de la table film.



Contrainte de non-nullité – Par défaut, les tables déclarées sont considérées obligatoires. Il est donc nécessaire de préciser quels sont les champs facultatifs à l'aide de l'option « NULL », qui est activée manuellement à la création d'une table. Comme cela n'a pas été fait au cours du processus de création, il est désormais nécessaire de modifier les tables concernées.

Pour modifier les champs d'une table sur *phpMyAdmin*, il faut d'abord se rendre dans le sous-menu de la table elle-même, puis cliquer sur l'onglet Structure.

Les champs de la table seront affichés comme ci-dessous :



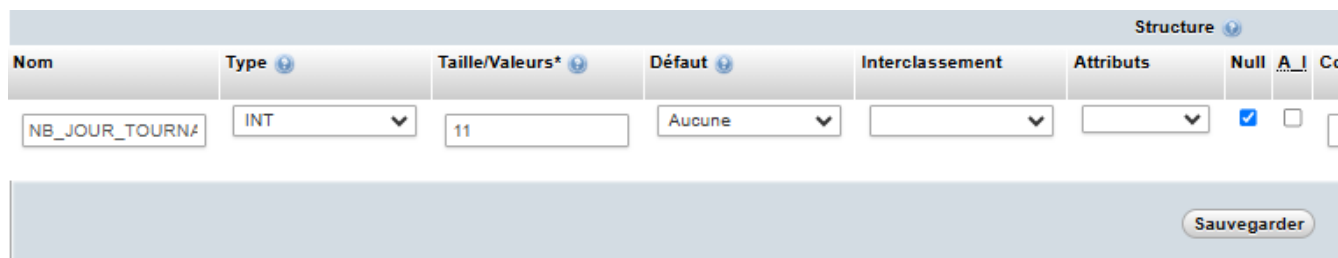
The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of the database 'cinema' containing tables: casting, distributeur, film, genre, pays, and realisateur. The main area displays the 'Structure' tab for the 'casting' table. It shows a table with 4 columns:

#	Nom	Type	Interclassement	Attributs	Null	Défaut	Extra	Action
<input type="checkbox"/>	1 IDENT_FILM	int(11)			Non	Aucune		Modifier Supprimer
<input type="checkbox"/>	2 IDENT_ACTEUR	int(11)			Non	Aucune		Modifier Supprimer
<input type="checkbox"/>	3 ROLE	varchar(60) latin1_swedish_ci			Non	Aucune		Modifier Supprimer
<input checked="" type="checkbox"/>	4 NB_JOUR_TOURNAGE	int(11)			Non	Aucune		Modifier Supprimer

Below the table, there are options for selection: 'Tout cocher', 'Pour la sélection :', and buttons for 'Afficher', 'Modifier', 'Supprimer', 'Primaire', and 'Unique'. At the bottom, there is a 'Ajouter' section with a text input '1' and radio buttons for 'En fin de table' (selected), 'En début de table', and 'Après', followed by a dropdown menu showing 'IDENT_FILM'.

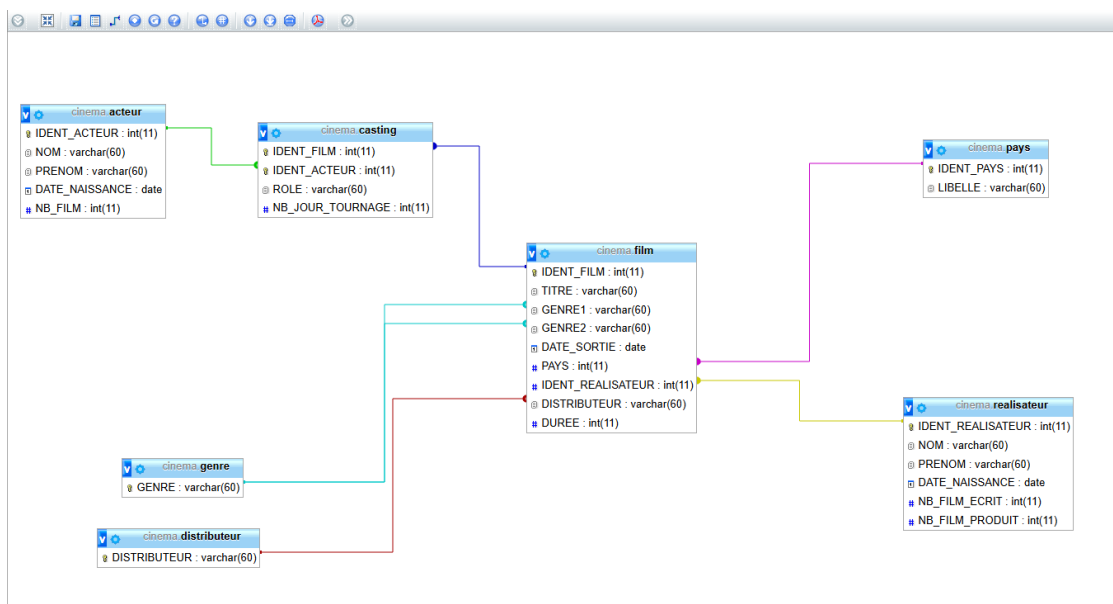
L'on peut cliquer sur l'élément Modifier sur le champ en question, ou sélectionner un ou plusieurs champs et cliquer sur Modifier en dessous de ces champs, à gauche de la phrase *Pour la sélection*.

Enfin, l'on effectue la modification souhaitée (ici, l'on coche la case NULL pour indiquer que le champ modifié deviendra facultatif) et l'on sauvegarde en cliquant sur l'icône correspondante.



This is a close-up of the 'Structure' tab in phpMyAdmin. It shows the configuration for the 'NB_JOUR_TOURNAGE' field. The fields are organized into columns: 'Nom', 'Type', 'Taille/Valeurs*', 'Défaut', 'Interclassement', 'Attributs', 'Null', and 'A.I.'. The 'NB_JOUR_TOURNAGE' row is highlighted. The 'Type' is 'INT', 'Taille/Valeurs*' is '11', 'Défaut' is 'Aucune', and 'Null' is checked (indicated by a blue checkmark in the checkbox). At the bottom right, there is a 'Sauvegarder' button.

Contrainte d'intégrité référentielle – Relations. Chaque relation lie l'un des champs qui fait référence à la clef primaire d'une autre table, à cette dernière.



Contrainte de normalité – A priori, chaque table a pour clef primaire son champ déterminant.

Objectif 3 – Insertion de valeurs selon l'annexe 1

Pour insérer des valeurs dans une table sur *phpMyAdmin*, après avoir vérifié que les contraintes d'intégrité ont bien été respectées, l'on clique sur le sous-menu qui correspond à la table à remplir. Parmi les onglets qui apparaissent, l'on cliquera sur « Insérer ».

phpMyAdmin

Tables récentes ...

Nouvelle base de données cinema

Nouvelle table

- acteur
- casting
- distributeur
- film
- genre
- pays
- realisateur
- orange

Table: acteur

Colonne	Type	Fonction	Null	Valeur
IDENT_ACTEUR	int(11)			1
NOM	varchar(60)			DURIS
PRENOM	varchar(60)			Romain
DATE_NAISSANCE	date			28/05/1974
NB_FILM	int(11)			60

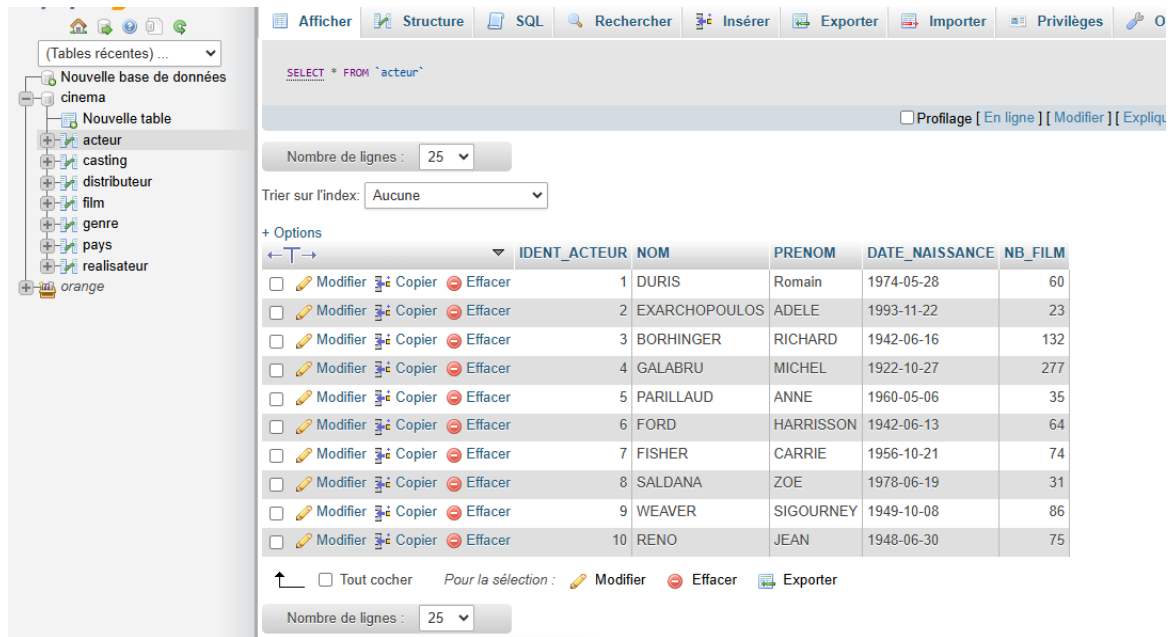
Exécuter

☐ Ignorer

Colonne	Type	Fonction	Null	Valeur
IDENT_ACTEUR	int(11)			2
NOM	varchar(60)			EXARCHOPOULOS
PRENOM	varchar(60)			ADELE
DATE_NAISSANCE	date			22/11/1993
NB_FILM	int(11)			23

Enfin, l'on insérera les valeurs dont il est sujet, avant d'appuyer sur l'icône « Exécuter ».

Et bien sûr, il est nécessaire de vérifier l'exactitude de ces informations.



SELECT * FROM `acteur`

Nombre de lignes : 25

Trier sur l'index: Aucune

+ Options

	IDENT_ACTEUR	NOM	PRENOM	DATE_NAISSANCE	NB_FILM
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	1	DURIS	Romain	1974-05-28	60
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	2	EXARCHOPOULOS	ADELE	1993-11-22	23
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	3	BORHINGER	RICHARD	1942-06-16	132
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	4	GALABRU	MICHEL	1922-10-27	277
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	5	PARILLAUD	ANNE	1960-05-06	35
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	6	FORD	HARRISSON	1942-06-13	64
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	7	FISHER	CARRIE	1956-10-21	74
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	8	SALDANA	ZOE	1978-06-19	31
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	9	WEAVER	SIGOURNEY	1949-10-08	86
<input type="checkbox"/> Modifier <input type="checkbox"/> Copier <input type="checkbox"/> Effacer	10	RENO	JEAN	1948-06-30	75

☐ Tout cocher Pour la sélection : ☐ Modifier ☐ Effacer ☐ Exporter

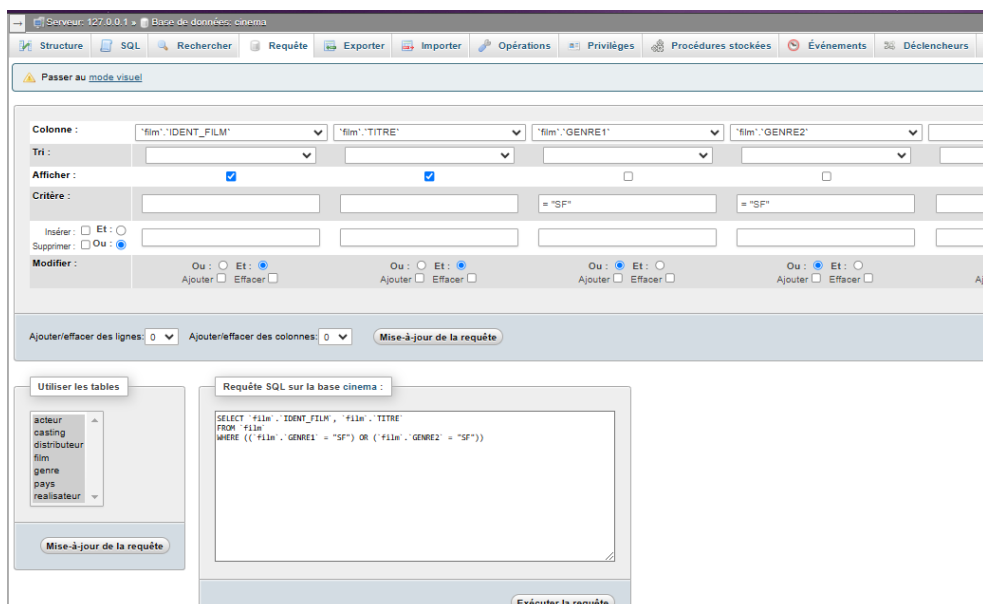
Nombre de lignes : 25

Objectif 4 – Requêtes

Pour effectuer une requête sous *phpMyAdmin*, l'on doit revenir sur le menu de la base de données (en cliquant, dans ce cas présent, sur le menu à gauche « cinema »). Arrivés dans ce menu, l'onglet Requête apparaîtra ; c'est là que la suite des événements se déroulera.

Première requête – L'on cherche à obtenir tous les noms et identifiants de films dont l'un des genres est la science-fiction.

L'on affiche donc les enregistrements de films et leur identifiant, si leur champ « genre1 » ou « genre2 » vaut « SF ».



Structure SQL Rechercher Requête Exporter Importer Opérations Privileges Procédures stockées Événements Déclencheurs

Passer au mode visual

Colonne : 'film'.IDENT_FILM 'film'.TITRE 'film'.GENRE1 'film'.GENRE2

Tri : [valeur] [valeur] [valeur] [valeur]

Afficher : ☒ ☒ ☐ ☐

Critère : [valeur] [valeur] = "SF" = "SF"

Insérer : ☐ Et : ☐ Supprimer : ☐ Ou : ☒

Modifier : [valeur] [valeur] [valeur] [valeur] Ou : ☐ Et : ☒ Ajouter ☐ Effacer ☐ Ou : ☐ Et : ☒ Ajouter ☐ Effacer ☐ Ou : ☐ Et : ☐ Ajouter ☐ Effacer ☐ Ou : ☐ Et : ☐ Ajouter ☐ Effacer ☐

Ajouter/effacer des lignes : 0 Ajouter/effacer des colonnes : 0 Mise-à-jour de la requête

Utiliser les tables

Requête SQL sur la base cinema :

```
SELECT `film`.`IDENT_FILM`, `film`.`TITRE`
FROM `film`
WHERE ((`film`.`GENRE1` = "SF") OR (`film`.`GENRE2` = "SF"))
```

Mise-à-jour de la requête

Exécuter la requête

Cette requête offre le résultat attendu :

The screenshot shows a database query interface. At the top, a SQL query is displayed: `SELECT 'film'.IDENT_FILM, 'film'.TITRE FROM 'film' WHERE (('film'.GENRE1 = "SF") OR ('film'.GENRE2 = "SF"))`. Below the query, there are controls for the number of lines (set to 25) and a sort index (set to 'Aucune'). A table of results is shown with columns 'IDENT_FILM' and 'TITRE'. The results are:

IDENT_FILM	TITRE
3	STAR WARS 6 : LE RETOUR DU JEDI
4	AVATAR

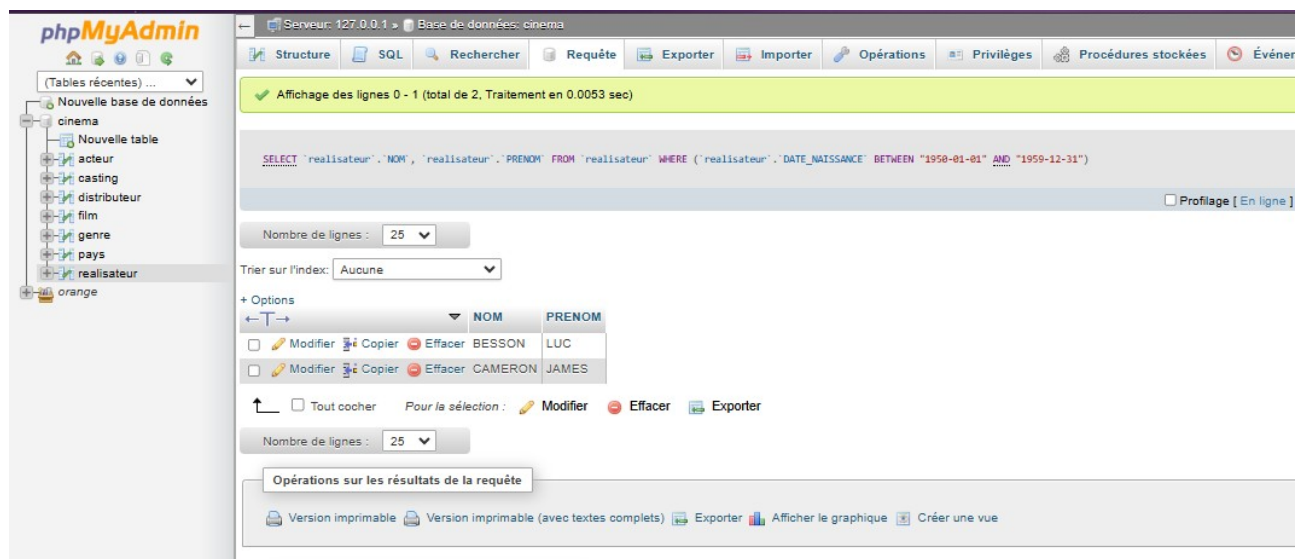
Below the table, there are buttons for 'Modifier', 'Copier', and 'Effacer' for each row. At the bottom, there are buttons for 'Version imprimable', 'Version imprimable (avec textes complets)', 'Exporter', 'Afficher le graphique', and 'Créer une vue'. There is also a section for saving the query as a bookmark with a title field and a checkbox for 'Signet visible pour les autres utilisateurs'.

Seconde requête – La seconde requête demande à afficher les noms et prénoms des réalisateurs nés dans les années 50.

Pour cela, l'on recherche donc les noms et prénoms des réalisateurs dont la date de naissance est comprise entre le 01/01/1950 et le 31/12/1959.

The screenshot shows a database query interface with a query builder. The query builder has fields for 'Colonne', 'Tri', 'Afficher', 'Critère', and 'Modifier'. The 'Colonne' field is set to 'realisateur'. The 'Tri' field is set to 'NOM'. The 'Afficher' field is set to 'PRENOM'. The 'Critère' field is set to 'DATE_NAISSANCE'. The 'Modifier' field is set to 'BETWEEN "1950-01-01" AND "1959-12-31"'. The query builder also has buttons for 'Ajouter/effacer des lignes', 'Ajouter/effacer des colonnes', and 'Mise-à-jour de la requête'. Below the query builder, there is a section for 'Utiliser les tables' with a list of tables: 'acteur', 'casting', 'distributeur', 'film', 'genre', 'pays', and 'realisateur'. The 'Requête SQL sur la base cinema' section shows the resulting SQL query: `SELECT 'realisateur'.NOM, 'realisateur'.PRENOM FROM 'realisateur' WHERE ('realisateur'.DATE_NAISSANCE BETWEEN "1950-01-01" AND "1959-12-31")`. At the bottom, there is a button for 'Exécuter la requête'.

Ce qui nous affiche le résultat suivant :



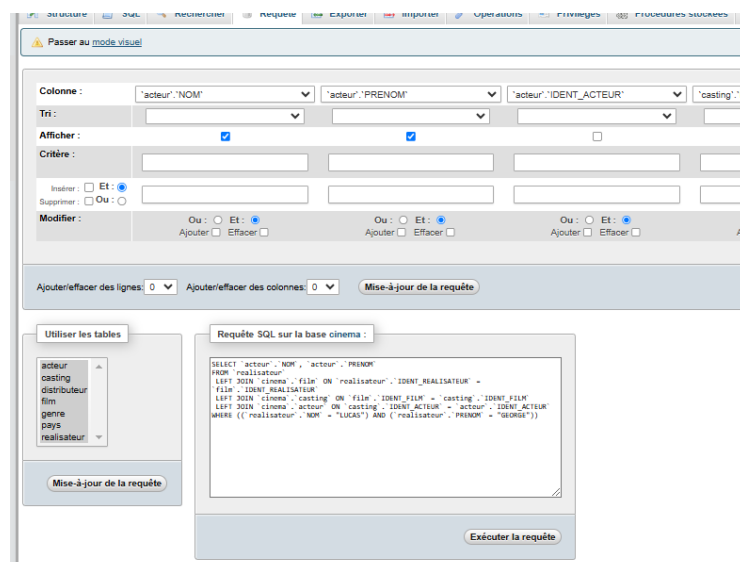
Qui correspond bien à notre demande.

Troisième requête – Il est demandé d’afficher le nom et prénom des acteurs qui ont joué dans un film réalisé par George Lucas.

Cette requête est particulière car elle demande d’utiliser spécifiquement les noms et prénoms du réalisateur, ce qui implique davantage de conditions dans la requête, et d’aller chercher l’identifiant de George Lucas.

Elle fera intervenir la table « film », pour répertorier les films de George Lucas ; la table acteurs, pour récupérer leurs noms et prénoms ; la table casting, qui fait le lien entre ces deux dernières.

Il est nécessaire de faire une jointure entre toutes ces tables afin de faire le lien entre le nom et prénom du réalisateur, ses films, l’identifiant de ses films et les acteurs.





Après avoir effectué cette grande requête, on obtient ceci :

```
SELECT `acteur`.`NOM`, `acteur`.`PRENOM` FROM `realisateur` LEFT JOIN `cinema`.`film` ON `realisateur`.`IDENT_REALI:
`casting`.`IDENT_ACTEUR` = `acteur`.`IDENT_ACTEUR` WHERE ((`realisateur`.`NOM` = "LUCAS") AND (`realisateur`.`PRENO
```

Nombre de lignes : 25 ▼

+ Options

				NOM	PRENOM
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	FORD	HARRISSON
<input type="checkbox"/>	 Modifier	 Copier	 Effacer	FISHER	CARRIE

↑ ☐ Tout cocher Pour la sélection :  Modifier  Effacer  Exporter

Nombre de lignes : 25 ▼

Opérations sur les résultats de la requête

 Version imprimable  Version imprimable (avec textes complets)  Exporter  Afficher le graphique 