

# CONCEPTION D'UNE BASE DE DONNEES

## CONTEXTE

Dans le cadre de la formation de Développeur Web et Web Mobile, j'ai été évalué sur un projet dont le but était de créer la base de données d'un complexe de cinéma.

Modalités / Restrictions :

- Modélisation de la base de données sous forme de MCD ou diagramme de classe UML.
- Écriture des différents scripts SQL de création sécurisée de base de données/de ses tables ainsi que des relations et contraintes entre elles.
- Remplissage des tables avec de fausses données (toujours à l'aide de commandes SQL).
- Explication dans la description du projet d'un processus de sauvegarde de la BDD complète.

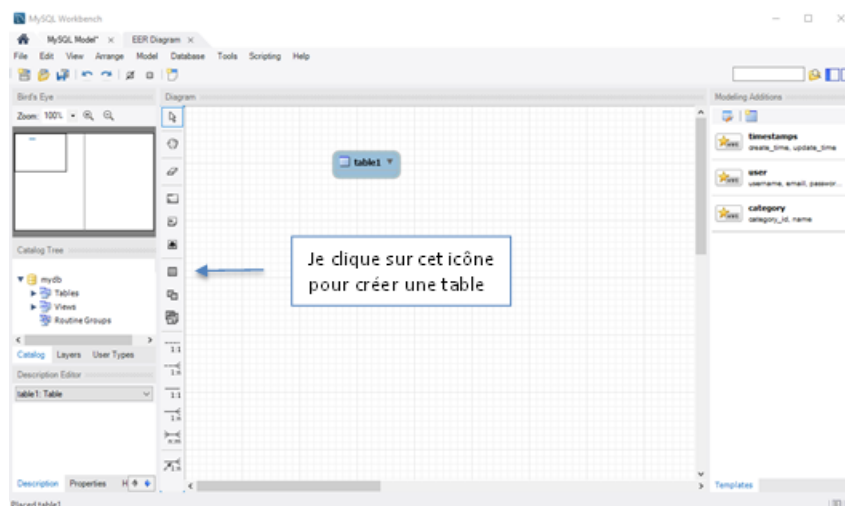
Nécessité de travailler uniquement dans le terminal !

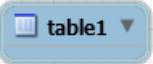
## TACHES EFFECTUEES

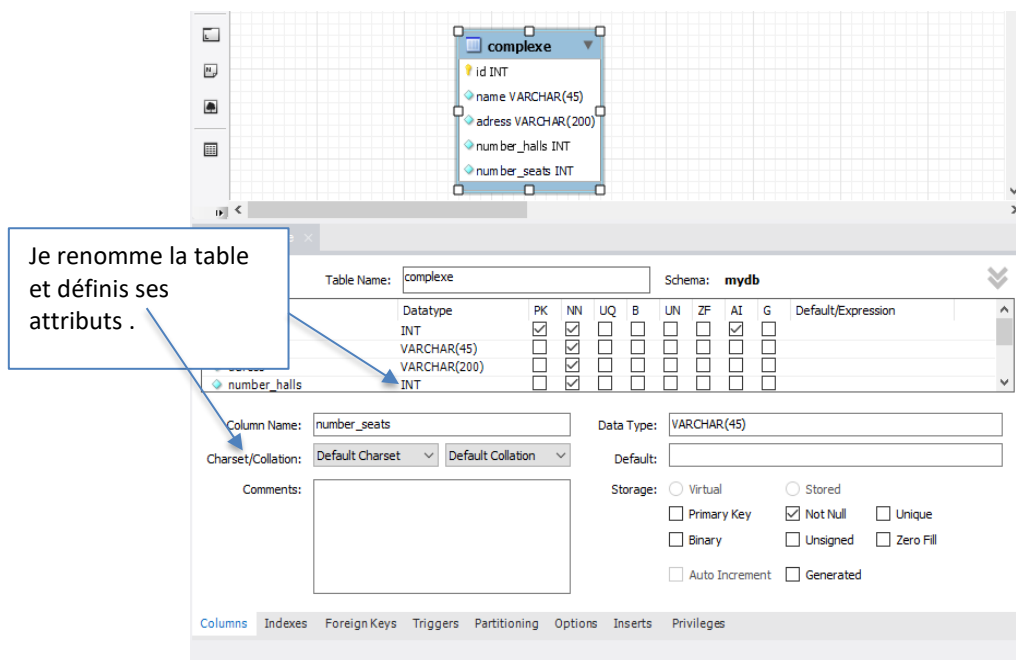
### Organisation physique des données

Pour modéliser la base de données sous forme de modèle conceptuel de données, j'ai utilisé le logiciel MySQL Workbench.

#### Création d'une entité

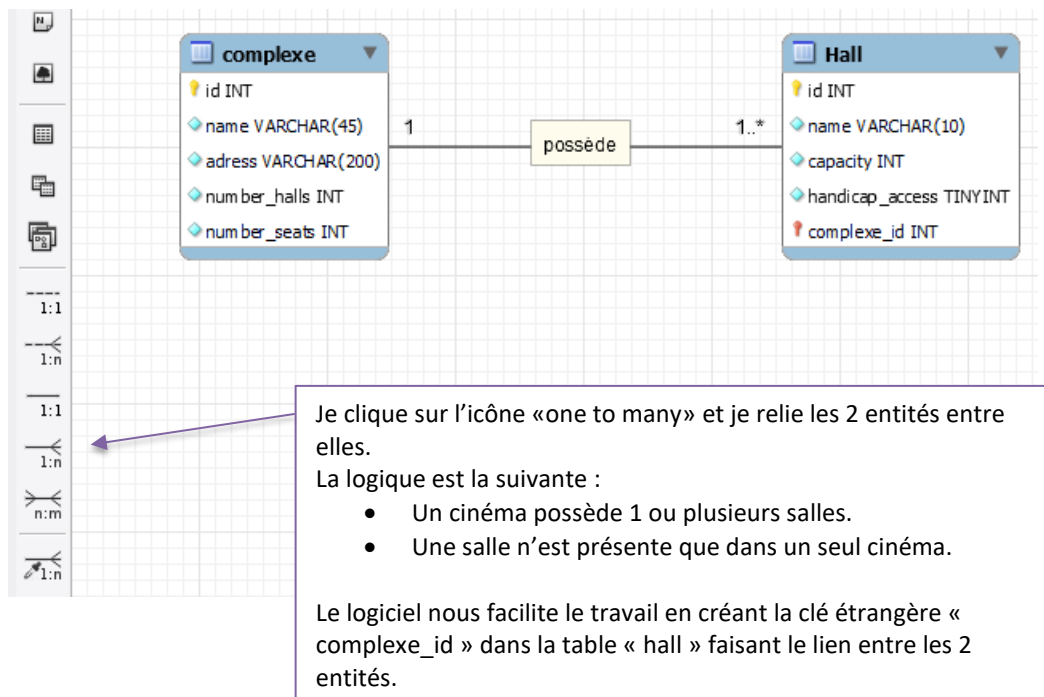


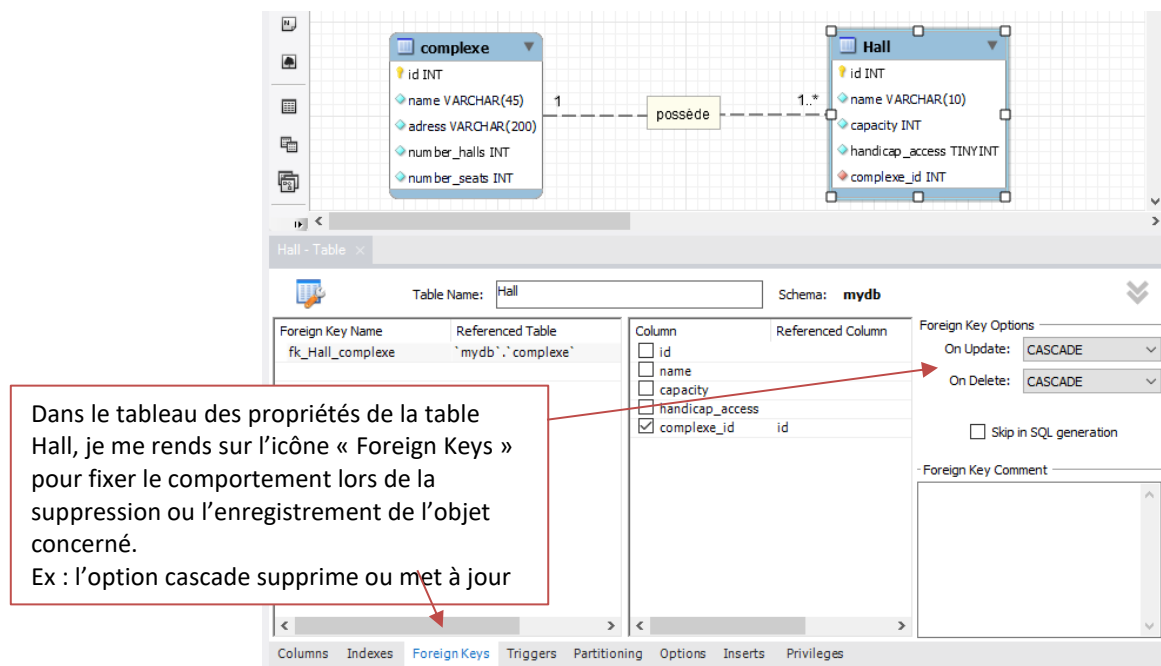
Je double clique sur  pour accéder aux propriétés de la table.



## Liaison entre les entités

Exemple d'une liaison entre une entité faible et une entité forte





## Explication des cardinalités

Une séance est diffusée dans un cinéma et un cinéma diffuse une ou plusieurs séances.  
 Une séance est diffusée dans une salle et une salle diffuse une ou plusieurs séances.  
 Une séance concerne un film et un film est concerné par plusieurs séances.  
 Une séance possède une version de film et une version de film est présente dans 0 ou plusieurs séances.

Un film possède une catégorie de spectateurs (age) et une catégorie concerne 0 à plusieurs films.  
 Un film est composé d'une équipe ayant un ou plusieurs participants avec un ou plusieurs rôles.  
 Un film a un genre et un genre concerne 0 ou plusieurs films.

Une réservation concerne une séance et une séance fait l'objet de 0 à plusieurs réservations.  
 Une réservation fait l'objet d'un paiement et un mode de paiement concerne une ou plusieurs réservations.  
 Une réservation concerne un siège et un siège est présent dans 0 à plusieurs réservations.  
 Une réservation fait l'objet d'un tarif et un tarif est présent dans 1 à plusieurs réservations.  
 Une réservation concerne un utilisateur et un utilisateur fait 0 à plusieurs réservations.

## Création de la base de données avec MySQL

### Création de la base de données

A partir du terminal, pour commencer à créer la base de données, je me connecte à MySQL en tapant la commande suivante : `mysql -u root -p`

Je tape ensuite le mot de passe que le terminal me demande.

J'exécute la commande `show databases` pour lister l'ensemble de mes bases de données et vérifier que le nom de ma base de données n'existe pas déjà.

Je crée la base de données « cinemas\_studi » :

```
CREATE DATABASE complexes DEFAULT CHARACTER SET utf8 ;
```

Je précise l'encodage attendu.

## Création d'une table

Pour créer les tables, je me positionne sur la base de données `USE complexes;`

Les tables intègrent un moteur de stockage transactionnel

```
CREATE TABLE complexe (
  id INT PRIMARY KEY NOT NULL AUTO_INCREMENT,
  name VARCHAR(45) NOT NULL,
  adress VARCHAR(45) NOT NULL,
  zip INT NOT NULL,
  city VARCHAR(45) NOT NULL,
  number_halls INT NOT NULL,
  number_seats INT NOT NULL)
ENGINE = InnoDB;
```

Pour créer les relations entre les tables, on doit utiliser des clés étrangères auxquelles on doit appliquer des contraintes lors de la suppression ou de la mise à jour d'un champ dans une table.

Voici un exemple avec la table booking comprenant de nombreuses clés étrangères.

```
CREATE TABLE booking (
  id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
  user_id INT NOT NULL,
  session_id INT NOT NULL,
  seat_id INT NOT NULL,
  price_id INT NOT NULL,
  payment_id INT NOT NULL,
  UNIQUE (seat_id, session_id),
  FOREIGN KEY (user_id) REFERENCES user (id) ON DELETE CASCADE ON UPDATE CASCADE,
  FOREIGN KEY (seat_id) REFERENCES seat (id) ON DELETE RESTRICT ON UPDATE CASCADE,
  FOREIGN KEY (price_id) REFERENCES price (id) ON DELETE RESTRICT ON UPDATE CASCADE,
  FOREIGN KEY (payment_id) REFERENCES payment (id) ON DELETE RESTRICT ON UPDATE CASCADE,
  FOREIGN KEY (session_id) REFERENCES session (id) ON DELETE CASCADE ON UPDATE CASCADE)
ENGINE = InnoDB;
```

Chaque table contient une clé primaire auto-incrémentée.

## Explications pour le champ user\_id :

On désigne le champ user\_id en tant que clé étrangère. Cette clé fait référence au champ id de la table user et c'est sa valeur qui figurera dans le champ user\_id de la table booking.

ON DELETE CASCADE ON UPDATE CASCADE : lorsqu'on supprime un utilisateur, les réservations le concernant le sont aussi. Lorsqu'on le met à jour, les modifications sont répercutées dans la table booking.

La contrainte UNIQUE figure également dans cette table. Elle est là pour préciser que 2 réservations ne peuvent avoir le même siège et la même séance. Je vérifie avec la commande suivante sachant qu'une réservation avec la même séance et le même siège a déjà été faite par un autre utilisateur :

```
INSERT INTO booking (user_id, session_id, seat_id, price_id, payment_id) VALUES ('2', '1', '1', '2', '2');
```

En retour on obtient bien un message d'erreur.

```
mysql> INSERT INTO booking (user_id, session_id, seat_id, price_id, payment_id) VALUES ('2', '1', '1', '2', '2');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`complexes`.`booking`, CONSTRAINT `booking_ibfk_5` FOREIGN KEY (`session_id`) REFERENCES `session` (`id`) ON DELETE CASCADE ON UPDATE CASCADE)
mysql>
```

## Insertion de données

Pour coller à la réalité, j'ai inséré de vrais films tirés de l'actualité ainsi que leurs acteurs et réalisateurs. Les données des autres tables sont factices. J'ai inséré manuellement ces données car elles n'étaient pas nombreuses.

En revanche, pour la table user je me suis servi du site <https://www.mockaroo.com/> qui génère des données automatiquement en fonction des champs choisis et renvoie le code correspondant.

The screenshot shows the Mockaroo website interface for generating fake data. It displays a schema for a table named 'user' with the following fields and options:

Field Name	Type	Options
firstname	First Name (European)	blank: 0%
lastname	Last Name	blank: 0%
email	Email Address	blank: 0%
password	Password	blank: 0%
date_creation	Datetime	07/29/2021 to 07/29/2022, format: m/d/yyyy, blank: 0%

Below the fields, there are options for the number of rows (10), format (SQL), table name (user), and a checkbox for 'Include CREATE TABLE'. At the bottom, there are buttons for 'DOWNLOAD DATA', 'PREVIEW', 'SAVE THIS SCHEMA', and 'MORE'.

```
insert into user (firstname, lastname, email, password, date_creation) values ('Joséphine', 'Campo', 'lcampo@linkedin.com', 'GZjRTUmrQP', '2021-10-20');
insert into user (firstname, lastname, email, password, date_creation) values ('Bérangère', 'Bassingden', 'ibassingden1@so-net.ne.jp', 'pwk5Uc', '2022-04-28');
insert into user (firstname, lastname, email, password, date_creation) values ('Uò', 'Paeckmeyer', 'hpaeckmeyer2@mozilla.org', '4foDc9Z', '2021-09-20');
insert into user (firstname, lastname, email, password, date_creation) values ('Yáo', 'Beacham', 'hbeacham3@ycombinator.com', 'jjaZ9eU', '2022-05-06');
insert into user (firstname, lastname, email, password, date_creation) values ('Dà', 'Winscum', 'wwinscum4@posteros.com', 'msfDGH5T2UFs', '2022-07-09');
insert into user (firstname, lastname, email, password, date_creation) values ('Pénélope', 'Matushevich', 'cmatushevich5@jugem.jp', '2cVrL1C', '2022-03-05');
insert into user (firstname, lastname, email, password, date_creation) values ('Almérinda', 'Shelsher', 'ashelsher6@elpais.com', 'TD2ecB1m4E', '2021-09-24');
insert into user (firstname, lastname, email, password, date_creation) values ('Kù', 'Wallenger', 'twallenger7@wordpress.org', '19Pbhm7gpax', '2022-02-26');
insert into user (firstname, lastname, email, password, date_creation) values ('Håkan', 'Mawd', 'dmawd8@europa.eu', '9bEVp5n3lMu', '2021-10-27');
insert into user (firstname, lastname, email, password, date_creation) values ('Clémentine', 'Raper', 'araper9@home.pl', 'eWQ4VR3X', '2022-04-09');
```

*Génération de données factices par le site [mockaroo.com](https://www.mockaroo.com/)*

Dans la pratique, avant d'entrer les différentes données en base de données, chacune fera l'objet d'un contrôle de sécurité. La donnée la plus importante étant le mot de passe qui subira un cryptage via PHP par exemple, avant insertion.

Pour simuler ce cryptage, j'utilise l'application bcrypt sur le site [bcrypt.fr](https://bcrypt.fr/).

Par exemple, pour la première insertion, nous passons du mot de passe original `GZjRTUmrQP` au mot de passe crypté ` \$2y\$10\$8.x4vglRMzM1awW34glwJODplqdq36M.SEb0uPJkNA4MKE3E0alqq`.

# Gestion de la base de données

## Création d'un compte administrateur ayant tous les droits

```
CREATE USER 'administrator'@'localhost'  
IDENTIFIED BY '$2y$10$msu6UfG8.kpgulr0HCWEfezHp1.o8AhewYxzJZvSAADTyMVx6aRxu';  
GRANT ALL PRIVILEGES ON complexes.* TO 'administrator'@'localhost';  
FLUSH PRIVILEGES;
```

## Création d'un compte utilisateur - lecture des données

```
CREATE USER 'user'@'localhost'  
IDENTIFIED BY '$2y$10$Gb01JFRa5Qvr5cQe7i8ry.SHOgMuk1QxueG9YXaSo32rlvPUfYF2u';  
GRANT SELECT ON complexes.* TO 'user'@'localhost';  
FLUSH PRIVILEGES;
```

## Sauvegarde et restauration

J'ouvre le terminal à partir du répertoire de travail et je rentre la commande suivante afin de faire une copie de la base de données.

```
Mysqldump -u root -p complexes > complexes.sql
```