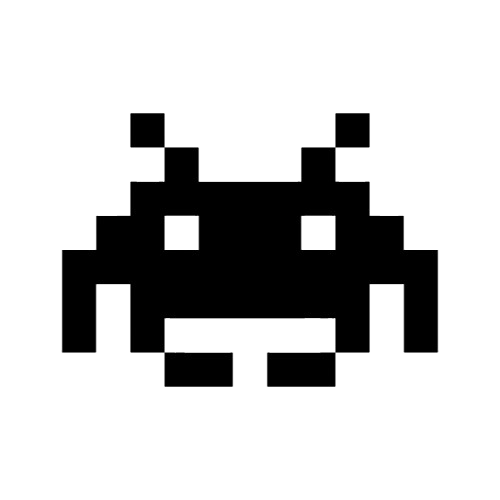
P\_DB\_106 - SpicyInvaders



César Klein - Cin2a

Etml - Vennes

32 périodes

Antoine Mveng

Table des matières

[1 Spécifications 3](#_Toc183005709)

[1.1 Titre 3](#_Toc183005710)

[1.2 Description 3](#_Toc183005711)

[1.3 Matériel et logiciels à disposition 3](#_Toc183005712)

[1.4 Prérequis 3](#_Toc183005713)

[2 Réalisation 4](#_Toc183005714)

[2.1 Dossier de Réalisation 4](#_Toc183005715)

[2.1.1 Réalisation du MCD 4](#_Toc183005716)

[2.1.2 Création de la base de données et importation des données 5](#_Toc183005717)

[2.1.3 Gestion des utilisateurs 7](#_Toc183005718)

[2.1.4 Requête de sélection 7](#_Toc183005719)

[2.1.5 Création des index 7](#_Toc183005720)

[2.1.6 Backup / Restore 7](#_Toc183005721)

[2.2 Modifications 7](#_Toc183005722)

[3 Tests 7](#_Toc183005723)

[3.1 Dossier des tests 7](#_Toc183005724)

[4 Conclusion 7](#_Toc183005725)

[4.1 Bilan des fonctionnalités demandées 7](#_Toc183005726)

[4.2 Bilan personnel 7](#_Toc183005727)

[5 Divers 8](#_Toc183005728)

[5.1 Journal de travail 8](#_Toc183005729)

[5.2 Bibliographie 8](#_Toc183005730)

[5.3 Webographie 8](#_Toc183005731)

# Spécifications

## Titre

**Thématique « SpicyInvader »**

## Description

Dans ce projet, nous allons travailler avec une base de données sur la thématique « SpaceInvader. »

## Matériel et logiciels à disposition

Pour ce projet nous avons droit a plusieurs logiciels à disposition. Voici les suivants :

* Un PC ETML
* Accès à Internet
* Docker
* MySQL
* Looping
* Git / GitHub

## Prérequis

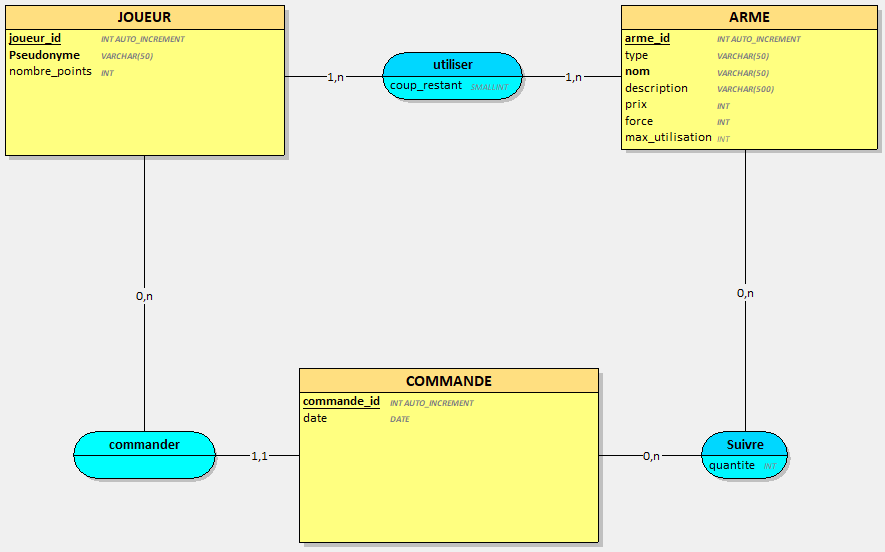
Les prérequis de ce projet sont les suivants :

* 162 analyser et modéliser des données
* 164 créer des bases de données et y insérer des données

# Réalisation

## Dossier de Réalisation

### Réalisation du MCD



Le pseudonyme est unique car sur des sites ou sur des jeux, il n’y a qu’une seul fois le même nom d’utilisateur.

Afin de créer correctement la base de données, des informations nous ont été données dans le cahier des charges du projet. Pour ne pas se tromper sur la création de la base donnée, une vérification de la base de données a été faite par le chef de projet.

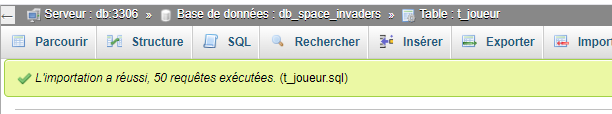
### Création de la base de données et importation des données

Pour la génération de données, j’ai utilisé l’outil Mockaroo excepté pour la table arme où j’ai utilisé chatgpt car la demande de génération de données était plus spécifique donc Mockaroo ne pouvait pas générer les données. Au niveau de la création et l’importation de la DB tout s’est passé correctement. J’ai utilisé le script que Looping m’a donnée et je l’ai inséré dans PHP MyAdmin. Nous pouvons aussi insérer le script dans MySQL en ligne de commande

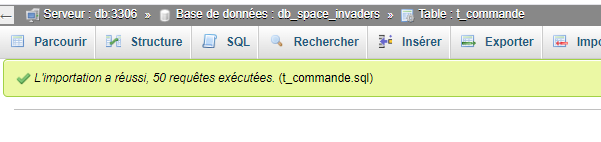


*Importation de la DB*

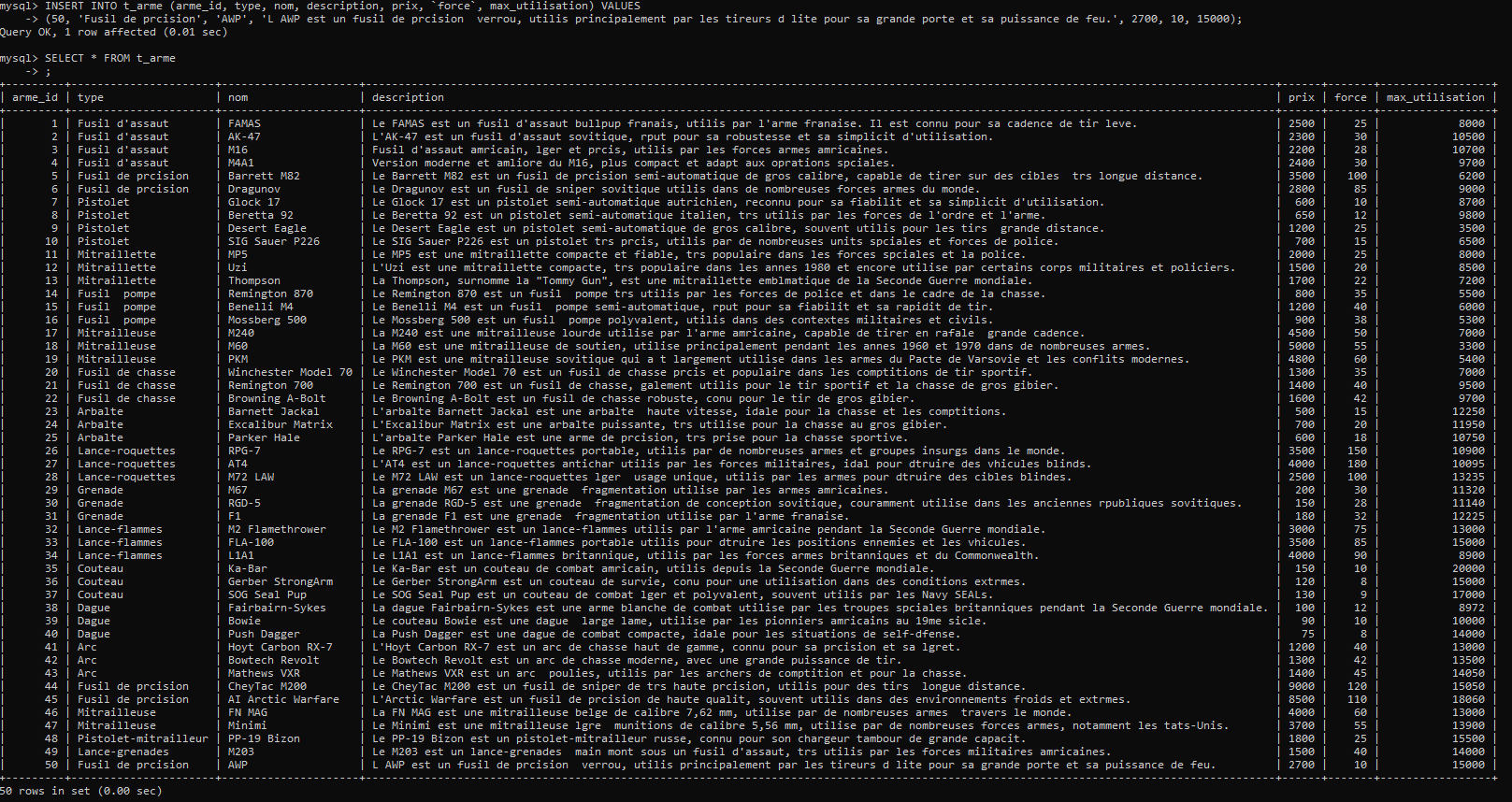
Pour l’insertion de données dans la base, j’ai utilisé pour certaines tables PHP MyAdmin mais aussi MySQL en ligne de commande comme le montre les captures d’écran ci-dessous :



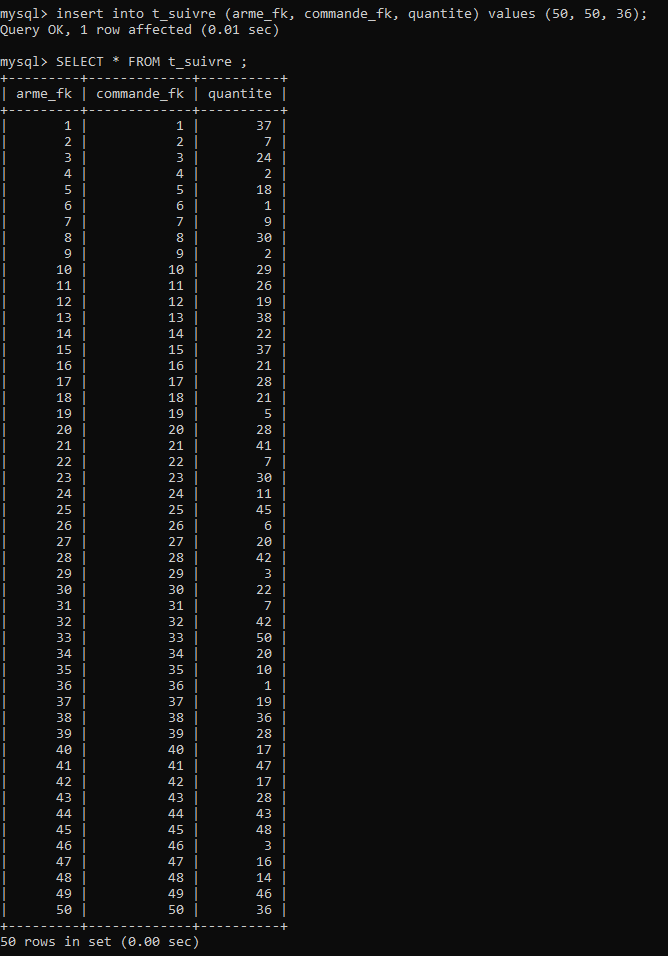
*Insertion t\_joueur*

**

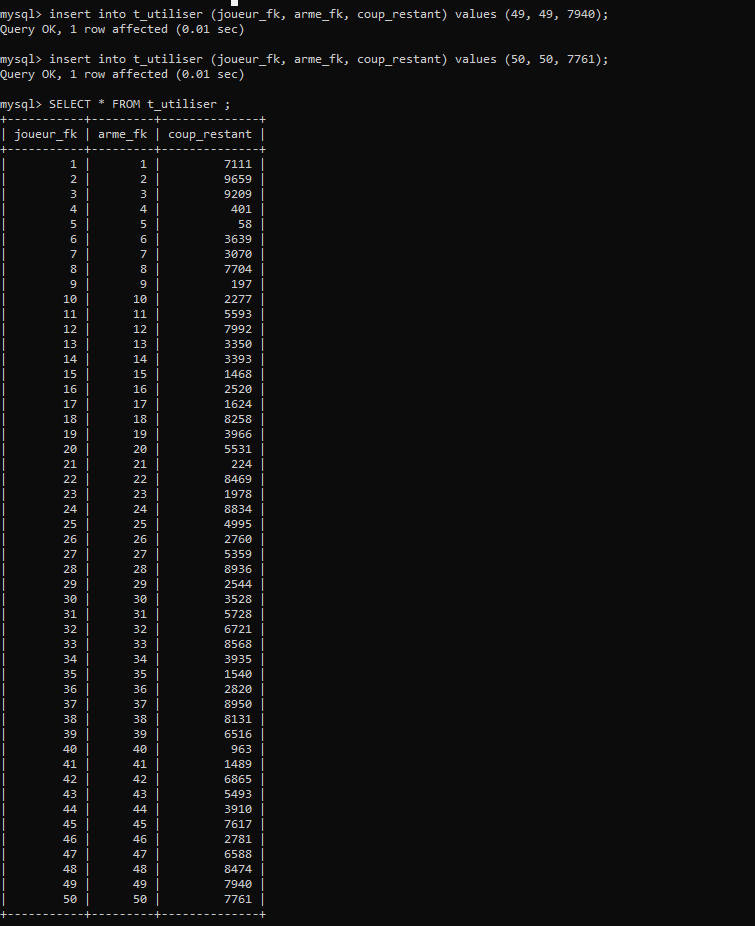
*Insertion t\_commande*

**

*Insertion t\_arme*

**

*Insertion t\_suivre*

**

*Insertion t’utiliser*

Pour insérer les données dans les tables j’ai donc utilisé les scripts que j’ai créé avec chatgpt et Mockaroo qui sont disponible dans le dossier :

*« GitHub\P\_DB-106-SpicyInvader\DB\03 - Insertion Données »*

Pour insérer des données dans une table, il faut utiliser la commande :

*« INSERT INTO nom\_de\_la\_table(données 1, donnée2, données3) values (1, 2, 3) ; »*

### Gestion des utilisateurs

Pour la gestion des utilisateurs, j’ai créé trois utilisateurs. Un joueur, un manager de boutique et également un administrateur. Dans le cahier des charges, il y avait plusieurs contraintes pour chaque utilisateur.

Le joueur pouvait :

* Lire les informations des armes (SELECT t\_arme).
* Créer une commande (INSERT t\_commande)
* Lire toutes les commandes (SELECT t\_commande).

Le joueur ne disposait que de ces droits. Il ne pouvait donc pas voir les autres tables que commandes et arme et il n’avait accès qu’a la base de données space invaders.

Pour le manager de boutique, il pouvait :

* Lire les informations sur les joueurs (SELECT t\_joueur)
* Mettre à jour, lire, supprimer des armes (UPDATE, SELECT, DELET t\_arme)
* Lire toutes les commandes (SELECT t\_commande)

Le manager comme le joueur, ne disposait que les droits si dessus. Et n’avait accès à aucune autre Base de données ou tables.

Pour terminer, l’administrateur avait tous les droits liés à la base de données space invaders.

Le script qui permet de créer les utilisateurs se trouve dans ce dossier :

« GitHub\P\_DB-106-SpicyInvader\DB\05 – Gestions des utilisateurs »

### Requête de sélection

Pour les requêtes SQL demandées dans le cahier des charges, tout a été fait en ligne de commande dans SQL.

Un fichier qui regroupe toutes les requêtes et leurs réponses se trouve dans GitHub

[Lien vers les requêtes](https://github.com/Cesar-Kln/P_DB-106-SpicyInvader/tree/main/DB/04%20-%20Requ%C3%AAte)

Il y avait en tout dix requêtes qui avait des spécifications différentes. Dans les requêtes, nous devions modifier le nom de certaines colonnes en utilisant l’outil « AS ». Nous devions également faire des jointures entre plusieurs tables en utilisant « JOIN, LEFT JOIN, RIGHT JOIN ». Nous devions aussi trier parfois l’affichage du résultat des requêtes en utilisant par exemple un « ORDER BY ».

Voici les requêtes demandées dans le cahier des charges :

**Requête n°1 :**

La première requête que l’on vous demande de réaliser est de sélectionner les 5 joueurs qui ont le meilleur score c’est-à-dire qui ont le nombre de points le plus élevé. Les joueurs doivent être classés dans l’ordre décroissant

**Requête n°2 :**

Trouver le prix maximum, minimum et moyen des armes.

Les colonnes doivent avoir pour nom « PrixMaximum », « PrixMinimum » et « PrixMoyen)

**Requête n°3 :**

Trouver le nombre total de commandes par joueur et trier du plus grand nombre au plus petit.

La 1ère colonne aura pour nom "IdJoueur", la 2ème colonne aura pour nom "NombreCommandes"

**Requête n°4 :**

Trouver les joueurs qui ont passé plus de 2 commandes.

La 1ère colonne aura pour nom "IdJoueur", la 2ème colonne aura pour nom "NombreCommandes"

**Requête n°5 :**

Trouver le pseudo du joueur et le nom de l'arme pour chaque commande.

**Requête n°6 :**

Trouver le total dépensé par chaque joueur en ordonnant par le montant le plus élevé en premier, et limiter aux 10 premiers joueurs.

La 1ère colonne doit avoir pour nom "IdJoueur" et la 2ème colonne "TotalDepense"

**Requête n°7 :**

Récupérez tous les joueurs et leurs commandes, même s'ils n'ont pas passé de commande.

Dans cet exemple, même si un joueur n'a jamais passé de commande, il sera quand même listé, avec des valeurs `NULL` pour les champs de la table `t\_commande`.

**Requête n°8 :**

Récupérer toutes les commandes et afficher le pseudo du joueur s’il existe, sinon montrer `NULL` pour le pseudo.

**Requête n°9 :**

Trouver le nombre total d'armes achetées par chaque joueur (même si ce joueur n'a acheté aucune Arme).

**Requête n°10 :**

Trouver les joueurs qui ont acheté plus de 3 types d'armes différentes

### Création des index

1. **Certains index existent déjà. Pourquoi** ?

Oui des index sont automatiquement créés dans les bases de données. Les clés primaires et les clés étrangères. Cela permet que les requêtes faites à la base de données soit plus optimisée / performantes. Par exemple grâce au index, les jointures entre deux tables sont généralement plus rapides.

1. **Quels sont les avantages et les inconvénients des index ?**

Comme je l’ai dit avant, les index sont créés afin que les résultats des requêtes soit plus rapides. Mais parfois quand nous devons mettre à jour des données, les index doivent être aussi mis à jour. Cela ajoute aussi du volume à la base de données (stockage).

1. **Sur quel champ (de quelle table), cela pourrait être pertinent d’ajouter un index ? Justifier votre réponse.**

J’ai décidé d’indexer le prix des armes (t\_arme)car si imaginons dans le shop des armes un utilisateur trie par prix entre 100$ et 1000$ donc grâce à l’index cela permet d’optimiser les réponses de requêtes dans ce genre.

Commande afin de trier le prix : CREATE INDEX idx\_prix ON t\_arme (prix) ;

### Backup / Restore

Pour le Backup, il suffit de se rendre dans le bash et de faire cette commande qui va permettre de créer le dump de la base de données

*mysqldump -u root -proot db\_space\_invaders > db\_space\_invaders\_Restore.sql*

Quand cela est fait vous pouvez changer l’emplacement du backup en utilisant la commande mv.

Pour restore la base de données, il faut rester dans le bash et exécuter cette commande :

*mysql -u root -proot < db\_space\_invaders\_Restore.sql*

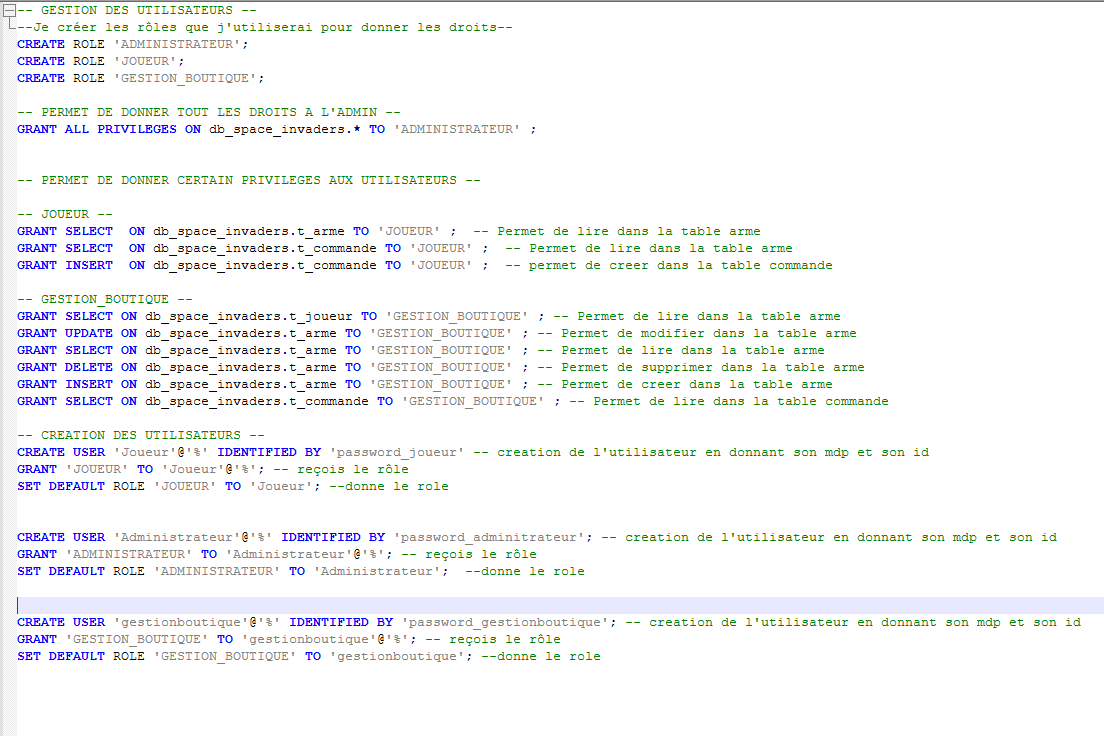
Nous utilisons le dump que nous venons de créer afin de restore la base de données.

# Tests

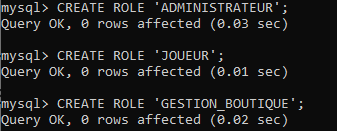
## Dossier des tests

### Gestion des utilisateurs

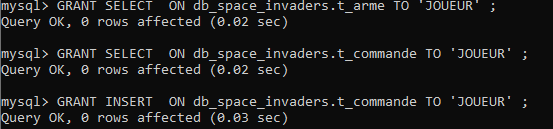
Pour la gestion des utilisateurs, j’ai créé un script SQL avec les commandes pour créer tous les utilisateurs demandés.

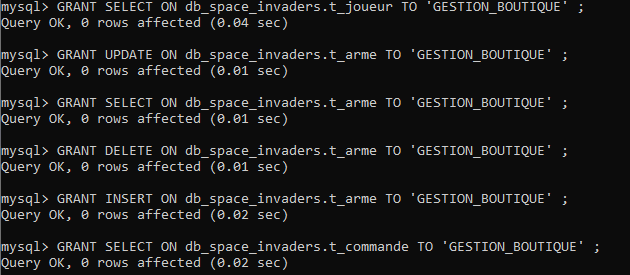


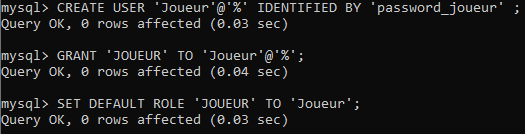
J’ai ensuite exécuté les commandes une par une afin de vérifier s’il y avait une erreur.

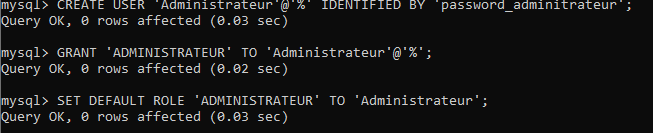


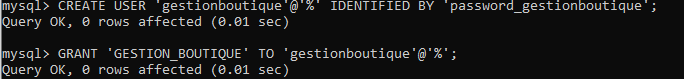








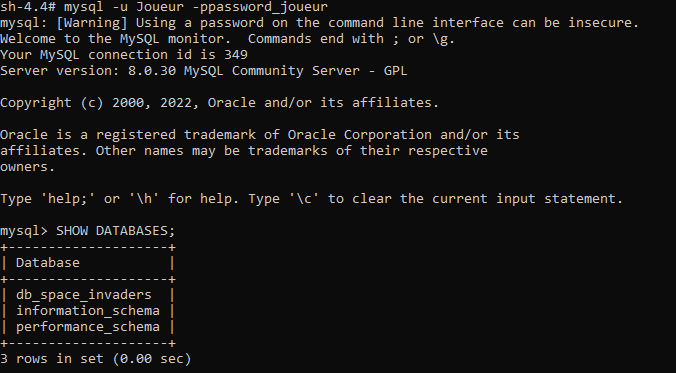


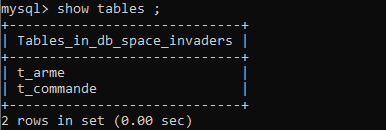




Nous pouvons voir que les requêtes de création d’utilisateurs et de rôle fonctionne correctement.

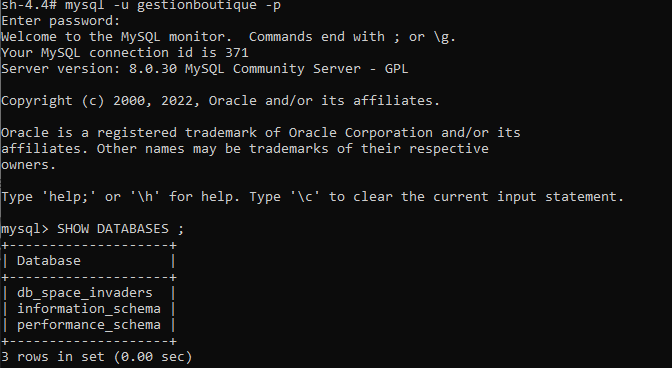
Je me suis ensuite connecté avec le compte joueur afin de vérifier si les rôles fonctionnent bien

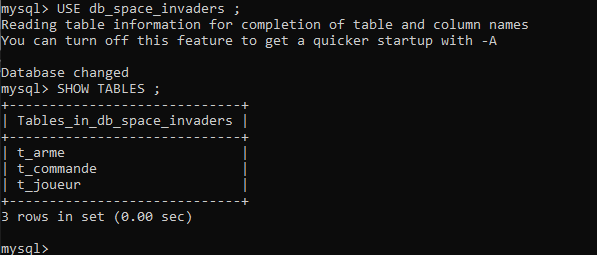




Nous pouvons voir que l’utilisateur a seulement les table dont il peut accéder.

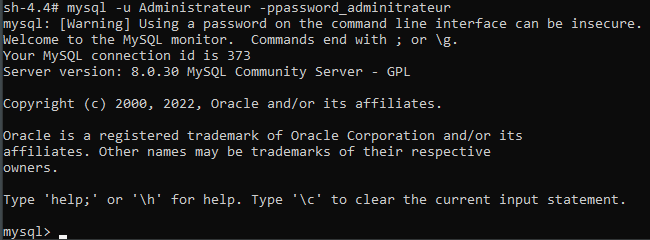
Je me suis également connecté avec le compte gestion boutique pour aussi vérifier que les rôles ont bien été fait

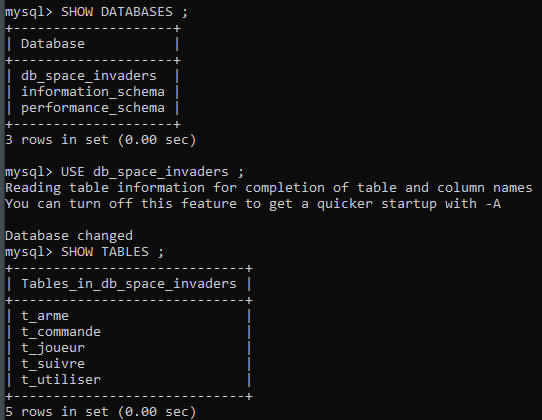




Nous pouvons encore voir qu’il n’pas toutes les tables donc pas tous les droits.

Pour finir ma vérification des utilisateurs, je me suis connecté avec le compte admin de la base afin de voir si tout fonctionne correctement.





L’administrateur a toutes les tables et a tous les droits sur la base de données ce qui est demandé.

### Requête de sélection

Pour les requêtes, il existe un fichier texte où se trouve toutes les requêtes et les résultats en lien avec ma base de données.

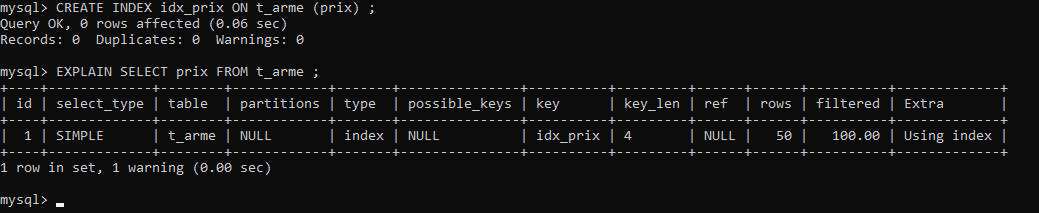
Dossier où se trouve le fichier SQL :

*« GitHub\P\_DB-106-SpicyInvader\DB\04 – Requête »*

Pour la requête il y avait des requêtes de tout type afin de voir si le sujet est bel et bien maitrisé par l’apprenti.

### Création des index

Pour la création de l’index sur le prix j’ai utilisé la commande présente dans la capture d’écran et afin de vérifier que l’index a bel et bien été créer et qu’il soit utilisé, j’ai fait un explain select qui montre et afficher quel index est utilisé pour la requête.



### Backup / Restore

Pour le backup j’ai le fichier dump qui permet de restore la base et afin d’être sûr qu’il n’y a pas de problème avec mon script, je l’ai envoyé à un collègue afin de vérifier si mon dump fonctionne correctement sur des machines différentes.

Pour le fichier dump, j’ai une copie en local sur mon PC et aussi un dans mon repository et un également dans mon SSD.

# Divers

## Journal de travail

Mon journal de travail se trouve dans le dossier JDT.

## Bibliographie

Théorie vue en module

## Webographie

* <https://www.w3schools.com>
* <https://www.mysql.com/>
* <https://www.mockaroo.com/>