



## GDWFSCAUBDDEXAIII1A

**Ceci est un modèle de copie. N'oubliez pas de renseigner vos prénom/nom, ainsi que le nom et le lien vers le projet.**

**Vous pouvez bien sûr agrandir les cadres pour répondre aux questions sur la description du projet si nécessaire.**

**Prénom :** Pierre

**Nom :** CHAMINADE

**ATTENTION ! PENSEZ À RENSEIGNER VOS NOM ET PRÉNOM DANS LE TITRE DE VOS FICHIERS / PROJETS !**

Nom du projet : Créer et administrer une base de données

Lien Github du projet : [git@github.com:Pierrechami/Evaluation-bdd-pb.git](https://github.com/Pierrechami/Evaluation-bdd-pb)

Lien Drive du projet (si nécessaire) :

URL du site (si vous avez mis votre projet en ligne) :

### Description du projet

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions. Dans cette rubrique, le jury cherche à voir comment vous procédez : comment vous organisez votre travail, comment vous réalisez concrètement la tâche ou l'opération pas à pas. Utiliser un langage professionnel. Employez le « je », car vous parlez en votre nom. Vous pouvez écrire au temps présent.

Une fois avoir pris connaissance du contexte et des enjeux du projet, je commence par la réalisation du diagramme de classe UML afin de pouvoir représenter les classes, les attributs et leurs relations. (Cf Informations complémentaires)

Je me connecte au serveur MySQL et au compte utilisateur. Une fois la requête effectuée, j'affiche la liste de mes bases de donnée afin de pouvoir m'assurer que je ne vais pas en créer une nouvelle avec le meme nom déjà existant.

Maintenant je peux créer ma base de données portant le nom « reservation\_place\_cinema », par la suite j'utilise la commande « USE » qui me permet de désigner la base de donnée courante, c'est-à-dire vers laquelle seront dirigées les prochaines requêtes SQL.

Au moment de la création des tables de ma base de données j'ai utilisé un utilitaire de sauvegarde et de restauration, pour cela j'ai lancé les requêtes suivante « ENGINE=InnoDB ». Les principaux avantages d'utiliser InnoDB concerne la possibilité de bien gérer l'intégrité des données, même lors d'un crash du serveur. J'ai aussi effectué l'instruction « CHARSET= utf8 »

Je créer 3 tables « Cinéma Complex », « info » et « administrator » qui ont pour type d'associations « la composition », permettant d'indiquer une dépendance forte entre deux objets et de lier leurs durées de vie.

Table « administrator » dispose de 3 colonnes : id (clé primaire) , le nom et le prénom de l'administrateur.

Table « infos » contient toutes les informations du complexe du Cinéma et à pour clé étrangère « id » de l'administrateur.

Table « Cinéma Complex » permet d'identifier le cinéma grâce a la clé primaire « id » et a la colonne « name », la table dispose aussi d'une clé étrangère faisant référence a la clés primaire de la table « infos » afin d'associé toute les informations à un complexe.

Je réalise 2 tables supplémentaire la table « hall » et la table « Movie théâtres » disposant d'une relation One-To-Many.

La table « hall » dispose de 2 colonnes un « Id » comme clé primaire et un « type\_hall » permettant de connaitre le type de salle.

La table « Movie théâtres » est importante dans la bdd car elle permet principalement de savoir qu'elle salle appartient a quelle complex et dispose d'une autre clé étrangère, faisant référence au type de salle.

Vu qu'un cinéma sans film n'est pas un cinéma, je créer 2 classes « Genre » et « Movie » ayant une cardinalité One-To-Many.

Table « Genre » dispose d'une clé primaire et d'une colonne « libelle »

Table « Movie » a pour colonne les informations du film ainsi qu'une clé étrangère faisant référence a la clé primaire de la table « Genre ».

Le gérant du complexe de cinéma nous informe aussi que ses clients on la possibilité de pouvoir payé de deux façons différentes, c'est pour cela que j'élabore une table « Payment » avec 2 colonnes «type\_payment » et un « id » clé primaire afin d'identifier les types de paiement disponible.

Par la suite je créer une table « PriceList » représentant les différent mode de paiement avec leur prix associé, cette table dispose d'une relation One-To-Many avec la table « Customers ».

La table « Customers » dispose de plusieurs colonnes permettant d'avoir des informations sur les clients du cinéma. La colonne « id » est la clé primaire mais contrairement au autres clé primaire, elle n'est pas auto\_increment mais de type « CHAR » afin d'utiliser « UUID() » au moment de l'insertion de des données. Les UUID permettent de régler les problèmes des entiers auto-incrémentés, au détriment de la lisibilité des URL et d'une mise en place souvent plus complexe, cela aura pour avantage de ne donner aucune information sur le nombre d'utilisateurs qui sont enregistré sur notre base de donnée. La table a pour clé étrangère « id » de la table « PriceList ».

Pour finir la création des tables j'initialise une dernière table nommée « Bookings » disposant d'une cardinalité One-To-Many avec les tables « Movies », « Customers », « MovieTheatres » et une cardinalité One-To-One avec la table « Payment » .

La table « Bookings » est une table clé dans la réalisation de mon diagramme, elle permet d'avoir les informations sur toutes les réservations grâce a ses différentes clés étrangères.

Afin de m'assurer que j'ai répondu à l'ensemble des demandes client j'effectue des requêtes sql permettant de démontrer la fiabilité de la bdd par rapport aux exigences.

La première requête permet de montrer grace au numéro de réservation qu'il y a la possibilité de réserver un film dans plusieurs cinémas , la seconde montre des réservations avec le passage du même film et du même complexe.

La 3eme requêtes permet de calculer le nombre de places restants dans une salle pour une séance donné.

La 4ème requêtes sélectionne la table « Payment » afin de montrer que les cinémas proposent différents tarifs possibles, l'avant dernière requête montre si les clients ont payé en ligne ou sur place.

La dernière requêtes permet de connaître l'administrateur du cinéma.

Pour terminer j'effectue une commande permettant la sauvegarde et la restauration de la base de données.

2. Précisez les moyens utilisés. Expliquez tout ce dont vous avez eu besoin pour réaliser vos tâches : langages de programmation, frameworks, outils, logiciels, documentations techniques, etc...

Pour réaliser l'ensemble de la base de données, j'ai eu besoin d'utiliser :

MAMP

- Permet l'utilisation d'un serveur Web local

phpMyAdmin

- Cela m'a uniquement servi dans le but de visualiser et vérifier mes différentes requêtes sql

DataGrip

- IDE de base de données

Bcrypt

- Fonction de hachage

Gitmind

- Outil en ligne permettant de créer des diagrammes UML

Documentation sql.sh

3. Contexte. Les noms des organismes, entreprises ou associations, dans lesquels vous avez exercé vos pratiques

NB: Pour le cas des exercices et évaluations demandées sur la plateforme Studi, il s'agit de...Studi

Studi et un gérant de cinéma, on prit contact, suite à cette entretien studi nous a communiqué les différents échanges et les demandes client.

Question 1

« Est-ce qu'il y aura uniquement les séances d'un seul cinéma?

- Non. Nous sommes partenaires de plusieurs complexes, et donc nous aimerions qu'il soit possible de réserver au choix dans l'un d'entre eux. »

Question 2

« Dans un cinéma, peut-il y avoir le passage du même film au même horaire dans plusieurs salles différentes ?

- Oui, cela arrive dans certains de nos complexes pour les blockbusters les plus rentables ! »

Question 3

« Désirez-vous que vos clients puissent choisir leur place?

- Pas nécessairement. Néanmoins, il faudra bien faire attention au nombre de places disponibles dans une salle! »

Question 4

« Quels tarifs possibles?

- Plein tarif: 9€20

- Étudiant: 7€60

- Moins de 14 ans: 5€90 »

Question 5

« Le paiement se fera sur place ou en ligne?

- Nous comptons à terme digitaliser nos tickets. S'il est possible de faire les deux, ce serait génial ! Sinon, nous nous contenterons pour l'instant d'un paiement sur place »

Question 6

« D'un point de vue back office, qui pourra inscrire les séances?

- Il nous faudrait un ou plusieurs administrateurs chez nous ayant une vue sur tout. Mais nous aimerions qu'un utilisateur par complexe ait les droits d'ajouter les séances de son propre cinéma »

4. Informations complémentaires (*facultatif*)

## Diagramme de classe UML

