Projet SpicyInvaders

Table des matières

[7.3.2 Spécificités UX 1](#_Toc148086911)

[1. Persona 1](#_Toc148086912)

[a. Contexte 1](#_Toc148086913)

[b. Données récoltés 1](#_Toc148086914)

[c. Constat grâce à mes Personas 1](#_Toc148086915)

[2. Palette graphique 2](#_Toc148086916)

[3. Eco-conception 2](#_Toc148086917)

[7.3.3 Spécificités POO 3](#_Toc148086918)

[Introduction 3](#_Toc148086919)

[Analyse fonctionnelle 3](#_Toc148086920)

[Analyse technique 3](#_Toc148086921)

[Diagramme de base 3](#_Toc148086922)

[Explications (docfx) 3](#_Toc148086923)

[Test Unitaire 3](#_Toc148086924)

[Conclusion 3](#_Toc148086925)

[7.3.4 Spécificités DB 4](#_Toc148086926)

[A.Importer les données et le schéma de base de données 5](#_Toc148086927)

[B. Gestion des utilisateurs 5](#_Toc148086928)

[1. Administrateur du jeu 5](#_Toc148086929)

[2. Joueur 5](#_Toc148086930)

[3. Gestionnaire de la boutique 5](#_Toc148086931)

[4. Comment implémenter cela 5](#_Toc148086932)

[C. Réaliser et expliquer en détail les requêtes 5](#_Toc148086933)

[Requête n°1 : 5](#_Toc148086934)

[Requête n°2 : 5](#_Toc148086935)

[Requête n°3 : 5](#_Toc148086936)

[Requête n°4 : 5](#_Toc148086937)

[Requête n°5 : 5](#_Toc148086938)

[Requête n°6 (TODO): 5](#_Toc148086939)

[Requête n°7 : 5](#_Toc148086940)

[Requête n°8 : 5](#_Toc148086941)

[Requête n°9 : 5](#_Toc148086942)

[Requête n°10 (TODO): 5](#_Toc148086943)

[D. Création des index 5](#_Toc148086944)

[1. Pourtant certains index existent déjà. Pourquoi ? 5](#_Toc148086945)

[2. Quels sont les avantages et les inconvéniants des index ? 5](#_Toc148086946)

[3. Sur quel champ (de quelle table), cela pourrait être pertinant d’ajouter un index ? 5](#_Toc148086947)

[E. Backup / Restore 5](#_Toc148086948)

[8.2 Utilisation d’IA dans le projet 5](#_Toc148086949)

[Partie POO 5](#_Toc148086950)

[Partie DB 5](#_Toc148086951)

[Partie UX 5](#_Toc148086952)

[Autre 5](#_Toc148086953)

# 7.3.2 Spécificités UX

## 1. Persona

### a. Contexte

Je crée d’abord les persona afin de savoir comment créer mes maquettes par la suite. Je crée un petit nombre de persona, 2, afin de pouvoir en créer plus par la suite si j’ai des idées intéressantes.

Pour la création de mes Personas j’ai utilisé le Template de Jingdi Ma : <https://www.figma.com/community/file/994974628735380661>

### b. Données récoltés

Je récupérerai l’âge, le genre, le domicile, le travail et le statut familial en premier.

Je créerai par la suite une biographie de mon persona à l’aide de ChatGPT.

À l’aide de graphique j’estime à quel point mon persona apprécie certains aspects du jeu tel que :

* Faire les meilleures scores que possibles
* Atteindre le plus haut niveau possible
* Explorer des modes de jeu alternatif
* Faire des défis (de la communauté)
* Passer le temps

Je cherche ce que souhaite et ne souhaite pas rencontrer mon persona sur un jeu de Space Invaders.

J’établie un pourcentage des appareils utilisés pour jouer à Space Invaders afin de savoir comment adapter mon jeu.

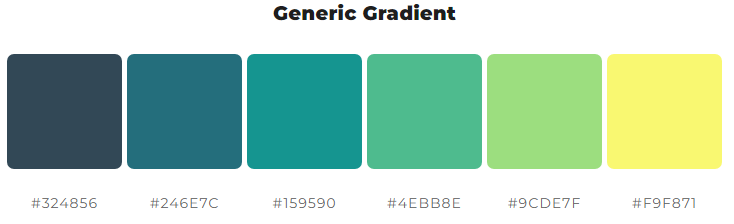
### c. Constat grâce à mes Personas

Mon premier persona, Robbie Heineman me fait prendre conscience que le premier jeu Space Invaders étant sorti en 1978 il peut s’adresser à des personnes actuellement âgé et qu’il vaut donc mieux ne pas créer trop d’effet spéciaux pour ne pas déranger l’utilisateur. De plus les jeux Space Invaders étant sorti sur arcade il pourrait être utile d’avoir des contrôles imitant les bornes d’arcade. D’anciens joueurs pourraient aussi souhaiter avoir une version original du jeu.

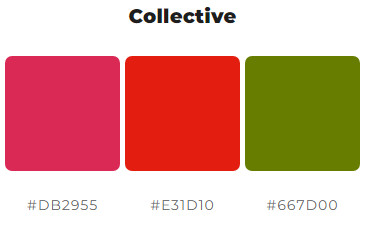
Mon second persona, Martine Dupont, me fait prendre conscience que mes utilisateurs pourraient jouer à Space Invaders depuis leur téléphone. Ils pourraient vouloir jouer à plusieurs en local comme en multijoueur. De nouveau joueur pourrait souhaiter avoir facilement accès à des modifications sur leur partie, des niveaux afin qu’une communauté puisse proposer des ajouts sur le jeu, il pourrait donc être utile que le jeu soit ouvert à la création de mod.

## 2. Palette graphique

Pour la palette de couleur je souhaitais effectuer un contraste entre la morosité de l’espace et l’intensité humaine. J’ai ainsi récupéré une gradation de couleur montrant une transformation de la morosité (teinte sobre) jusqu’à la chaleur humaine. Ainsi du début à la fin de ma palette de couleur l’on aperçoit premièrement l’espace puis l’eau caractérisant ainsi notre planète. Par la suite la verdure et finalement l’humain et la chaleur de ses émotions. C’est ainsi un zoom depuis l’immensité de l’espace jusqu’à l’humanité.



J’ai aussi créé une seconde palette de couleur destiné au daltonien ayant des contrastes de couleurs plus distingué.



## 3. Eco-conception

Chercher parmi les 115 bonnes pratiques d’éco-conception.

# 7.3.3 Spécificités POO

## Introduction

TODO : pas de nom personnel

Dans le cadre d’apprentissage de l’ETML nous sommes amenés à effectuer un projet de programmation orienté objet. Dans ce projet je dois créer une réplique du célèbre jeu Space Invaders en version console ou Forms. Pour ma part j’ai choisi de créer mon projet en version console.

Les objectifs de ce projet sont :

* Créer un vaisseau capable de se déplacer et de tirer.
* Créer 10 ennemies se déplaçant sur l’axe vertical.

Pour la gestion du projet c’est IceScrum qui est utilisé. Toutes les Stories doivent être validés par le chef de projet avant de commencer à implémenter ce que décrit la Story. Chaque Story contient un/des tests d’acceptance pouvant être accompagné d’une maquette. Tout le projet se fait en un seul sprint.

Le projet doit contenir des tests unitaires afin de tester la fonctionnalité de chaque fonction.

## Analyse fonctionnelle

## Analyse technique

### Diagramme de base

### Explications (docfx)

## Test Unitaire

## Conclusion

Pour conclure, dans le cadre de ce projet j’ai appris des notions de base en programmation orienté objet et comment les utiliser en pratique. J’ai également appris à mieux utiliser Visual Studio 2022 que ça soit pour ajouter des bibliothèques de classe.

Les problèmes que j’ai rencontrés :

Beaucoup de temps passé à faire des stories, j’aurai dû faire une story créé des story pour savoir réellement le temps passé

# 7.3.4 Spécificités DB

## A. Importer les données et le schéma de base de données

Création de la base de données :

docker exec -i db mysql -uroot -proot < db\_space\_invaders.sql

TODO :

* Expliquer la connexion entre DB et C# (Nuggets, etc.).
* Affichage des 5 meilleures scores dans le programme C#.
* Save le score à la fin d’une partie dans la DB.
* Modification du CDC par MMN (B 3)
* En-tête du document et pied de page

## B. Gestion des utilisateurs

### 1. Administrateur du jeu

*Peut créer, lire, mettre à jour et supprimer (CRUD) n’importe quelle table.*

*Gérer les utilisateurs et leurs privilèges.*

CREATE ROLE `administrateur\_du\_jeu`;

GRANT CREATE, [SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html), [UPDATE](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html), [DELETE](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) ON \*.\* TO `administrateur\_du\_jeu`;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Création de l’utilisateur dans le rôle `Admin-jeu`.

CREATE [USER](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/information-functions.html%23function_user) 'admin-Leo'@'localhost' IDENTIFIED BY 'Password';

GRANT 'Admin-jeu' TO 'admin-Leo'@'localhost';

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des requêtes :

La première commande me permet de créer un rôle nommé « Admin-jeu ».

La seconde commande donne les privilèges CREATE, SELECT, UPDATE et DELETE sur toutes les tables de toutes les bases de données au rôle ‘administrateur\_du\_jeu’.

La troisième commande crée un utilisateur nommé ‘Admin-Leo’ en localhost et lui donne le mot de passe ‘Password’.

La quatrième commande donne le rôle ‘administrateur\_du\_jeu’ à l’utilisateur admin-Leo.

### 2. Joueur

*Lire les informations des armes (pour voir quelles armes il peut acheter).*

*Créer une commande.*

*Lire toutes les commandes.*

CREATE ROLE `Player`;

GRANT [SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) ON db\_space\_invaders.t\_arme TO `Player`;

GRANT CREATE ON db\_space\_invaders.t\_commande TO `Player`;

GRANT [SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) ON db\_space\_invaders.t\_commande TO `Player`;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Création de l’utilisateur dans le rôle `Player` .

CREATE [USER](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/information-functions.html%23function_user) 'player-Leo'@'localhost' IDENTIFIED BY 'Password';

GRANT 'Player' TO 'player-Leo'@'localhost';

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des commandes :

La première commande crée le rôle ‘Player’.

La seconde commande donne le droit SELECT au rôle ‘Player’ dans la table ‘t\_commande’ de la base de données ‘db\_space\_invaders’.

La troisième commande donne le droit CREATE au rôle ‘Player’ dans la table t\_commande de la base de données ‘db\_space\_invaders’.

La quatrième commande donne le droit SELECT au rôle ‘Player’ sur la table ‘t\_commande’ de la base de données ‘db\_space\_invaders’.

La cinquième commande crée l’utilisateur ‘player-Leo’ avec l’Host ‘localhost’ et avec le mot de passe ‘Password’.

La sixième commande ajoute l’utilisateur ‘Player-Leo’ au rôle ‘Player’

### 3. Gestionnaire de la boutique

*Lire les informations sur tous les joueurs (pour savoir qui a passé une commande).*

*Mettre à jour, lire et supprimer des armes (ajout de nouvelles armes, modification des prix, etc.).*

*Lire toutes les commandes*

CREATE ROLE `ShopKeeper`;

GRANT [SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) ON db\_space\_invaders.t\_joueur TO `ShopKeeper`;

GRANT [UPDATE](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/update.html), [SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html), [INSERT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/insert.html), [DELETE](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/delete.html) ON db\_space\_invaders.t\_arme TO `ShopKeeper`;

GRANT [SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) ON db\_space\_invaders.t\_commande TO `ShopKeeper`;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Création de l’utilisateur dans le rôle `shopkeeper`.

CREATE [USER](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/information-functions.html%23function_user) 'shopkeeper-Leo'@'localhost' IDENTIFIED BY 'Password';

GRANT 'shopkeeper' TO 'shopkeeper-Leo'@'localhost';

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des commandes :

La première commande crée le rôle ‘ShopKeeper’.

La seconde commande donne le droit ‘SELECT’ au rôle ‘ShopKeeper’ sur la table ‘t\_joueur’ de la base de données ‘db\_space\_invaders’.

La troisième commande donne les droits ‘Update’, ‘SELECT’, ‘INSERT’ et ‘DELETE’ au rôle ‘ShopKeeper’ sur la table ‘t\_arme’ de la base de données ‘db\_space\_invaders’.

La quatrième commande donne le droit ‘SELECT’ au rôle ‘ShopKeeper’ sur la table ‘t\_commande’ de la base de données ‘db\_space\_invaders’.

La cinquième commande crée l’utilisateur ‘shopkeeper-Leo’ avec l’host ‘localhost’ et le mot de passe ‘Password’.

La sixième commande ajoute l’utilisateur ‘shopkeeper-Leo’ au rôle ‘shopkeeper’.

### 4. Comment implémenter cela

Comme vu sur les 3 points précédents l’implémentation de rôle pour la gestion des utilisateurs se fait ainsi.

Tout d’abord la création du rôle. Pour cela la commande CREATE ROLE ‘nom\_du\_role’ ; permet de créer un rôle.

Par la suite l’on peut donner le droit voulu au rôle grâce à la commande GRANT suivi du nom du droit voulu. Si l’on souhaite ajouter plusieurs droits l’on peut utiliser une virgule suivi du prochain droit. Après avoir spécifié le droit l’on écrit « ON » puis le nom de la base de données s’il y en a une de particulier suivi d’un point puis du nom de la table s’il y en a une de particulier. S’il n’y a aucune table ou base de données où le droit prendra effet en particulier alors on peut utiliser le caractère joker ‘\*’. Ainsi si le droit n’est pas spécifique à un lieu en particulier l’on peut très bien écrire « \*.\* » pour échapper à cette restriction. L’on spécifie par la suite à qui l’on donne ce droit avec un « TO ‘nom\_du\_role\_ou\_de\_l’utilisateur’ ; ».

Et en dernier il faut assigner des utilisateurs au droits créé. Pour cela l’on utilise à nouveau la commande « GRANT » suivi du nom du rôle puis de « TO » le nom de l’utilisateur « @ » l’host de l’utilisateur « ; ».

## C. Réaliser et expliquer en détail les requêtes

### 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*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) \* FROM `t\_joueur` ORDER BY jouNombrePoints DESC LIMIT 5;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des nouveaux éléments de la requête :

SELECT \* FROM `t\_joueur` : permet de sélectionner tous les attributs de la table `t\_joueur`.

ORDER BY jouNombrePoints DESC : le DESC permet d’afficher les valeurs dans l’ordre décroissant du ORDER BY selon les valeurs dans jouNombrePoints.

LIMIT 5 ; : me permet de n’afficher que les 5 premiers résultats et le ; me permet de terminer la commande.

### Requête n°2 :

*Trouver le prix maximum, minimum et moyen des armes.*

*Les colonnes doivent avoir pour nom « Prix Maximum », « PrixMinimum » et « PrixMoyen »*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [MAX](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_max)(armPrix) AS Maximum, [MIN](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_min)(armPrix) AS Minimum, [AVG](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(armPrix) AS Moyenne FROM `t\_arme`;

Une image contenant texte, capture d’écran, carte de visite, Police

Description générée automatiquement

**Explication** des nouveaux éléments de la requête :

SELECT Max(armPrix) AS Maximum  : permet d’afficher la plus grande valeur dans l’attribut armPrix. La virgule permet de sélectionner d’autre valeurs.

MIN(armPrix) AS Minimum  : me permet d’afficher le plus petit résultat dans l’attribut armPrix. La virgule permet de sélectionner d’autre valeurs.

AVG(armPrix) AS Moyenne FROM `t\_arme` ; : me permet d’afficher la moyenne des données dans l’attribut armPrix.

### 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".*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DISTINCT `fkJoueur` AS IdJoueur, [Count](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(`fkJoueur`) AS NombreCommandes FROM t\_commande GROUP BY IdJoueur;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des nouveaux éléments de la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DISTINCT `fkJoueur` AS IdJoueur, Permet d’afficher les valeurs de fkJoueur une seule fois par valeur.

[Count](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(`fkJoueur`) AS NombreCommandes Permet de compter le nombre de valeur dans fkJoueur.

FROM t\_commande GROUP BY IdJoueur; Permet de sélectionner la table t\_commande puis grouper par l’identifiant du joueur afin de calculer le nombre de commande par joueur et non pas au total.

### 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 "*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) DISTINCT `fkJoueur` AS IdJoueur, [COUNT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_count)(`fkJoueur`) AS NombreCommandes FROM t\_commande GROUP BY IdJoueur HAVING NombreCommandes > 2;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des nouveaux éléments de la requête :

Ceci est la même commande que celle de la requête 3 hormis l’utilisation de

 HAVING NombreCommandes > 2

Avec ceci on ne sélectionne que les lignes qui ont un nombre de commande supérieur à 2.

### Requête n°5 :

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

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) t\_joueur.jouPseudo, t\_arme.armNom FROM t\_joueur INNER JOIN t\_commande ON t\_joueur.idJoueur = t\_commande.fkJoueur INNER JOIN t\_detail\_commande ON t\_commande.idCommande = t\_detail\_commande.fkCommande INNER JOIN t\_arme ON t\_detail\_commande.fkArme = t\_arme.idArme;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** des nouveaux éléments de la requête :

INNER JOIN t\_commande ON t\_joueur.idJoueur = t\_commande.fkJoueur  Le INNER JOIN permet de créer une jointure entre la table t\_commande et la table t\_joueur. Cela permet de sélectionner des éléments sur plusieurs tables. Le ON par la suite permet de ne prendre **que** les valeurs sélectionner auparavant ou la clef primaire de t\_joueur(idJoueur)et la clef étrangère de t\_commande sont équivalente. La clef étrangère étant le référencement entre les 2 tables.

Dans cet exercice j’utilise un INNER JOIN pour ne récupérer que les valeurs reliés à d’autres tables.

### Requête n°6 (TODO):

*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".*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) t\_joueur.idJoueur AS IdJoueur, [SUM](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(armPrix) AS TotalDepense FROM t\_arme INNER JOIN t\_arsenal ON t\_arme.idArme = t\_arsenal.fkArme INNER JOIN t\_joueur ON t\_arsenal.fkJoueur = t\_joueur.idJoueur GROUP BY IdJoueur LIMIT 10;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** de la commande :

Utilisation de 2 INNER JOIN afin de relier 2 tables qui sont séparés par une table. Dans ce cas la table arsenal est entre la table joueur et arme, elle contient les clefs privés de la table joueur et de la table arme.

### 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’,*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) idJoueur, t\_commande.idCommande, t\_commande.comDate, t\_commande.comNumeroCommande FROM t\_joueur [LEFT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left) JOIN t\_commande ON t\_joueur.idJoueur = t\_commande.fkJoueur;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** de la commande :

Utilisation de [LEFT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left) JOIN pour faire une jointure au lieu d’un INNER JOIN afin de récupérer tous les joueurs même s’ils n’ont aucun lien avec la table commande.

### Requête n°8 :

*Récupérer toutes les commandes et afficher le pseudo du joueur si elle existe, sinon montrer ‘NULL’ pour le pseudo.*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) idCommande, comDate, comNumeroCommande, t\_joueur.jouPseudo FROM t\_commande [LEFT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left) JOIN t\_joueur ON t\_commande.fkJoueur = t\_joueur.idJoueur;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** de la commande :

Utilisation d’un [LEFT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/string-functions.html%23function_left) JOIN afin de récupérer toutes les valeurs de la table commande même si elles ne sont pas liées à la table joueur. De cette façon si la commande n’est reliée à aucun joueur elle sera quand même affichée et le nom du joueur restera ‘NULL’.

### Requête n°9 :

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

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) [SUM](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(`detQuantiteCommande`) AS nbr\_commandes, t\_joueur.idJoueur FROM t\_detail\_commande INNER JOIN t\_commande ON t\_detail\_commande.fkCommande = t\_commande.idCommande INNER JOIN t\_joueur ON t\_commande.fkJoueur = t\_joueur.idJoueur GROUP BY t\_joueur.idJoueur;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** de la commande :

Afin de sélectionner le nombre total d’arme acheté je fais un [SUM](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum) de la quantité commandé car une commande peut comporter plusieurs armes. Par la suite je rejoins les tables nécessaires pour pouvoir accéder de la table t\_detail\_commande dans laquelle il y a le nombre de commande et la table t\_joueur ou je peux obtenir l’identifiant et le nom du joueur.

### Requête n°10 (TODO):

*Trouver les joueurs qui ont achetés plus de 3 types d’armes différentes.*

**Réaliser** la requête :

[SELECT](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/select.html) t\_joueur.jouPseudo, [SUM](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(t\_detail\_commande.detQuantiteCommande) AS nbr\_arme\_diffrente\_achete FROM t\_joueur INNER JOIN t\_commande ON t\_joueur.idJoueur = t\_commande.fkJoueur INNER JOIN t\_detail\_commande ON t\_commande.idCommande = t\_detail\_commande.fkCommande GROUP BY t\_joueur.jouPseudo HAVING [SUM](http://localhost:8081/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_sum)(t\_detail\_commande.detQuantiteCommande) > 3;

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

**Explication** de la commande :

## D. Création des index

### 1. Pourtant certains index existent déjà. Pourquoi ?

MySQL crée automatiquement des indexes sur les clefs primaires et sur les clefs étrangères.

Ces indexes sont créés automatiquement car les clefs primaires et les clefs étrangères sont fréquemment utilisé lors de requêtes, spécifiquement lorsque l’on souhaite créer des jointures

### 2. Quels sont les avantages et les inconvénients des index ?

**Avantages :**

Les indexes permettent de structurer les données afin d’effectuer des recherches dans les données beaucoup plus rapidement.

**Inconvénients :**

Les indexes prennent de la place en mémoire

Ils ralentissent les requêtes car l’index doit se remettre à jour à chaque changement :

* D’insertion
* De modification
* De suppression

### 3. Sur quel champ (de quelle table), cela pourrait être pertinent d’ajouter un index ?

Sur les clefs primaires et les clefs étrangères (dans le cas où elle n’ont pas été automatiquement créé par le SGBDR utilisé). Cela permettrait d’effectuer des requêtes plus rapidement lorsque l’on souhaite effectuer des jointures. De plus des indexes peuvent être rajoutés.

## E. Backup / Restore

**Backup** de la base de données :

Mysqldump -uroot -proot nom\_db > nom.sql

Mysqldump -u root -p nom\_db > nom.sql

**Explication** de la Backup

Mysqldump permet de spécifier que c’est l’utilitaire « Mysqldump » qui est utilisé.

-uroot spécifie que l’Utilisateur est ROOT

-p spécifie que l’on va rentrer le « *password* » (mot de passe). Rien n’est mis à la suite afin de ne pas le divulguer.

**Restore** la base de données :

Mysql -uroot -p < nom.sql

**Explication** du Restore :

# 8.2 Utilisation d’IA dans le projet

## Partie POO

Dans le cadre programmation orienté objet de ce projet je n’ai pas utilisé d’IA. En cas de difficulté j’ai préféré m’orienté sur des forums ou demander de l’aide à des connaissances.

## Partie DB

## Partie UX

## Autre

J’ai utilisé l’IA « ChatGPT » (<https://chat.openai.com/>) afin de savoir comment faire tenir mon sommaire sur une seule page.