



# Créer et administrer une base de données



## **GDWFSCAUBDDEXAIII1A**

**Prénom :** Aurélie **Nom :** DUMAS

Nom du projet : Evaluation\_5

Lien Github du projet : <a href="https://github.com/Aurelie-Dumas/evaluation5">https://github.com/Aurelie-Dumas/evaluation5</a>
Lien Google Drive du projet : <a href="https://drive.google.com/drive/folders/1a0vjU--FW1G-">https://drive.google.com/drive/folders/1a0vjU--FW1G-</a>

axKipD7tCOlPNINYNxgT?usp=share link

## **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 réalisé mon évaluation par le biais de 4 étapes.

Étapes de la conception d'une base de données :

- 1. Analyse de la situation existante et des besoins (note de clarification)
- 2. Création d'un modèle conceptuel de données (MCD) qui permet de représenter tous les aspects importants du problème. Spécifications Fonctionnelles + Techniques (Diagrammes de cas d'utilisation, d'activité et de séquence + Diagramme de classe)
- 3. Traduction du modèle conceptuel en modèle logique de données (MLD) qui permet de traduire le problème en structure relationnelle. (Modèle relationnel)
- 4. Implémentation d'une base de données en SQL dans un SGBD, à partir du modèle logique

## **Etape 1 : Note de Clarification**

La note de clarification est une reformulation du cahier des charges, qui précise, ajoute et supprime des éléments.

J'ai réalisé ma note de clarification sur Google Doc afin que chacune des parties puissent :

- avoir un document facile à mettre à jour
- suggérer des modifications si nécessaire

Il est également possible de tracer l'auteur des demandes de modifications.



Note de Clarification accessible via ce lien :

https://docs.google.com/document/d/1ipDXK17VJJJiTl5 pD2oocxLujFp9sNz/edit?usp=share link&ouid=1 11545192049531792637&rtpof=true&sd=true

## **Etape 2 : Spécifications Fonctionnelles et Techniques**

## **Spécifications Fonctionnelles**

Les spécifications fonctionnelles précisent les fonctionnalités du site et retranscrivent de manière visuelle les liens entre les applications ainsi que tous les cas d'utilisation possibles.

Les diagrammes vont nous permettre de les représenter. Ils sont disponibles via ce lien : https://drive.google.com/drive/folders/1WwxughYbxC42np zhsfzpUF76-F6XYik?usp=share link

J'ai réalisé deux types de diagramme : Des diagrammes de structure afin de cadrer les spécifications techniques et des diagrammes de comportements répertoriant les spécifications fonctionnelles.

Tout d'abord, j'ai commencé par réaliser le Diagramme de cas d'utilisation. Ce diagramme est important car il permet de dresser une liste exhaustive et claire des fonctionnalités du site.

On y retrouve les acteurs, les cas d'utilisateurs ainsi que les relations entre les fonctionnalités.

Je commence par représenter les principaux acteurs du projet, qui seront :

- Les visiteurs
- Les clients
- Les gérants
- Les administrateurs

Les visiteurs pourront consulter des films et se créer un compte.

Les clients ont accès aux mêmes fonctionnalités que les visiteurs et pourront également consulter leur compte et réserver une séance. J'ai donc créé une relation de généralisation entre les acteurs.

Idem pour les administrateurs et les gérants. Les gérants pourront gérer les films/séances. Les administrateurs auront accès à cette fonctionnalité ainsi qu'à la consultation des commandes.

J'ai ensuite matérialisé les dépendances entre les fonctionnalités et les acteurs.

Les confirmations de commande seront envoyées uniquement si le client a procédé à la réservation d'une séance en amont. Idem pour les actions nécessitant d'être connecté.

Afin de mettre en avant le fait qu'un utilisateur pourra se connecter via login/mdp, Google ou Facebook, j'utilise la généralisation. La généralisation permet de préciser qu'un cas d'utilisation est une variante d'un autre.

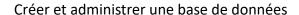
Pour plus de visibilité et afin de représenter le cas d'une commande client dans son ensemble, je décide de réaliser le diagramme d'activité de la mise au panier d'un film.

Je commence par réaliser le nœud initial et le nœud de fin d'activité.

J'ajoute les différentes actions. Lorsque qu'un cas d'utilisation possède plusieurs scénarios possibles pendant leur exécution, j'ajoute un nœud de décision.

C'est le cas, par exemple, lors de la vérification du stock :

- Soit le stock est égal à zéro et dans ce cas j'affiche le message d'erreur et renvoie le visiteur sur la page lui permettant de choisir une autre séance.
- Soit le stock est positif et l'ajout au panier peut être effectué





Pour finir et afin de mieux comprendre les liens entre un client, l'application et la base de données, je décide de réaliser deux Diagrammes de Séquence.

Un pour représenter le cas de gestion d'un film et un autre pour représenter le cas d'une consultation de commande.

Comme pour le diagramme de cas d'utilisation, j'ajoute les acteurs ainsi que leur ligne de vie. Les acteurs sont représentés avec des bonhommes bâtons. Les autres sont représentés par des formes géométriques.

J'ajoute ensuite les communications entre les acteurs par le biais de flèches et de messages. Les réponses sont symbolisées par des flèches pointillées.

## **Spécifications Techniques**

Les spécifications techniques permettent de s'assurer que tous les membre de l'équipe développement utiliseront la même structure. Elles sont issues des spécifications fonctionnelles et permettent de découper le code en classes.

Pour ce faire, j'ai réalisé un diagramme de classes. Il permet de représenter les différentes structures de données de l'application et les relations qui existent entre elles.

Pour réaliser mon diagramme de classe, je me suis aidée de ma note de clarification réalisée en amont ainsi que des retours du commanditaire.

J'ai tout d'abord répertorié tous les objets que j'avais identifié. J'ai ensuite déterminé comment chacun des objets étaient liés les uns aux autres. J'ai ensuite créé la structure de chacune des classes en ajoutant des attributs.

#### Ce qui donne :

Un ou plusieurs Administrateurs gèrent un ou plusieurs Cinémas.

Un cinéma possède une ou plusieurs Salles.

Une ou plusieurs Séances ont lieu dans une salle de cinéma.

Aucune ou plusieurs Séances projettent un Film.

Une séance possède plusieurs Tarifs.

Un client passe une ou plusieurs Commande.

Une ou plusieurs commandes sont composées de plusieurs tarifs.

Une commande possède un Détail de commande.

## Etape 3 : Passage au Modèle Relationnel

J'ai ensuite converti mon diagramme de classe en modèle relationnel.

Le modèle relationnel est une manière de modéliser les relations existantes entre plusieurs informations, et de les ordonner entre elles.

J'ai commencé par créer mes tables.

Ensuite, j'ai converti les associations du diagramme en relations et contrainte en me basant sur la cardinalité des associations.

J'ajoute les clauses ON DELETE CASCADE quand c'est nécessaire. Si une entité (parent) est supprimée, les valeurs (enfants) qui dépendent de cette entité seront également supprimées.

De même, j'ajoute également, quand c'est nécessaire, la clause ON DELETE SET NULL. Si une entité est supprimée, toutes les valeurs qui dépendent de cette entité seront automatiquement définies sur NULL.



Créer et administrer une base de données

## Etape 4 : Implémentation d'une base de données en SQL dans un SGBD, à partir du modèle logique

J'ai commencé par installer MySQL par le biais de Docker dans mon environnement WSL (windows subsystem for linux). Cela m'a permis d'avoir une base de données isolée dans une machine virtuelle.

Je me suis ensuite connecté à mon serveur MySQL via mon terminal grâce à la commande suivante : mysql -u root -p afin d'exécuter mes requêtes SQL préparées précédemment.

A noté : Pour l'enregistrement de mes données utilisateurs et notamment leurs mots de passe, j'ai utilisé la fonction MD5() afin de hasher les mots de passe. La fonction md5() retourne le md5 de la chaîne, sous la forme d'un nombre hexadécimal de 32 caractères.

J'ai ensuite exporté ma base de données pour la sauvegarder grâce à la commande suivante que j'ai exécuté dans mon terminal.

mysqldump -u -p -h 127.0.0.1 --opt evaluation\_5 > evaluation\_5.sql

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...

#### Outils:

Visual Paradigm Online : Conception des différents diagrammes

## Logiciels:

Docker Desktop: Installation de MySQL

Terminal : Ecriture de lignes de commande

## **Documentations Techniques:**

MySQL: https://dev.mysql.com/doc/

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

4. Informations complémentaires (facultatif)

Temps passé: 20h

### Annexes:

Diagrammes

Note de Clarification

Fichier SQL « Modèle Relationnel »

Fichier SQL « Base de données Evaluation 5 »

Présentation power point Evaluation\_5