

## ▀ ▀ ▀

**Créer et administrer une base de données**

**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 : Cédric**

**Nom : DUMAINE**

Nom du projet : Créer et administrer une base de données

Lien Github du projet : <https://github.com/Hed40/Evaluation_Studi-6.git>

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.

J’ai tout d’abord pris connaissance du sujet de l’évaluation. En fonction des éléments demandés par notre client, j’ai pu réfléchir au schéma et déterminer les noms des entités que je devais créer en les surlignant sur l’annexe du sujet puis en les écrivant sur une simple feuille.

Ainsi, j’ai pu définir ma méthode conceptuelle de donnée et y intégrer les entités suivantes **USERS, ROLES, THEATERS, MOVIESROOMS, MOVIESSESSIONS, MOVIES, BOOKINGS, PAYMENTS, PRICES** dans mon Diagramme de classe en me servant de la méthode MERISE. J’appellerais ma base de données « cinémas ». J’ai ensuite intégré mes champs en commençant par toutes mes ID par entités en choisissant l’ **UUID** afin de proposer une identité unique à chacune de mes entités et pour leur très faible probabilité de répétitivité. J’ai ensuite ajouté mes champs d’entités, les relations entre elles et enfin leurs cardinalités.

J’ai ensuite pu définir la méthode logique de donnée, à partir des cardinalités maximums de chacune des entités pour ainsi laisser une trace de l’id dans l’entité la moins forte. Ainsi, si un Theater programme plusieurs ou un nombre indéfini de séances soit N contre 1, N étant plus fort alors l’id\_theater ( programmed\_by) sera inscrit dans l’entité Movie Sessions.

Une fois mon diagramme terminé au brouillon, j’ai pu le reporter numériquement, j’ai pour cela utilisé ***FIGMA*** car je préférais directement le dessiner à la main. (Et mes versions d’évaluations de Star UML et autre ne fonctionnaient plus …)

Ensuite, j’ai pu commencer à recréer ma base de données en SQL. Pour cela j’ai simplement utilisé un fichier texte sous Windows.

J’ai ensuite et au fur et à mesure de la conception de mon code, effectué divers tests de bon fonctionnement en local via **PHPMYADMIN**, mais uniquement dans le but de vérifier si tout se passait bien.

J’ai donc commencé par écrire mon code SQL pour créer ma Data base « ***cinémas***» en insérant l’option **IF NOT EXISTS** afin de ne pas retourner d’erreurs si une base de même nom existe déjà.

J’ai ensuite créé toutes mes tables une par une dans un ordre logique afin d’éviter les erreurs lors de leur construction, pour ainsi pouvoir effectuer des tests sans encombre. J’ai inséré et défini les colonnes et les types de données qui seront contenus dans chacune des colonnes (Entier, chaînes de caractères, date, time, Décimal) et aussi le nombre caractères maximum.

J’ai aussi inséré l’encodage **Utf8mb4** afin de pouvoir avoir beaucoup plus de caractères que **l’UTF8.**

Dans un même temps, j’ai défini et mis en place les clés étrangères ou contraintes afin de lier les champs ensembles **CONSTRAINT … FOREIGN KEY … REFERENCES …**

Une fois toutes mes tables créées, j’ai attribué et intégré les rôles (Privilèges) pour chaque type d’utilisateurs, dans notre cas, il y aura **Admin, Manager, Seller et Clients** qui auront tous un accès différent en fonction de leur rôles respectif. (Les clients n’auront pas la possibilité de programmer une séance ! Logique !)

Une fois toutes mes tables créés et fonctionnelles, j’ai élaboré des scripts d’alimentation **« Factices »** de données dans la base en utilisant **INSERT INTO nom de la table () VALUES ()** et le site [Mockaroo - Générateur de données aléatoires et outil de simulation d’API | JSON / CSV / SQL / Excel](https://www.mockaroo.com/) pour générer de la donnée « LAMBDA » en utilisant le **type GUID** pour mes id par exemple.

Bien sûre, j’aurais pu en insérer beaucoup plus de données dans mes tables, mais j’ai préféré aller à l’essentiel uniquement pour vérifier si les relations se faisaient correctement.

Pour plus de sécurité en cas de plantage du serveur et dans un souci d’optimisation des performances, j’ai intégré et précisé le moteur de stockage transactionnel **Inno DB** pour chacune de mes tables.

J’ai aussi ***encrypté (hashé) tous les mots de passes utilisateurs*** via le site [bcrypt.fr](https://www.bcrypt.fr/) (Très pratique !)

Enfin, j’ai créé un fichier « **Cinémas.sql** » sous ***VSCODE*** puis j’ai copié/ collé mon code afin de mettre à disposition.

1. 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...

FIGMA, VS CODE, DOCUMENT TEXTE, XAMPP, PHPMYADMIN, METHODE MERISE, DOCUMENTATION SQL

1. 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

1. Informations complémentaires (*facultatif*)

Joyeuses fëtes à vous 😊