

Devoir - MySQL : Gestion d'une Plateforme de Streaming

Vous êtes chargé de concevoir une base de données pour une plateforme de streaming. Cette plateforme gère des utilisateurs, des films, des abonnements, et des historiques de visionnage. Vous devez modéliser les tables, insérer des données, et exécuter les requêtes SQL demandées.

Partie 1 : Modèle Logique des Données (MLD)

1. Table : Utilisateur

- **id_utilisateur** : Identifiant unique de l'utilisateur (INT, PK, AUTO_INCREMENT)
- **nom** : Nom de l'utilisateur (VARCHAR(255))
- **email** : Adresse email de l'utilisateur (VARCHAR(255), UNIQUE)
- **date_inscription** : Date d'inscription de l'utilisateur (DATE)

```
✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0008 seconde(s).)

CREATE TABLE Utilisateur ( id_utilisateur INT AUTO_INCREMENT PRIMARY KEY, nom VARCHAR(255), email VARCHAR(255) UNIQUE, date_inscription DATE );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]
```

2. Table : Film

- **id_film** : Identifiant unique du film (INT, PK, AUTO_INCREMENT)
- **titre** : Titre du film (VARCHAR(255))
- **genre** : Genre du film (VARCHAR(100))
- **annee_sortie** : Année de sortie du film (YEAR)
- **duree** : Durée du film en minutes (INT)

```
✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0021 seconde(s).)

CREATE TABLE Film ( id_film INT AUTO_INCREMENT PRIMARY KEY, titre VARCHAR(255), genre VARCHAR(100), annee_sortie YEAR, duree INT );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]
```

3. Table : Abonnement

- **id_abonnement** : Identifiant unique de l'abonnement (INT, PK, AUTO_INCREMENT)
- **id_utilisateur** : Référence à l'utilisateur (INT, FK vers Utilisateur.id_utilisateur)

- **type_abonnement** : Type d'abonnement (ENUM : 'Basic', 'Standard', 'Premium')
- **date_debut** : Date de début de l'abonnement (DATE)
- **date_fin** : Date de fin de l'abonnement (DATE)

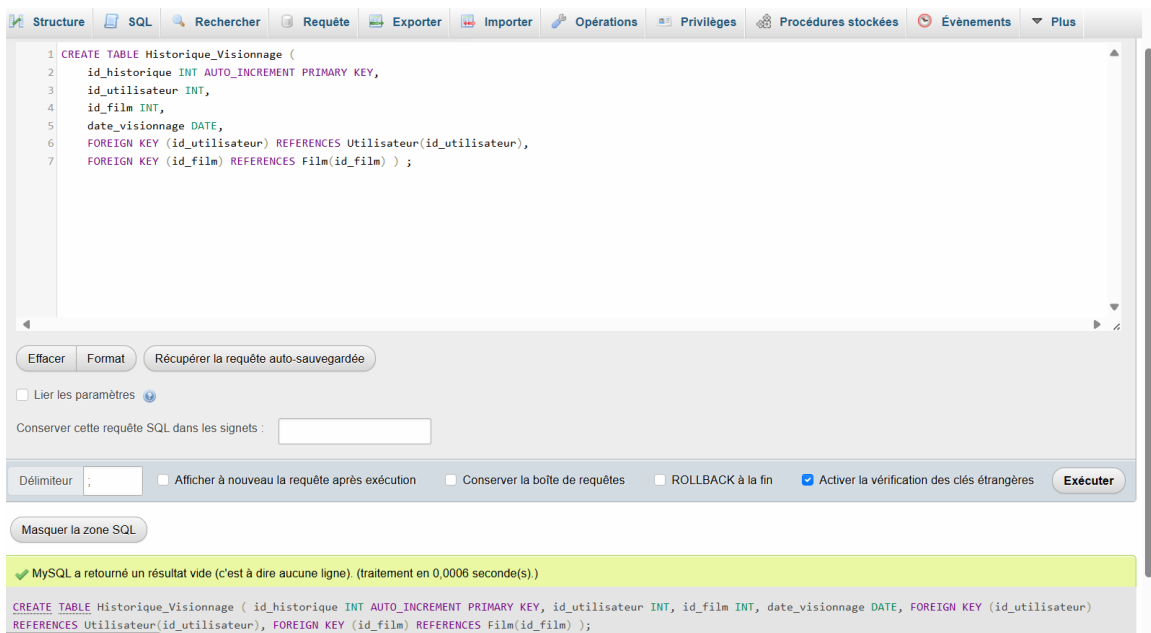
```
✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0009 seconde(s))

CREATE TABLE Abonnement ( id_abonnement INT AUTO_INCREMENT PRIMARY KEY, id_utilisateur INT, type_abonnement ENUM('Basic', 'Standard', 'Premium'), date_debut DATE, date_fin DATE, FOREIGN KEY (id_utilisateur) REFERENCES Utilisateur(id_utilisateur) );

[ Éditer en ligne ] [ Éditer ] [ Créer le code source PHP ]
```

4. Table : **Historique_Visionnage**

- **id_historique** : Identifiant unique de l'historique (INT, PK, AUTO_INCREMENT)
- **id_utilisateur** : Référence à l'utilisateur (INT, FK vers Utilisateur.id_utilisateur)
- **id_film** : Référence au film (INT, FK vers Film.id_film)
- **date_visionnage** : Date du visionnage (DATE)



```
1 CREATE TABLE Historique_Visionnage (
2   id_historique INT AUTO_INCREMENT PRIMARY KEY,
3   id_utilisateur INT,
4   id_film INT,
5   date_visionnage DATE,
6   FOREIGN KEY (id_utilisateur) REFERENCES Utilisateur(id_utilisateur),
7   FOREIGN KEY (id_film) REFERENCES Film(id_film) );
```

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0006 seconde(s))

```
CREATE TABLE Historique_Visionnage ( id_historique INT AUTO_INCREMENT PRIMARY KEY, id_utilisateur INT, id_film INT, date_visionnage DATE, FOREIGN KEY (id_utilisateur) REFERENCES Utilisateur(id_utilisateur), FOREIGN KEY (id_film) REFERENCES Film(id_film) );
```

Partie 2 : Questions

1. Listez tous les utilisateurs inscrits depuis plus d'un an.

Serveur : 127.0.0.1 » Base de données : bd_devoir » Table : Utilisateur

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations

Afficher la zone SQL

✓ Affichage des lignes 0 - 9 (total de 10, traitement en 0,0004 seconde(s).)

SELECT * FROM Utilisateur WHERE date_inscription < CURDATE() - INTERVAL 1 YEAR;

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	id_utilisateur	nom	email	date_inscription
<input type="checkbox"/> Éditer Copier Supprimer	1	Ali Ben Ahmed	ali.benahmed@example.com	2022-05-15
<input type="checkbox"/> Éditer Copier Supprimer	2	Sara Al-Farsi	sara.alfarsi@example.com	2021-07-20
<input type="checkbox"/> Éditer Copier Supprimer	3	Mohammed El-Mansouri	mohammed.elmansouri@example.com	2020-08-25
<input type="checkbox"/> Éditer Copier Supprimer	4	Layla Al-Hakim	layla.alhakim@example.com	2023-01-12
<input type="checkbox"/> Éditer Copier Supprimer	5	Omar Bouselham	omar.bouselham@example.com	2023-06-10
<input type="checkbox"/> Éditer Copier Supprimer	6	Fatima Zahra	fatima.zahra@example.com	2021-03-01
<input type="checkbox"/> Éditer Copier Supprimer	7	Youssef Al-Mahdi	youssef.almadhi@example.com	2022-11-18
<input type="checkbox"/> Éditer Copier Supprimer	8	Khadija Khalil	khadija.khalil@example.com	2019-12-03
<input type="checkbox"/> Éditer Copier Supprimer	9	Rania El-Mahmoud	rania.elmahmoud@example.com	2023-04-27
<input type="checkbox"/> Éditer Copier Supprimer	10	Tariq Ben Ali	tariq.benali@example.com	2021-09-05

☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

2. Affichez les films d'un genre spécifique, triés par année de sortie.

Afficher la zone SQL

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0005 seconde(s).) [annee_sortie: 2020... - 2021...]

SELECT * FROM `film` WHERE genre = 'drame' order by annee_sortie;

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	id_film	titre	genre	annee_sortie	1	duree
<input type="checkbox"/> Éditer Copier Supprimer	1	La Cité du Soleil	Drame	2020		130
<input type="checkbox"/> Éditer Copier Supprimer	12	Le Silence des Cieux	Drame	2020		105
<input type="checkbox"/> Éditer Copier Supprimer	9	L'Envol	Drame	2021		95

☐ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

3. Trouvez les utilisateurs ayant un abonnement 'Premium' dont la date de fin est dans moins d'un mois.

```

1 SELECT *
2 FROM Utilisateur u
3 JOIN Abonnement a ON u.id_utilisateur = a.id_utilisateur
4 WHERE a.type_abonnement = 'Premium'
5 AND a.date_fin <= CURDATE() + INTERVAL 1 MONTH;

```

SELECT * SELECT INSERT UPDATE DELETE Effacer Format Récupérer la requête auto-sauvagée

☐ Lier les paramètres

Conserver cette requête SQL dans les signets :

Délimiteur : ☐ Afficher à nouveau la requête après exécution ☐ Conserver la boîte de requêtes ☐ ROLLBACK à la fin ☒ Activer la vérification des clés étrangères **Exécuter**

Masquer la zone SQL

⚠ La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les cases à cocher ainsi que les liens Éditer, Copier et Supprimer ne sont pas disponibles.

✓ Affichage des lignes 0 - 3 (total de 4, traitement en 0,0004 seconde(s))

```

SELECT * FROM Utilisateur u JOIN Abonnement a ON u.id_utilisateur = a.id_utilisateur WHERE a.type_abonnement = 'Premium' AND a.date_fin <= CURDATE() + INTERVAL 1 MONTH;

```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

utilisateur	nom	email	date_inscription	id_abonnement	id_utilisateur	type_abonnement	date_debut	date_fin
1	Ali Ben Ahmed	ali.benahmed@example.com	2022-05-15	13	1	Premium	2023-01-01	2024-01-01
4	Layla Al-Hakim	layla.alhakim@example.com	2023-01-12	16	4	Premium	2023-03-01	2024-03-01
7	Youssef Al-Mahdi	youssef.almahdi@example.com	2022-11-18	19	7	Premium	2022-09-10	2023-09-10
10	Tariq Ben Ali	tariq.benali@example.com	2021-09-05	22	10	Premium	2022-12-15	2023-12-15

4. Affichez les utilisateurs et les films qu'ils ont visionnés au cours de la dernière semaine.

```

1 SELECT u.nom, f.titre, hv.date_visionnage
2 FROM Historique_Visionnage hv
3 JOIN Utilisateur u ON hv.id_utilisateur = u.id_utilisateur
4 JOIN Film f ON hv.id_film = f.id_film
5 WHERE hv.date_visionnage >= CURDATE() - INTERVAL 7 DAY;

```

SELECT * SELECT INSERT UPDATE DELETE Effacer Format Récupérer la requête auto-sauvagée

☐ Lier les paramètres

Conserver cette requête SQL dans les signets :

Délimiteur : ☐ Afficher à nouveau la requête après exécution ☐ Conserver la boîte de requêtes ☐ ROLLBACK à la fin ☒ Activer la vérification des clés étrangères **Exécuter**

Masquer la zone SQL

⚠ La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les cases à cocher ainsi que les liens Éditer, Copier et Supprimer ne sont pas disponibles.

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0004 seconde(s))

```

SELECT u.nom, f.titre, hv.date_visionnage FROM Historique_Visionnage hv JOIN Utilisateur u ON hv.id_utilisateur = u.id_utilisateur JOIN Film f ON hv.id_film = f.id_film WHERE hv.date_visionnage >= CURDATE() - INTERVAL 7 DAY;

```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

nom	titre	date_visionnage
Ali Ben Ahmed	La Cité du Soleil	2024-12-05

5. Listez les films qui n'ont pas été visionnés par aucun utilisateur.

Afficher la zone SQL

⚠ La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les cases à cocher ainsi que les liens Éditer, Copier et Supprimer ne sont pas disponibles.

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0006 seconde(s))

```

SELECT f.* FROM Film f LEFT JOIN Historique_Visionnage hv ON f.id_film = hv.id_film WHERE hv.id_film IS NULL;

```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

id_film	titre	genre	annee_sortie	duree
10	Le Retour des Dragons	Fantasy	2023	160
11	Les Ombres de Bagdad	Thriller	2022	120
12	Le Silence des Cieux	Drame	2020	105

6. Trouvez les utilisateurs qui ont regardé plus de 4 films depuis leur inscription.

La requête SQL a été exécutée avec succès.

```
SELECT u.nom, COUNT(hv.id_film) AS films_vus FROM Utilisateur u JOIN Historique_Visionnage hv ON u.id_utilisateur = hv.id_utilisateur GROUP BY u.id_utilisateur HAVING COUNT(hv.id_film) > 4;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

nom films_vus
Ali Ben Ahmed 5

7. Affichez tous les abonnements, y compris ceux des utilisateurs non enregistrés dans la table des visionnages.

✓ Affichage des lignes 0 - 9 (total de 10, traitement en 0,0004 seconde(s).)

```
SELECT a.*, u.nom FROM Abonnement a LEFT JOIN Utilisateur u ON a.id_utilisateur = u.id_utilisateur;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

id_abonnement	id_utilisateur	type_abonnement	date_debut	date_fin	nom
13	1	Premium	2023-01-01	2024-01-01	Ali Ben Ahmed
14	2	Standard	2022-06-15	2023-06-15	Sara Al-Farsi
15	3	Basic	2021-05-10	2022-05-10	Mohammed El-Mansouri
16	4	Premium	2023-03-01	2024-03-01	Layla Al-Hakim
17	5	Standard	2023-07-01	2024-07-01	Omar Bouselham
18	6	Basic	2021-11-15	2022-11-15	Fatima Zahra
19	7	Premium	2022-09-10	2023-09-10	Youssef Al-Mahdi
20	8	Standard	2020-08-20	2021-08-20	Khadija Khalil
21	9	Basic	2023-04-01	2024-04-01	Rania El-Mahmoud
22	10	Premium	2022-12-15	2023-12-15	Tariq Ben Ali

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

8. Affichez tous les utilisateurs, même ceux qui n'ont pas d'abonnement.

✓ Affichage des lignes 0 - 9 (total de 10, traitement en 0,0006 seconde(s).)

```
SELECT u.* FROM Utilisateur u LEFT JOIN Abonnement a ON u.id_utilisateur = a.id_utilisateur;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

id_utilisateur	nom	email	date_inscription
1	Ali Ben Ahmed	ali.benahmed@example.com	2022-05-15
2	Sara Al-Farsi	sara.alfarsi@example.com	2021-07-20
3	Mohammed El-Mansouri	mohammed.elmansouri@example.com	2020-08-25
4	Layla Al-Hakim	layla.alhakim@example.com	2023-01-12
5	Omar Bouselham	omar.bouselham@example.com	2023-06-10
6	Fatima Zahra	fatima.zahra@example.com	2021-03-01
7	Youssef Al-Mahdi	youssef.almadhi@example.com	2022-11-18
8	Khadija Khalil	khadija.khalil@example.com	2019-12-03
9	Rania El-Mahmoud	rania.elmahmoud@example.com	2023-04-27
10	Tariq Ben Ali	tariq.benali@example.com	2021-09-05

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

9. Listez les utilisateurs qui ont regardé le même film plusieurs fois.

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations Suivi Déclencheurs

Afficher la zone SQL

La requête SQL a été exécutée avec succès.

```
SELECT hv.id_utilisateur, hv.id_film, COUNT(*) AS nb_visionnages FROM Historique_Visionnage hv GROUP BY hv.id_utilisateur, hv.id_film HAVING COUNT(*) > 1;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

id_utilisateur	id_film	nb_visionnages
1	1	2

Opérations sur les résultats de la requête

10. Trouvez les 3 films les plus regardés au cours des 6 derniers mois.

localhost/phpmyadmin/index.php?route=/table/sql&db=bd_devoir&table=Historique_Visionnage

Admin

Server: 127.0.0.1 - Base de données: bd_devoir - Table: Historique_Visionnage

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations Suivi Déclencheurs

Afficher la zone SQL

La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les cases à cocher ainsi que les liens Éditer, Copier et Supprimer ne sont pas disponibles.

Affichage des lignes 0 - 2 (total de 3, traitement en 0.0005 seconde(s))

```
SELECT f.titre, COUNT(hv.id_film) AS nb_visionnages FROM Historique_Visionnage hv JOIN Film f ON hv.id_film = f.id_film WHERE hv.date_visionnage >= CURDATE() - INTERVAL 6 MONTH GROUP BY f.id_film ORDER BY nb_visionnages DESC LIMIT 3;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

titre	nb_visionnages
La Cité du Soleil	2
L'Étoile du Désert	1
Renaissance	1

Opérations sur les résultats de la requête

11. Identifiez les utilisateurs ayant un abonnement actif, mais n'ayant visionné aucun film.

12. Créez un utilisateur SQL avec un accès en lecture seule sur toutes les tables.

Exécuter une ou des requêtes SQL sur la table « bd_devoir.historique_visionnage »:

```
1 CREATE USER 'Taha'@'localhost' IDENTIFIED BY 'motdepasse';
2 GRANT SELECT ON *.* TO 'Taha'@'localhost';
3 FLUSH PRIVILEGES;
```

SELECT * SELECT INSERT UPDATE DELETE Effacer Format Récupérer la requête auto-sauvegardée

☐ Lier les paramètres

Conserver cette requête SQL dans les signets :

Délimiteur : ☐ Afficher à nouveau la requête après exécution ☐ Conserver la boîte de requêtes ☐ ROLLBACK à la fin ☒ Activer la vérification des clés étrangères Exécuter

Masquer la zone SQL

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0041 seconde(s))

```
CREATE USER 'Taha'@'localhost' IDENTIFIED BY 'motdepasse';
```

13. Accordez à un administrateur SQL le privilège de modifier les informations des films et des abonnements.

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privileges Opérations Suivi Déclenche

Exécuter une ou des requêtes SQL sur la table « bd_devoir.historique_visionnage »:

```
1 GRANT UPDATE, DELETE, INSERT ON Film TO 'root'@'localhost';
2 GRANT UPDATE, DELETE, INSERT ON Abonnement TO 'root'@'localhost';
3 FLUSH PRIVILEGES;
4
```

SELECT * SELECT INSERT UPDATE DELETE Effacer Format Récupérer la requête auto-sauvegardée

☐ Lier les paramètres

Conserver cette requête SQL dans les signets :

Délimiteur : ☐ Afficher à nouveau la requête après exécution ☐ Conserver la boîte de requêtes ☐ ROLLBACK à la fin ☒ Activer la vérification des clés étrangères

Masquer la zone SQL

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0035 seconde(s).)

14. Affichez les films qui durent plus de 2 heures et ont été visionnés par des utilisateurs ayant un abonnement 'Standard'.

Affichage des lignes 0 - 0 (total de 1, traitement en 0.0008 seconde(s).)

```
SELECT DISTINCT f.titre FROM Film f JOIN Historique_Visionnage hv ON f.id_film = hv.id_film JOIN Abonnement a ON hv.id_utilisateur = a.id_utilisateur WHERE f.duree > 120 AND a.type_abonnement = 'Standard';
```

☐ Profilage [Éditer en ligne] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table

Options supplémentaires

titre
Le Dernier Testament

☐ Tout afficher Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table

Opérations sur les résultats de la requête

15. Trouvez les utilisateurs ayant regardé un film d'un genre spécifique, mais n'ayant pas d'abonnement 'Premium'.

Afficher la zone SQL

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0005 seconde(s).)

```
SELECT DISTINCT u.nom FROM Utilisateur u JOIN Historique_Visionnage hv ON u.id_utilisateur = hv.id_utilisateur JOIN Film f ON hv.id_film = f.id_film JOIN Abonnement a ON u.id_utilisateur = a.id_utilisateur WHERE f.genre = 'VotreGenre' AND a.type_abonnement != 'Premium';
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

nom

Opérations sur les résultats de la requête

Créer une vue

Conserver cette requête SQL dans les signets

Intitulé : ☐ Signet visible pour les autres utilisateurs

Conserver cette requête SQL dans les signets