

Documentation

1. Introduction :

Objectif de la base de données :

Développer une base de données robuste et flexible pour gérer les réservations de places dans plusieurs cinémas partenaires. Le système devra permettre la réservation de billets, la gestion des séances, et la maintenance des données tarifaires et clientèle.

Portée du projet :

Ce projet couvre la conception, la création, le peuplement, la sécurité, la sauvegarde et la maintenance d'une base de données relationnelle. L'objectif est de fournir une solution centralisée pour la réservation en ligne et sur place, accessible aux administrateurs et aux clients.

Besoins des utilisateurs :

Les utilisateurs auront besoin de réserver des places sans choix spécifique de siège, avec un processus de paiement en ligne et sur place. Les administrateurs doivent pouvoir insérer les séances, gérer les réservations et visualiser les transactions. Des droits spécifiques seront attribués aux gérants de cinéma pour administrer les séances de leurs établissements respectifs.

2. Modèle de Données :

Le modèle conceptuel de données (MCD) pour le système de réservation de cinéma est conçu pour faciliter les interactions entre les cinémas, les films, les séances, et les clients. Il prend en compte les aspects suivants :

- **Entité Film** : Stocke les informations sur les films, y compris le titre, la durée, le genre, le réalisateur, et la date de sortie.

- **Entité Cinéma** : Contient les détails sur chaque cinéma, comme le nom, l'adresse, et le nombre de salles disponibles.

- **Entité Salle** : Représente les différentes salles dans chaque cinéma, avec des attributs tels que la capacité et les équipements.

- **Entité Séance** : Gère les horaires des films, associant chaque séance à un film, une salle et un cinéma spécifiques.

- **Entité Tarif** : Définit les différents tarifs applicables aux réservations.

- **Entité Client** : Conserve les données personnelles des clients pour la gestion des réservations et la communication.

- Entité Réservation : Suit les réservations effectuées par les clients, en liant les séances, les tarifs, et le nombre de places réservées.

- Entité Paiement : Enregistre les détails des transactions financières liées aux réservations.

Justification de la structure choisie :

Le modèle vise à maximiser l'efficacité et la normalisation des données. En séparant les informations dans des entités distinctes reliées par des relations claires, la base de données est plus facile à maintenir et à évoluer. Les contraintes de clé étrangère garantissent l'intégrité des données en ne permettant que des références valides entre les tables, réduisant les risques d'incohérences. Cette conception favorise également des requêtes plus performantes, car elle minimise les redondances et facilite les jointures entre les tables.

3. Description des Tables :

Table Cinéma :

idCinema (INT) : Clé primaire, identifiant unique pour chaque cinéma.

nom (VARCHAR) : Nom du cinéma.

adresse (VARCHAR) : Adresse complète du cinéma.

nombreSalles (INT) : Nombre de salles disponibles dans le cinéma.

Table Film :

idFilm (INT) : Clé primaire, identifiant unique pour chaque film.

titre (VARCHAR) : Titre du film.

durée (INT) : Durée du film en minutes.

genre (VARCHAR) : Genre du film.

réalisateur (VARCHAR) : Nom du réalisateur du film.

dateSortie (DATE) : Date de sortie du film.

Table Salle :

idSalle (INT) : Clé primaire, identifiant unique pour chaque salle.

capacite (INT) : Capacité d'accueil de la salle.

equipements (VARCHAR) : Description des équipements de la salle.

Table Scéance :

idSeance (INT) : Clé primaire, identifiant unique pour chaque séance.

idCinema (INT) : Clé étrangère liée à idCinema de la table Cinéma.

heure (TIME) : Heure de début de la séance.

date (DATE) : Date de la séance.

idFilm (INT) : Clé étrangère liée à idFilm de la table Film.

idSalle (INT) : Clé étrangère liée à idSalle de la table Salle.

Les relations entre les tables sont établies par des clés étrangères. Par exemple, idCinema dans la table Scéance est une référence à idCinema de la table Cinéma, permettant de lier une séance à un cinéma spécifique. Ces relations permettent des requêtes inter-tables, comme la recherche de toutes les séances pour un film donné ou la localisation des cinémas offrant une séance particulière. Les contraintes sont utilisées pour maintenir l'intégrité des données, comme les contraintes de clé étrangère pour assurer des références valides entre les tables.

4. Scripts SQL :

Dans cette section, nous détaillons les scripts SQL utilisés pour la création, la manipulation et la maintenance des données.

Création de tables avec contraintes :

Les tables ont été conçues pour refléter fidèlement les besoins du système de réservation de cinéma, intégrant des contraintes d'intégrité pour maintenir la qualité des données.

Exemple de script pour la création de la table Film :

```
CREATE TABLE Film (  
    idFilm INT PRIMARY KEY AUTO_INCREMENT,  
    titre VARCHAR(255) NOT NULL,  
    duree INT CHECK (duree > 0),  
    genre VARCHAR(100),  
    realisateur VARCHAR(255),  
    dateSortie DATE,  
    CONSTRAINT UC_Film UNIQUE (titre, dateSortie)  
);
```

Insertion de données factices :

Nous avons peuplé les tables avec des données factices pour simuler des scénarios réalistes et valider le comportement du système.

Exemple d'insertion dans la table Film :

```
INSERT INTO Film (titre, duree, genre, realisateur, dateSortie) VALUES
```

('Les Aventuriers de l'Arche Perdue', 115, 'Action', 'Steven Spielberg', '1981-06-12');

Requêtes d'opérations courantes

Les opérations courantes comme l'ajout, la modification, la suppression et la recherche de données sont essentielles pour la gestion quotidienne de la base de données.

Exemple de mise à jour d'un film :

UPDATE Film SET duree = 120 WHERE idFilm = 1;

5. Procédures de Sécurité :

Cette section aborde les méthodes mises en place pour sécuriser l'accès à la base de données et protéger les données.

Pratiques de sécurisation :

- Utilisation de mots de passe forts et renouvelés périodiquement.
- Restreindre les privilèges au strict nécessaire pour chaque utilisateur.
- Activation du chiffrement pour les données en transit et au repos.

Application des patches de sécurité : Des procédures sont établies pour veiller à ce que notre système soit à jour avec les derniers patches de sécurité, appliqués lors de fenêtres de maintenance programmées.

6. Stratégie de Sauvegarde et de Maintenance :

La durabilité et l'intégrité des données sont assurées par une stratégie de sauvegarde complète et un plan de maintenance régulier.

Fréquence des sauvegardes :

- Sauvegardes complètes hebdomadaires.
- Sauvegardes différentielles quotidiennes.

Méthode de sauvegarde :

- Les sauvegardes sont stockées sur des disques sécurisés avec réplication dans un second site géographique.

Plan de maintenance régulière :

- Vérification et optimisation des tables chaque mois.
- Révision des logs et analyse de performance chaque trimestre.

7. Captures d'écran :

The first screenshot shows the 'Utilisateurs ayant accès à « projet-cinema.scéance »' page in phpMyAdmin. It displays a table of users with their privileges and actions.

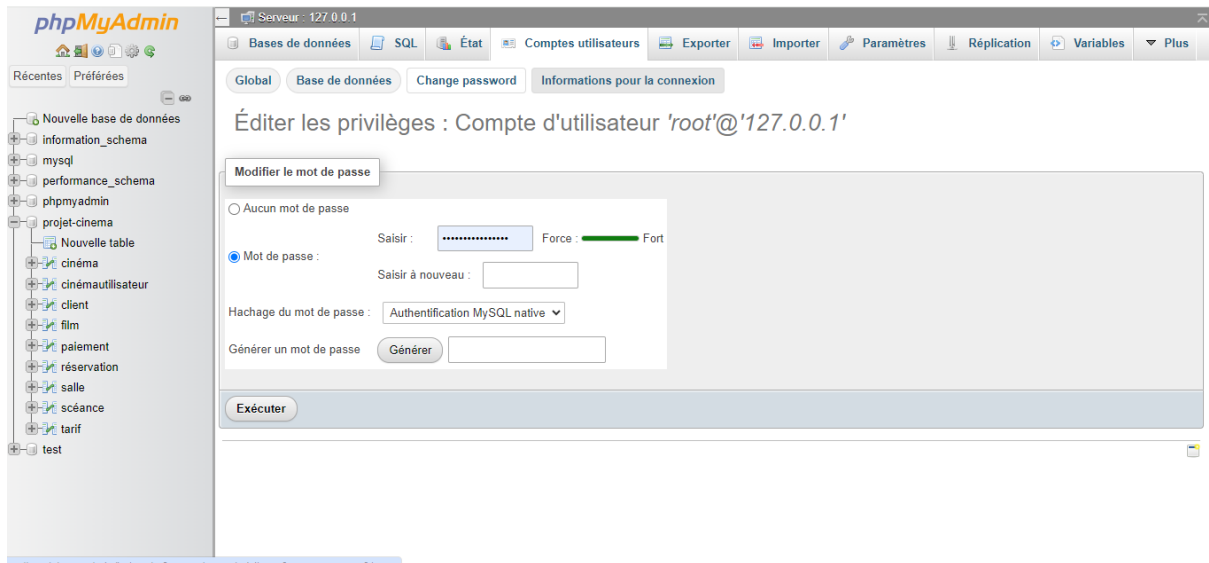
Nom d'utilisateur	Nom d'hôte	Type	Privileges	« Grant »	Action
<input type="checkbox"/> root	127.0.0.1	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter
<input type="checkbox"/> root	:::1	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter
<input type="checkbox"/> root	localhost	global	ALL PRIVILEGES	Oui	Éditer les privilèges Exporter

Below the table, there is a section for 'Nouvel utilisateur' with a button to 'Ajouter un compte d'utilisateur'.

The second screenshot shows the 'Structure' page in phpMyAdmin for the 'projet-cinema' database. It displays a list of tables with their actions and statistics.

Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> cinéma	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> cinémautilisateur	Parcourir Structure Rechercher Insérer Vider Supprimer	0	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> client	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> film	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> paiement	Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_general_ci	32,0 kio	-
<input type="checkbox"/> réservation	Parcourir Structure Rechercher Insérer Vider Supprimer	1	InnoDB	utf8mb4_general_ci	64,0 kio	-
<input type="checkbox"/> salle	Parcourir Structure Rechercher Insérer Vider Supprimer	3	InnoDB	utf8mb4_general_ci	16,0 kio	-
<input type="checkbox"/> scéance	Parcourir Structure Rechercher Insérer Vider Supprimer	2	InnoDB	utf8mb4_general_ci	64,0 kio	-
<input type="checkbox"/> tarif	Parcourir Structure Rechercher Insérer Vider Supprimer	4	InnoDB	utf8mb4_general_ci	16,0 kio	-
9 tables	Somme	17	InnoDB	utf8mb4_general_ci	256,0 kio	0 o

At the bottom, there is a button to 'Créer une nouvelle table'.



8. Conclusion :

Au terme de ce projet, une base de données fonctionnelle pour la gestion des réservations de cinéma a été mise en place. Le système est capable de gérer les films, les séances, les réservations, et les paiements. Un accent particulier a été mis sur la sécurité et les bonnes pratiques de sauvegarde.