

Partie Persistance Hibernate

Procédure : Création et vérification d'une base de données MySQL sous Docker

1. ⚡ Crédit de la base de données via script SQL

✓ Étape 1 : Créer un fichier SQL (depuis le terminal)

`nano creationbase.sql`

✓ Étape 2 : Coller le contenu SQL

Ex1 :

Script de création des tables clients et commandes, avec insertion de données. (Voir le contenu du script précédemment fourni.)

-- 🔑 1. Crédit de la base de données

`DROP DATABASE IF EXISTS GC2000;`

`CREATE DATABASE GC2000;`

`USE GC2000;`

-- 📊 2. Crédit de la table des clients

`CREATE TABLE clients (`

`client_id INT AUTO_INCREMENT PRIMARY KEY,`

`nom VARCHAR(100) NOT NULL,`

`prenom VARCHAR(100) NOT NULL,`

`email VARCHAR(150) UNIQUE NOT NULL,`

`ville VARCHAR(100),`

`date_inscription DATE`

`);`

-- 📦 3. Crédit de la table des commandes

`CREATE TABLE commandes (`

`commande_id INT AUTO_INCREMENT PRIMARY KEY,`

`client_id INT NOT NULL,`

```
date_commande DATE NOT NULL,
montant DECIMAL(10,2) NOT NULL,
statut ENUM('en cours', 'expédiée', 'livrée', 'annulée') DEFAULT 'en cours',
FOREIGN KEY (client_id) REFERENCES clients(client_id) ON DELETE CASCADE
);
```

--  4. Insertion de données dans la table clients

```
INSERT INTO clients (nom, prenom, email, ville, date_inscription)
VALUES
('El Amrani', 'Youssef', 'youssef.elamrani@example.com', 'Casablanca', '2025-11-01'),
('Bennani', 'Fatima', 'fatima.bennani@example.com', 'Rabat', '2025-11-03'),
('Alaoui', 'Omar', 'omar.alaoui@example.com', 'Fès', '2025-11-07'),
('Chakir', 'Hajar', 'hajar.chakir@example.com', 'Marrakech', '2025-11-10'),
('Taleb', 'Nadia', 'nadia.taleb@example.com', 'Agadir', '2025-11-12');
```

--  5. Insertion de données dans la table commandes

```
INSERT INTO commandes (client_id, date_commande, montant, statut)
VALUES
(1, '2025-12-01', 950.00, 'livrée'),
(1, '2025-12-10', 320.00, 'expédiée'),
(2, '2025-12-05', 1100.50, 'en cours'),
(3, '2025-12-04', 450.00, 'annulée'),
(4, '2025-12-06', 780.00, 'livrée'),
(5, '2025-12-08', 640.75, 'en cours');
```

Ex2 : à compléter par votre base de données (Projet)

Étape 3 : Enregistrer et quitter

- **Ctrl + O** → Entrée pour enregistrer
 - **Ctrl + X** pour quitter l'éditeur nano
-

2. Exécution du script SQL dans le conteneur Docker

Étape 1 : Copier le script dans le conteneur

```
docker cp creationbase.sql mon-mysql:/tmp/
```

Remplacer mon-mysql par le nom exact de votre conteneur MySQL.

Étape 2 : Exécuter le script SQL

```
docker exec -it mon-mysql sh -c "mysql -u root -pMonMotDePasse123 < /tmp/creationbase.sql"
```

Adapter MonMotDePasse123 au mot de passe root MySQL de votre conteneur.

3. Vérification de la création et du contenu

Étape 1 : Accéder au conteneur

```
docker exec -it mon-mysql sh
```

Étape 2 : Se connecter au client MySQL

```
mysql -u root -p
```

Étape 3 : Afficher les bases existantes

```
SHOW DATABASES;
```

Étape 4 : Utiliser la base créée

```
USE GC;
```

Étape 5 : Lister les tables

```
SHOW TABLES;
```

Étape 6 : Vérifier les données insérées

SELECT * FROM clients;

SELECT * FROM commandes;

- Si vous voyez les données affichées correctement, la base de données est bien créée et fonctionnelle.

Adapter ces dernières vérifications à votre projet et soumettre le fichier creationbase.sql sur un entrepôt Github qui porte votre nom.