P-OO\_SpaceInvaders



Auteur : Thomas Nardou

Date : 28.08.2023 – 03.11.2023

Périodes : 60

Chef de projet : Mathieu Meylan, Aurélie Curchod, Xavier Carrel

Table des matières

[Chapitre 2 : Partie programmation 1](#_Toc146544260)

[Chapitre 3 : Partie UX/UI 1](#_Toc146544261)

[3.2 : Analyse 1](#_Toc146544262)

[3.2.1 : Conception centré utilisateur 1](#_Toc146544263)

[3.2.2 : Palette Graphique 2](#_Toc146544264)

[Chapitre 4 : Partie Base de données 3](#_Toc146544265)

[4.1 : Importer la base de données 3](#_Toc146544266)

[4.2 : Requête N°1 3](#_Toc146544267)

[4.3 : Requête N°2 4](#_Toc146544268)

[4.4 : Requête N°3 4](#_Toc146544269)

[4.5 : Requête N°4 4](#_Toc146544270)

[4.6 : Requête N°5 5](#_Toc146544271)

[4.7 : Requête N°6 5](#_Toc146544272)

[4.8 : Requête N°7 6](#_Toc146544273)

# Chapitre 2 : Partie programmation

# Chapitre 3 : Partie UX/UI

## 3.2 : Analyse

### 3.2.1 : Conception centré utilisateur

Pendant ce projet, j'ai choisi de créer deux persona, un homme et une femme, afin de mieux comprendre les types d'utilisateurs potentiels de mon application, en prenant en compte les besoins et les préférences de chacun des sexes. Afin de réaliser cette partie j’ai dû utiliser l’IA chatGPT afin de me donner :

* Les informations personnelles de mes persona
* Leurs biographies
* Leurs envies
* Leurs frustrations

Lors de la réalisation de mes maquettes j’ai essayé de respecter les envies de mes persona le plus possible afin d’attirer un publique plus large. Voici mes persona :

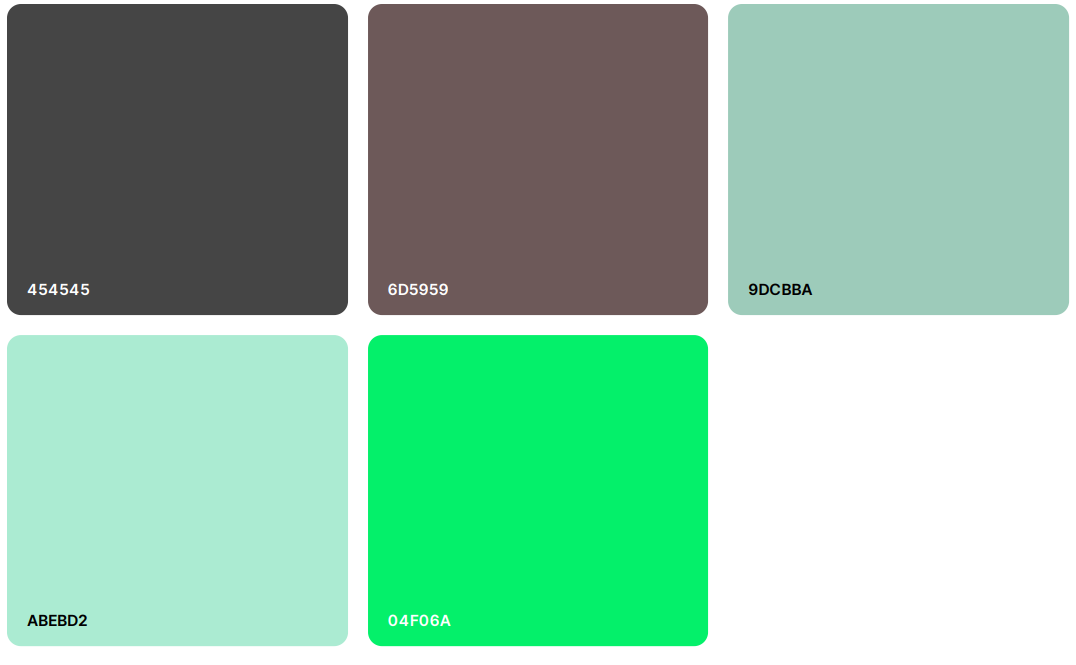




Vous trouverez aussi les persona en pièce jointe ou en appuyant sur [ce lien](Personna.pdf)

### 3.2.2 : Palette Graphique

Lors de la réalisation de mes maquettes j’ai dû choisir une palette de couleur afin d’avoir des pages plus jolies afin de réaliser cette partie je suis allé sur le site [coolors.co](https://coolors.co/) qui est un site qui génère des palettes de couleur et après avoir générer un certain nombre de palette je suis arrivé à ce résultat :



**678479**

J’ai décidé aussi de rajouter encore une nuance de vert (#678479) car je trouvais qu’il manquait des couleurs sombres donc j’ai décidé de la rajouter pour ajouter plus de contraste dans maquette tout en gardant une cohérence entre les couleurs. Si je me suis arrêté sur ces couleur c’est pour ces deux raisons :

1. Le vert et ses nuances peuvent faire rappeler les aliens et aussi le space Invaders original où le joueur est vert
2. Le noir et sa nuance peuvent faire rappeler l’espace c’est donc pour cela que j’ai choisi

# Chapitre 4 : Partie Base de données

Pour cette partie on nous a demandé d’exécuter des requêtes SQL dont une qui sert un importer un dump d’une base de données existante depuis un invite de commande et les autres qui servent à aller chercher des informations qui se trouvent dans la base de données.

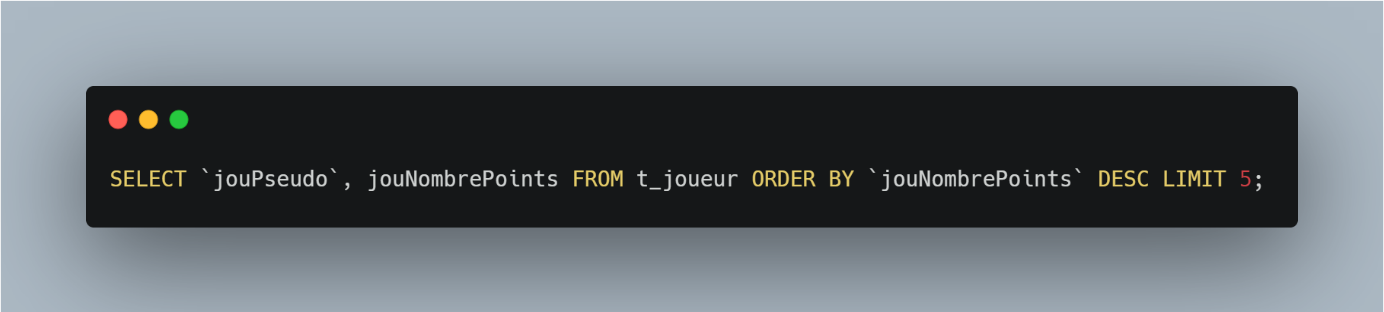
## 4.1 : Importer la base de données

Pour cette partie voici la commande que j’ai utilisé pour pouvoir importer la base de données



Sur cette commande on peut voir un "docker exec -i cc94b…" cette partie signifie que l’on va exécuter la commande dans un conteneur docker qui possède l’id "cc94b… " la seconde partie va être la commande que l’on va exécuter dans le conteneur dans notre cas la commande va lancer MYSQL avec l’username (-u) "root " et le mot de passe (-p) "root " et la dernière partie signifie que le conteneur va exécuter le script sql qui se trouve à l’emplacement indiqué.

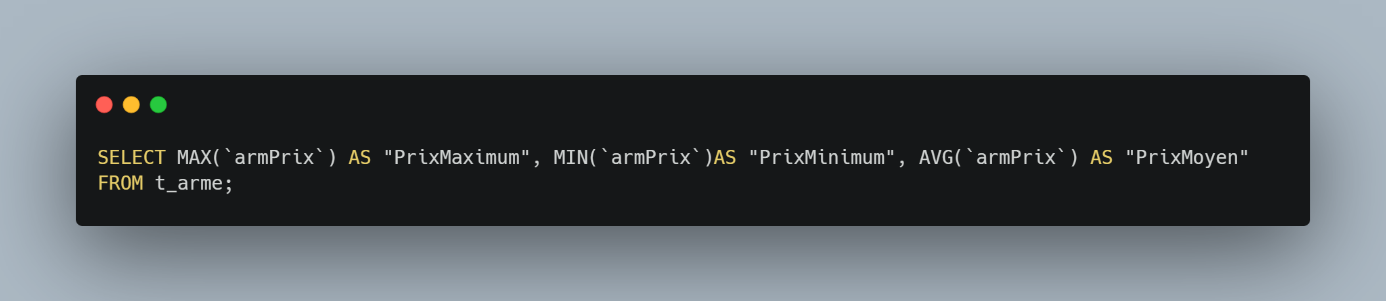
## 4.2 : Requête N°1

Voici la requête que j’ai utilisé pour répondre à la demande du cahier des charges : 

Pour cette requête je vais sélectionner le pseudo et les leurs nombres de points comme demander dans la requête dans la table qui leur correspond grâce au "FROM". Le "ORDER BY" va permettre d’ordonner mes résultats (par défaut : par ordre croissant) le "DESC" va permettre d’ordonner par ordre décroissant et pour finir le "LIMIT 5" va permettre de sélection seulement les 5 premier résultats.

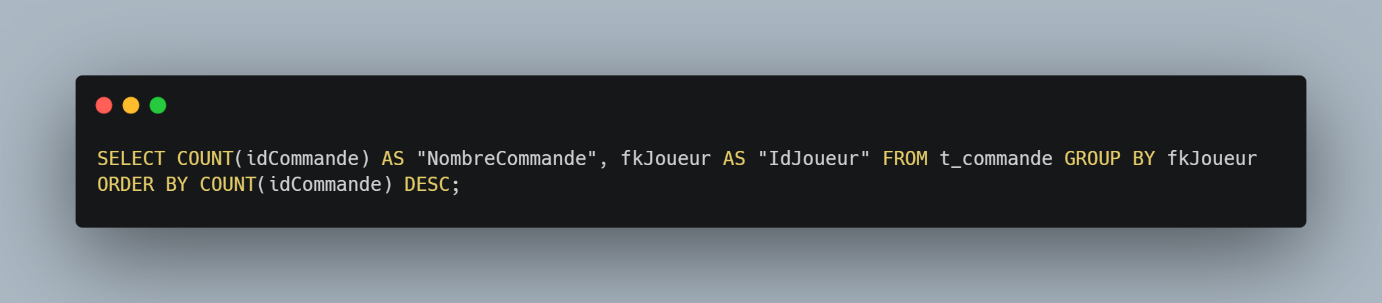
## 4.3 : Requête N°2

Voici ma requête :



Sur cette requête je sélectionne le prix le plus haut, le plus bas et Moyen des armes grâce au fonctions MAX(), MIN(), AVG() je renomme ensuite les colonnes par les noms imposés grâce au alias (AS) depuis la table t\_arme.

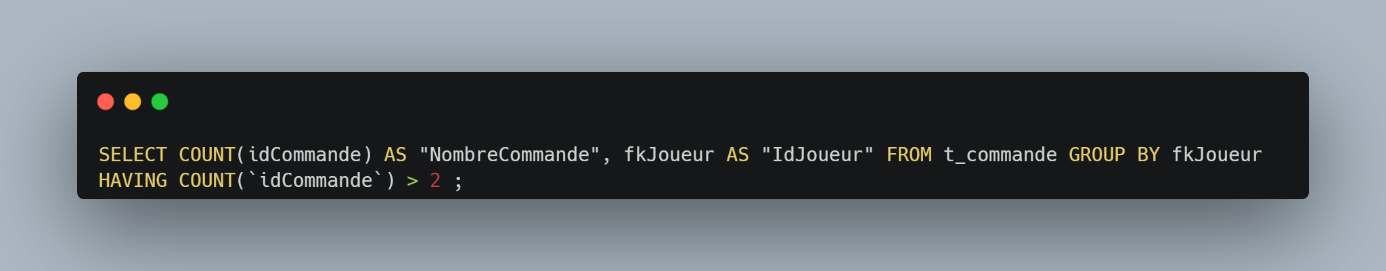
## 4.4 : Requête N°3

Voici la requête que j’ai utilisé : 

Sur cette commande je sélectionne le nombre de commande que chaque joueur à fait grâce à la fonction COUNT() et les joueurs les colonnes je les renomme avec des alias (AS) ensuite je regroupe le tout par joueur (GROUP BY) et ensuite j’ordonne le tout par le nombre de commande du plus grand au plus petit

## 4.5 : Requête N°4

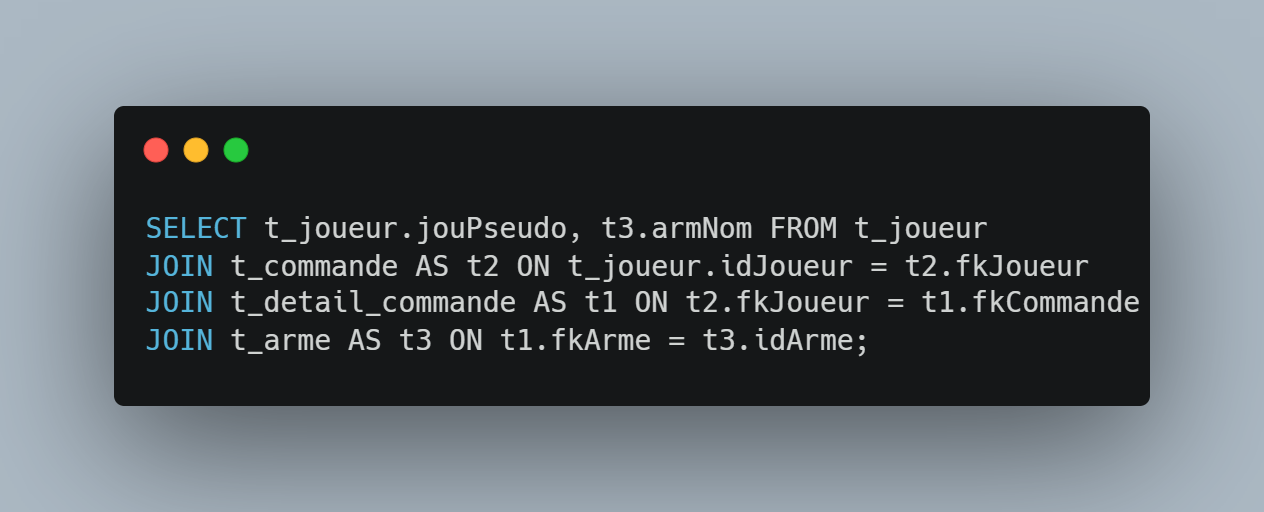
Voici la commande que j’ai utilisé :



Sur cette requête les début et pareil que pour la requête N°3 sauf que en plus de regrouper le tout par joueur je regroupe par les joueur dont le nombre de commande et plus grand que deux si j’ai pu faire cette commande c’est grâce à la clause "HAVING" qui va mettre une condition sur chaque groupement

## 4.6 : Requête N°5

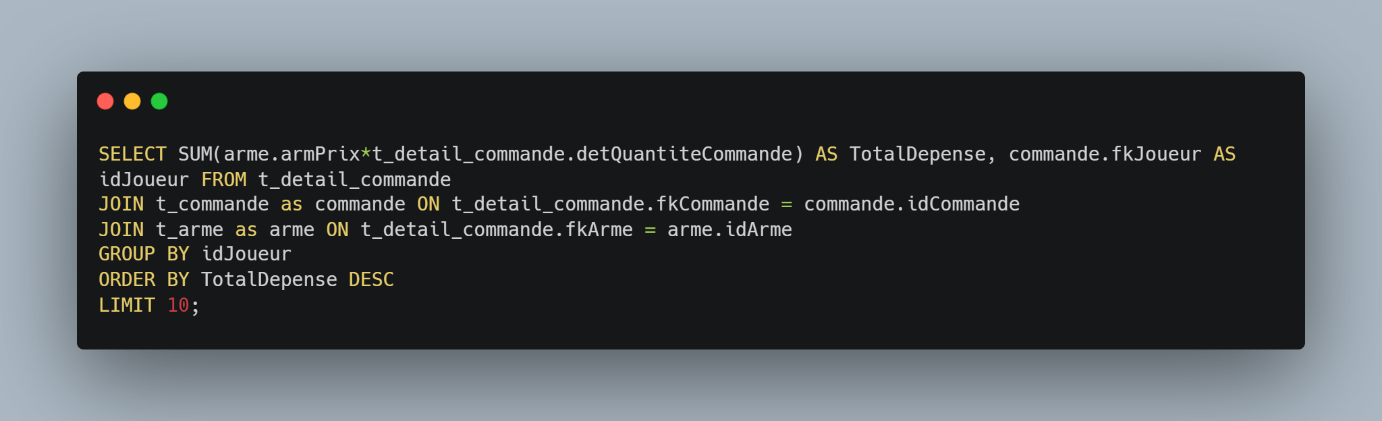
Pour cette question voici la requête que j’ai utilisé :



Pour cette requête je sélectionne le pseudo des joueur le nom des armes de la commande ensuite fais des jointure entre les différente table pour pouvoir utilisé leurs données, j’ai décidé de faire un inner join (JOIN) car je veux seulement les joueur qui ont passé une commande

## 4.7 : Requête N°6

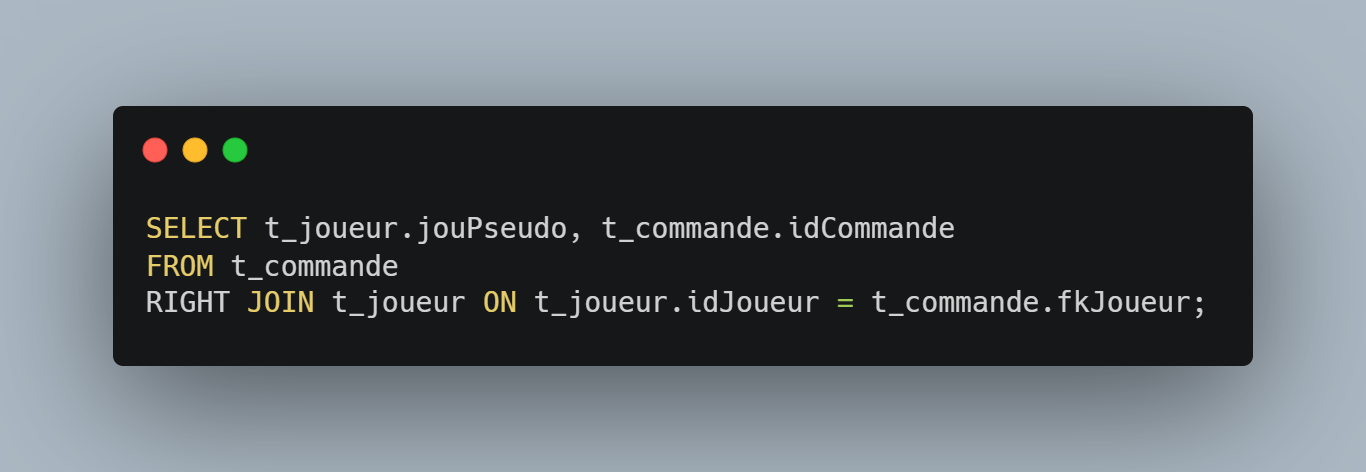
Voici ma requête :



Sur cette requête je sélectionne le total dépensé un calculant la somme du prix de l’arme \* quantité d’arme dans la commande je nomme ensuite la colonne « TotalDepense » ensuite je sélectionne les joueur de la table commande je renomme la colonne en « idJoueur » je joins mes table pour pouvoir utilisé les données des différentes tables ensuite je fais un GROUP BY pour connaitre le total de chaque joueur ensuite d’ordonne mes résultat par ordre décroissant en fonction du total dépensé et pour finir je prends que les 10 premier résultats

## 4.8 : Requête N°7

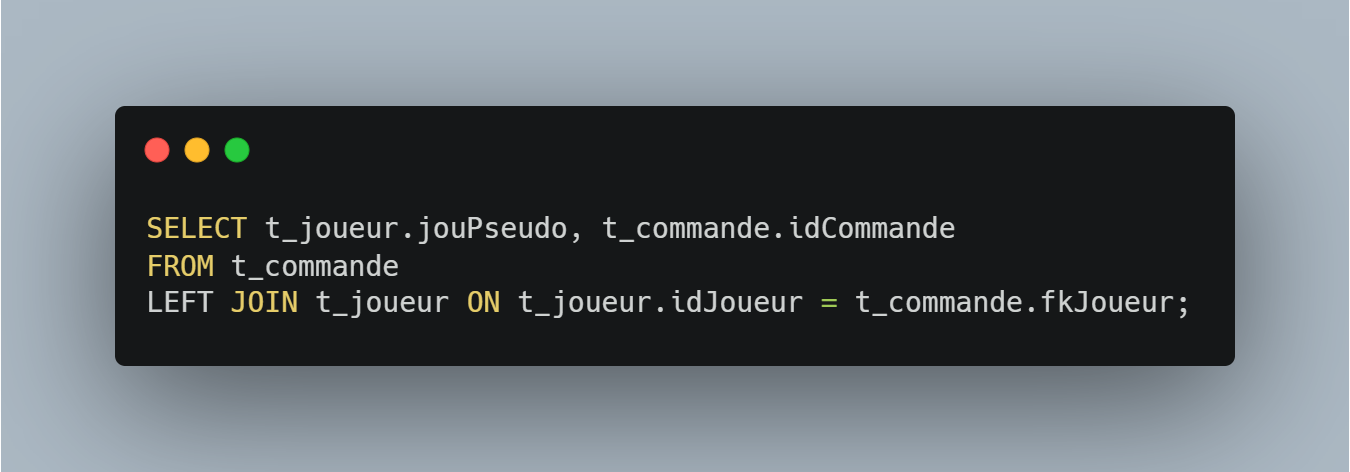
Voici la requête que je trouvé :



Sur cette requête je sélectionne le pseudo des joueurs ainsi que l’id de leurs commandes ensuite je joina avec un right join car je veux aussi les joueurs qui n’ont pas passé de commande

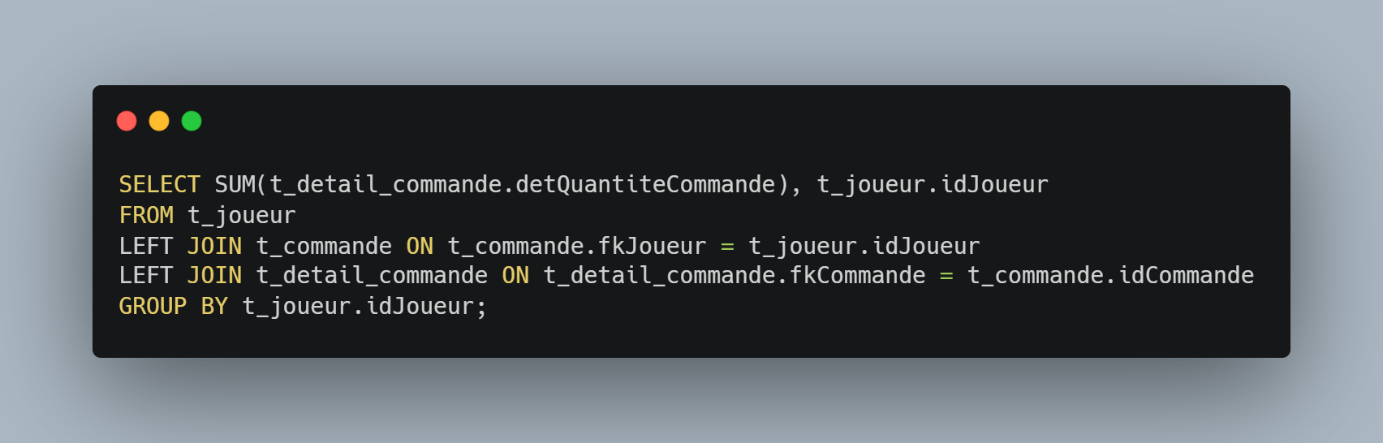
## 4.9 : Requête N°8

Voici la requête :



La requête est la même que dans la requête N°7 mais au lieu de faire un right join je fais un left join car je veux afficher les commandes si elles existent ou pas

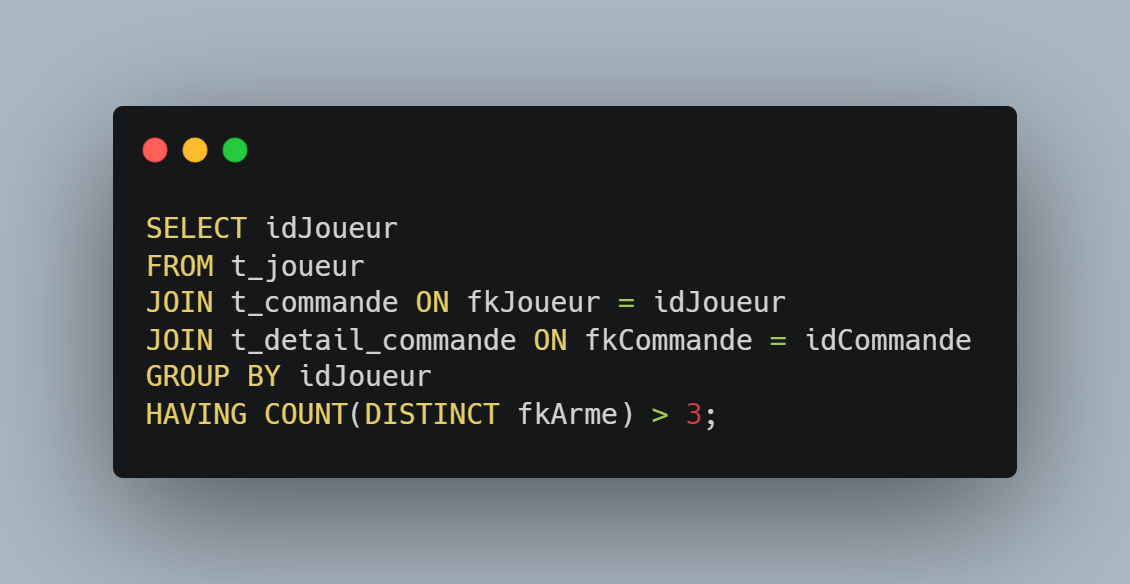
## 4.10 : Requête N°9

Voici ma requête : 

Dans cette requête je sélectionne le nombre total de commande qu’un joueur a passé ensuite je joins les tables entre elles avec un left join car je veux aussi les joueur qui n’ont pas passé de commande ensuite je trie les tout par joueur.

## 4.11 : Requête N°10

Voici la requête :



Sur cette requête je sélection l’id des joueurs ensuite je joins me table entres elles avec un JOIN (INNER JOIN) ensuite je trie les tout par joueur mais seulement si le nombre d’arme differente (COUNT(DISTINCT fkArme)) et plus grand que trois

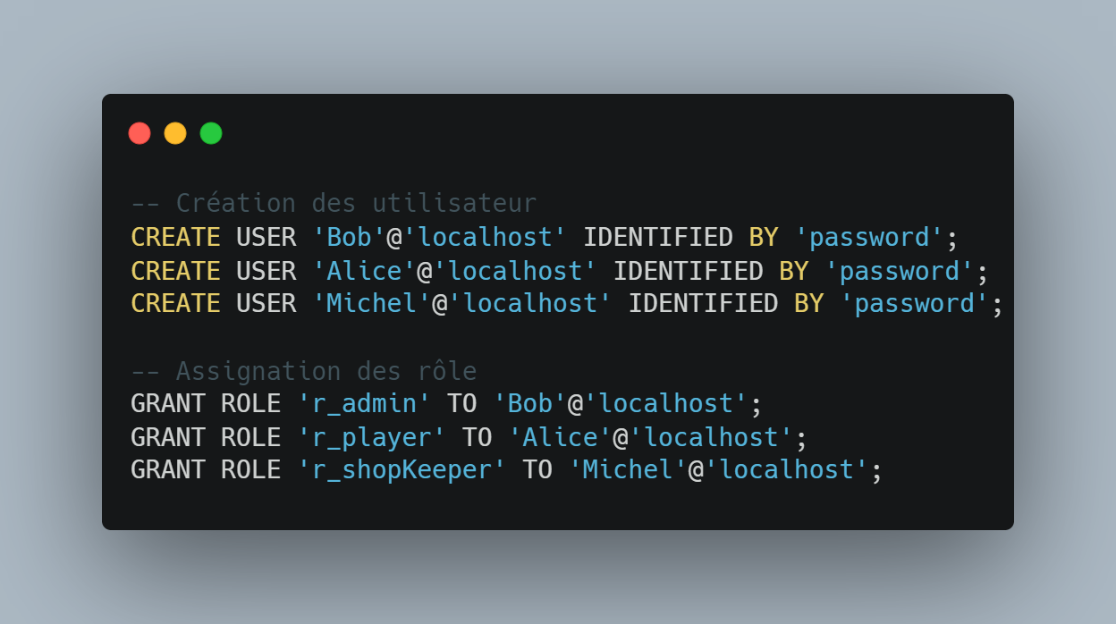
## 4.12 : Création des types d’utilisateur et utilisateur

Pour cette partie on nous a demandé de donner des permissions a des types d’utilisateurs mais pour ce faire nous devons d’abord créer ces types d’utilisateur j’ai alors utilisé ces trois requêtes :



J’ai utilisé des rôles pour pouvoir représenter mes types d’utilisateur.

J’ai aussi dû utiliser créer un utilisateur pas type voici les requêtes que j’ai utilisé pour les créer et assigné les utilisateurs à leurs rôles :

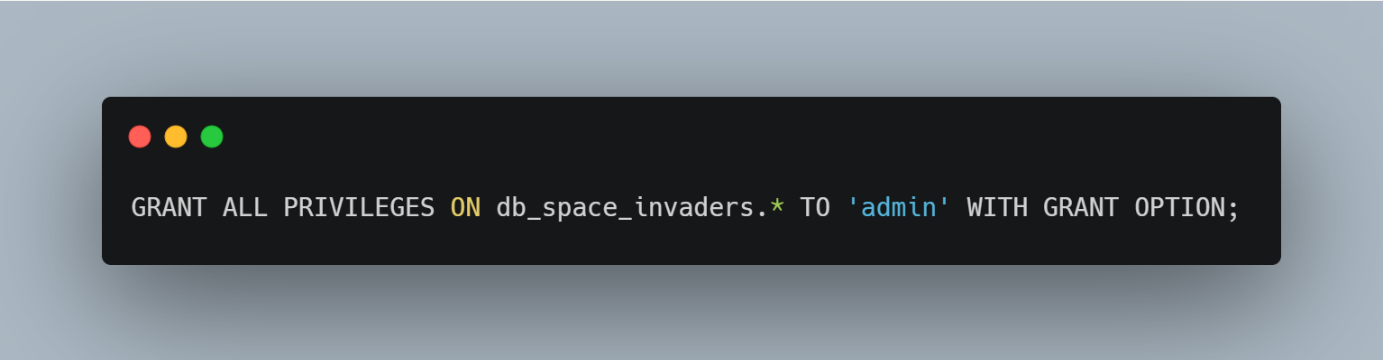


Dans la première partie je créer les utilisateurs dans le host "localhost" et je leur donne le mot de passe "password" grâce au "IDENTIFIED BY". Ensuite j’assigne les utilisateur au rôle que j’ai envie.

## 4.13 : Assignation des permissions

### 4.13.1 : Rôle "r\_admin"

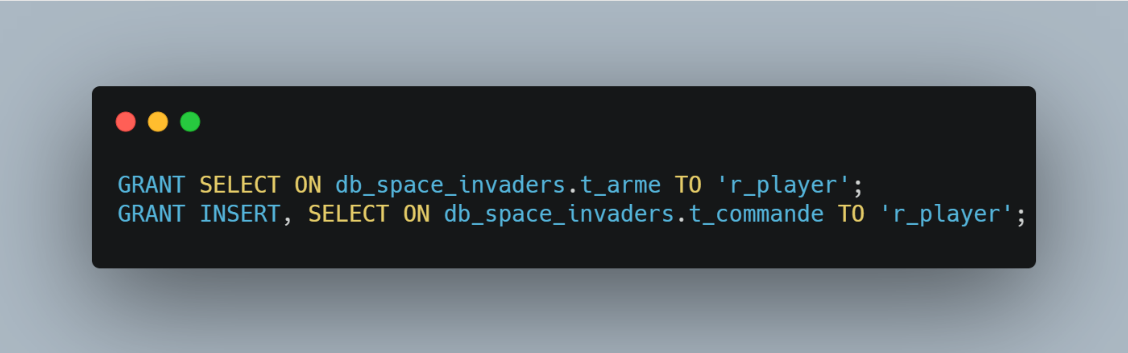
Voici la requête que j’ai utilisé pour donner les permissions d’écriture, lecture, suppression, mis à jour, gestion des permissions pour ce rôle :



Sur cette requête je donne toutes les permissions au rôle (select, insert, …) sur la base de donnée db\_space\_invaders avec les permission de pouvoir gérer les utilisateur

### 4.13.2 : Rôle "r\_player"

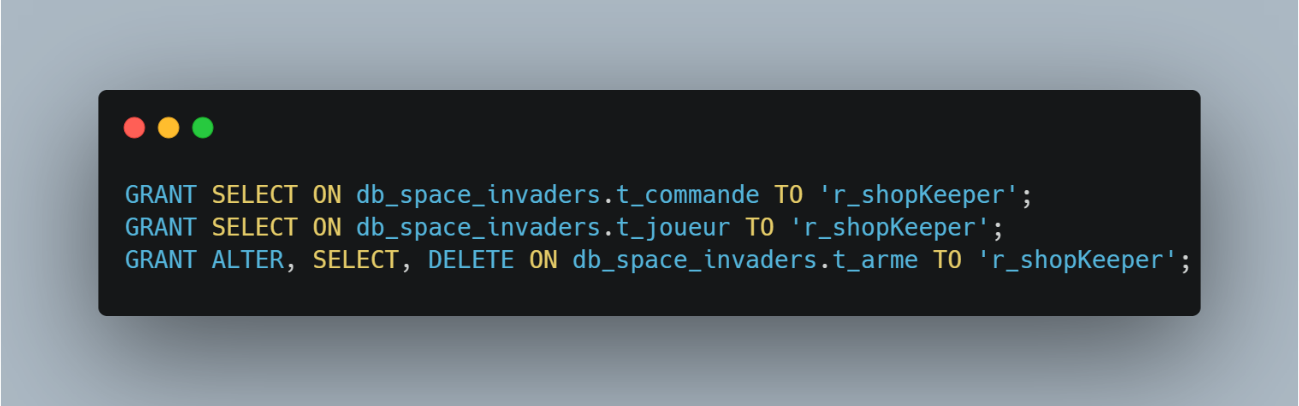
Voici la requête que j’ai utilisé pour donner les permissions de Lire les informations des armes, créer une commande et lire toutes les commandes :



Sur la première requête je lui donne uniquement la permission de sélectionner les données de la table t\_arme. Ensuite je lui permet d’insérer et de sélectionner uniquement sur la table t\_commande comme demandé.

### 4.13.3 : Rôle "r\_shopKeeper"

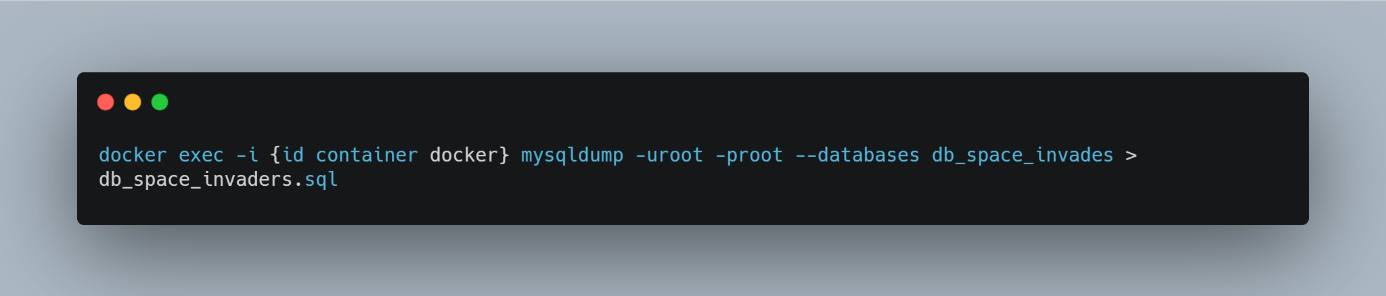
Voici la requête que j’ai utilisé pour donner les permissions de Lire les informations sur tous les joueurs, mettre à jour, lire et supprimer des armes et lire toutes les commandes :



En premier lieu je lui donne les permissions de lire les informations sur les tables t\_commande (pour pouvoir lire toutes les commandes) et sur la table t\_joueur (pour pouvoir lire toutes les informations d’un joueur), ensuite je lui donne les permission de modifier, lire et supprimer sur la table t\_armes pour pouvoir gérer la vente des armes.

## 4.14 : exporter / importer la base de données

Pour exporter un dump de la base de données voici la commande que j’ai executé dans l’invite de commande windows :



Dans cette commande d’abord j’ouvre un terminale docker grâce au « docker exec -i {id container docker} » ensuite j’exécute la commande mysqldump qui va exporter ma base de donné avec l’utilisateur root le « --databases » va permettre de mettre la création et l’utilisation de la base de donnée dans le script car par défaut ils n’y sont pas ensuite je dis la base de donnée que je veux executé et ensuite je donne le chemin d’accès où le fichier va se créer.

Pour restaurer le dump de la base de donnée j’ai dû exécuter cette commande toujours dans le terminal Windows :



Dans cette commande d’abord j’ouvre un terminale docker grâce au « docker exec -i cc94…» ensuite j’exécute la commande mysql qui va restaurer ma base de donné avec l’utilisateur root ensuite j’indique le chemin d’accès du fichier .sql.