|  |
| --- |
| ETML |
| Rapport de Projet SpicyInvaders |
| CID2B |

|  |
| --- |
| Mateen Salem Khaiy  06/10/2023 |

Table des matières

[UX – User Expérience 2](#_Toc148994341)

[DB – Data base 2](#_Toc148994342)

[A. Importer les données et le schéma de base de données 2](#_Toc148994343)

[B. Gestion des utilisateurs 2](#_Toc148994344)

[Administrateur : 2](#_Toc148994345)

[Joueur : 2](#_Toc148994346)

[Gestionnaire de la boutique : 2](#_Toc148994347)

[C. Requêtes de sélection 2](#_Toc148994348)

[D. Création des index 2](#_Toc148994349)

[E. Backup / Restore 2](#_Toc148994350)

[Backup 2](#_Toc148994351)

[Restore 2](#_Toc148994352)

[POO – Programmation orientée objet 3](#_Toc148994353)

[A. Introduction 3](#_Toc148994354)

[B. Analyse fonctionnelle 3](#_Toc148994355)

[C. Analyse technique 3](#_Toc148994356)

[D. Tests Unitaire 3](#_Toc148994357)

[E. Utilisation ChatGPT 3](#_Toc148994358)

[F. Conclusion 3](#_Toc148994359)

# UX – User Expérience

# DB – Data base

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

Pour pouvoir charger un fichier dump dans une base de données, il faut utiliser les commandes suivantes :

1. En premier lieu il faut utiliser la commande **‘’mysql -u (suivis du nom d’utilisateur) -p (suivis du mot de passe)’’** cette dernière commande permet de se connecter au serveur MySQL avec le nom d’utilisateur choisi, après s’être connecter il est possible d’interagir avec un serveur de base de données.
2. La première partie de la commande, **‘’mysql’’** permet d’indiquer que nous voulons utiliser le client MySQL dans notre invite de commande.
3. Deuxièmement, ‘’-u (suivis du nom d’utilisateur)’’ permet de spécifier en tant que quel utilisateur nous voulons-nous connecter, il est important de noter que le nom d’utilisateur de base crée par MySQL est le super utilisateur **‘’root’’**.
4. En dernier lieu, la commande **‘’-p’’**, permet d’indiquer à MySQL qu’il faut nous demander un mot de passe après la saisie de la commande.
5. Ensuite, la commande à insérer après s’être login dans MySQL est la suivante : Une image contenant texte, capture d’écran, Police

   Description générée automatiquement

Cette commande permet de charger un fichier dump dans le serveur MySQL.

1. Comme dit précédemment, **‘’mysql -uroot -proot’’** permet de se connecter au client MySQL en tant qu’utilisateur **‘’root’’.**
2. Ensuite, ‘’< (chemin où se trouve le dump)’’ permet de rediriger le contenu du fichier qui se trouve après le signe ‘’<’’ dans la base de données.
3. Si le serveur MySQL se situe dans un conteneur docker, la commande à utiliser est la suivante : Une image contenant texte, capture d’écran, Police

   Description générée automatiquement
4. **‘’docker exec’’** permet d’exécuter la commande qui suit, à l’intérieur d’un conteneur docker.
5. La commande : **‘’-i (id du container mysql)’’** permet de spécifier quelle container docker nous voulons utiliser.

## Gestion des utilisateurs

### Administrateur :

1. La commande suivante permet de crée un nouvel utilisateur dans MySQL

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. **‘’CREATE USER’’** permet d’indiquer à mysql que nous voulons crée un nouvel utilisateur.
2. **‘’ 'administrateur\_de\_jeu' ’’** spécifiequel’utilisateur que nous voulons crée vas se nommer adminisateur\_de\_jeu.
3. **‘’ 'localhost' ’’** permet de spécifier à partir de quelle endroit l’utilisateur aura le droit se connecter.
4. **‘’ IDENTIFIED BY 'adminjeu' ’’** cette dernière partie de la commande informe le mot de passe associé à l’utilisateur que nous allons créer, pour se faire le mot de passe est placé entre des guillemets pour informer a mysql qu’il s’agit d’une suite de caractère.
5. La commande suivante permet de crée un rôle nommer ‘’r\_Administrateur\_de\_jeu’’.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La première partie, **‘’CREATE ROLE’’** permet d’indiquer que l’on veut crée un nouveau rôle.
2. Ensuite, **‘’ 'r\_Administrateur\_de\_jeu' ‘’** est le nom du rôle que nous voulons crée.
3. La commande suivante permet d’accorder des privilèges à un utilisateurs ou à un rôle :

Une image contenant capture d’écran, texte, Police

Description générée automatiquement

1. **‘’ 'GRANT' ‘’** cette première partie de la commande informe MySQL que nous voulons accorder des privilèges à un rôle.
2. **‘’ 'SELECT, CREATE, UPDATE, DROP' ‘’** cette partie de la commande spécifie tous les privilèges que l’on veut accorder au rôle, dans l’ordre,

SELECT : permet d’exécuter des requêtes SELECT sur la base de données pour y lire ses données.

CREATE : autorise à l’utilisateur de pouvoir crée des tables.

UPDATE : autorise à l’utilisateur de mettre à jours les données des tables dans la base de données

DROP : Autorise la suppression d’objet dans la base de données.

1. **‘’ 'ON db\_space\_invaders.\* ' ‘’** permet de spécifier sur quelle base de données et sur quelles tables l’utilisateurs pourra jouir de ses privilèges, dans cet exemple les utilisateurs qui auront le rôle pourrons utiliser leurs privilèges sur la base de données ‘’db\_space\_invaders’’, le signe ‘’\*’’ signifie que les privilèges précédemment citer s’applique sur toutes les tables.
2. **‘’TO 'r\_Administrateur\_de\_jeu'‘’** signifie que l’on donne les privilèges au rôle ‘’r\_Administrateur\_de\_jeu’’.
3. **‘’WITH GRANT OPTION’’** cette dernière partie de la commande indique que le rôle ‘’r\_adminisatrateur\_de\_jeu’’ peut déléguer les privilèges SELECT, CREATE, UPDATE et DROP à d’autre utilisateurs ou rôles.
4. La commande suivante permet d’accorder un rôle à un utilisateur :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. **‘’GRANT’’** indique que nous allons accorder un rôle à un utilisateur
2. **‘’ 'r\_Administrateur\_de\_jeu' ‘’** ici nous spécifions quel rôle nous allons accorder par la suite
3. **‘’ TO 'Administrateur\_de\_jeu'@'localhost' ‘’** cette dernière partie de commande indique à quel utilisateur nous allons accorder le rôle.

### Joueur :

1. La commande suivante crée un utilisateur nommer Joueur

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. **‘’CREATE USER’’** indique à MySQL que nous voulons crée un nouvel utilisateur.
2. La commande suivante crée un rôle :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La commande suivante donne le privilège ‘’SELECT’’ sur la table ‘’t\_arme’’ de la base de données ‘’db\_space\_invaders’’ au rôle ‘’r\_joueur’’ :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La commande suivante donne le privilège ‘’SELECT’’ ainsi que ‘’CREATE’’ sur la table ‘’t\_commande’’ de la base de données ‘’db\_space\_invaders’’ au rôle ‘’r\_Joueur’’ :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La commande suivante donne le rôle ‘’r\_Joueur’’ à l’utilisateur ‘’Joueur’’ :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

### Gestionnaire de la boutique :

1. La commande suivante crée l’utilisateur ‘’Gestionnaire\_de\_la\_boutique’’ dont le mot de passe est ‘’gestboutique’’ et qui peut se connecter depuis l’hôte ‘’localhost’

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La commande suivante crée le rôle ‘’r\_Gestionnaire\_de\_la\_boutique’’

Une image contenant texte, capture d’écran, Police

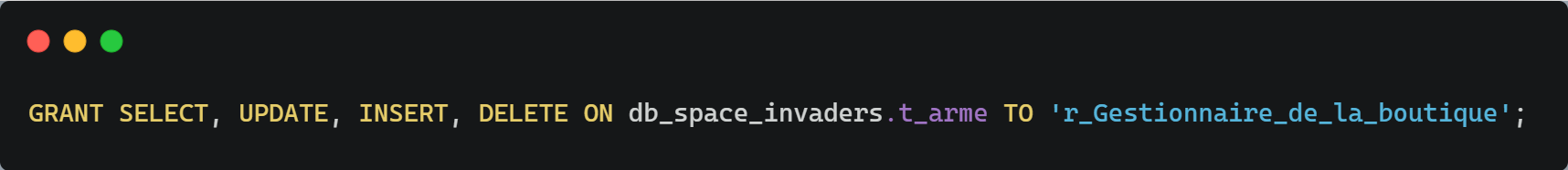
Description générée automatiquement

1. La commande suivante donne le privilège au rôle ‘’r\_Gestionnaire\_de\_la\_boutique’’ sur la table ‘’t\_Joueur’’ de la base de données ‘’db\_space\_invaders’’

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La commande suivante donne les privilèges ‘’SELECT, UPDATE, INSERT, DELETE’’ sur la table ‘’t\_arme’’ sur la base de données ‘’db\_space\_invaders’’ au rôle ‘’r\_Gestionnaire\_de\_la\_boutique’’



1. Le privilège ‘’INSERT’’ permet de rajouter des données dans une table.
2. La commande suivante donne le privilège ‘’SELECT’’ au rôle ‘’r\_Gestionnaire\_de\_la\_boutique’’ sur la table ‘’t\_commande’’ dans la base de données ‘’db\_space\_invaders’’ :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. La commande suivante donne le rôle ‘’r\_Gestionnaire\_de\_la\_boutique’’ à l’utilisateur ‘’Gestionnaire\_de\_la\_boutique’’ :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## Requêtes de sélection

*Requête n01*: La requête suivante sélectionne toutes les données de la table ‘’t\_joueur’’ et ensuite trie les résultats par ordre décroissant en fonction de la colonne ‘’jouNombrePoints’’ et limites les résultats à 5, ce qui affiche les 5 meilleures joueurs en fonction de leurs points :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’SELECT’’ permet d’indiquer à MySQL les tables que nous voulons sélectionner, dans ce cas nous sélectionnons toutes les tables car ‘’\*’’ signifie toutes les colonnes d’une table
2. ‘’FROM t\_joueur’’ permet de spécifier le ou les tables à partir desquelles nous avons sélectionner les colonnes précédentes
3. ‘’ORDER BY jouNombrePoints’’ permet d’afficher les résultats de la requête en fonction de la colonne ‘’jouNombrePoints’’
4. ‘’DESC’’ permet d’afficher les résultats dans l’ordre décroissant
5. ‘’LIMIT 5’’ permet de limiter les résultats afficher, dans ce cas il limite les résultats afficher à 5.

*Requête n02*: La requête suivante affiche le prix maximal, le prix minimal et le prix moyen des armes de la tables ‘’t\_arme’’

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’MAX (armPrix) AS PrixMaximum’’ permet d’afficher la valeur maximale de la colonne ‘’armPrix’’ et de renommer le résultat par ‘’PrixMaximum’’
2. ‘’MIN (armPrix) AS PrixMinimum’’ permet d’afficher la valeur minimale de la colonne ‘’armPrix’’ et de renommer le résultat par ‘’PrixMinimum’’
3. ‘’AVG (armPrix) AS PrixMoyen’’ permet d’afficher la valeur moyenne de la colonne ‘’armPrix’’ et de renommer le résultat par ‘’PrixMoyen’’

*Requête n03*: La requête suivante sélectionne les id des joueurs, affiche le nombre de commandes effectuées en fonction de chaque utilisateur et les tries par ordre décroissant.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’COUNT (fkJoueur) AS NombreCommandes’’ permet de compter le nombre de fois que chaque identifiant que ‘’fkJoueur’’ apparaît dans la table ‘’t\_commande’’.

De plus dans cette commande, ‘’AS NombreCommandes’’ permet de renommer la colonne ou résultat apparaitra, dans cette exemple la colonne s’appellera ‘’NombreCommandes’’

1. ‘’GROUP BY fkjoueur’’ groupe les résultats en fonction de la colonne ‘’fkJoueur’’

*Requête n04*: La requête suivante sélectionne les id des joueurs affiche le nombre de commandes effectuées en fonction de chaque utilisateur, seuls les joueurs ayants exécuter plus de 2 commandes seront afficher, de plus le nombre de commandes sera afficher dans l’ordre décroissant.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’HAVING’’ est utilisé pour filtrer les groupes, dans cette exemple il est utiliser pour spécifier que seuls les ‘’fkJoueur’’ qui apparaissent plus de 2 fois seront afficher.

*Requête n05*: La requête suivante permet d’afficher l’entièreté des colonnes ‘’jouPseudo’’, ‘’armNom’’ ainsi que ‘’comNumeroCommande’’

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’JOIN t\_arsenal ON t\_joueur.idJoueur = t\_arsenal.fkJoueur’’ permet d’interconnecter deux table entre elle, dans cette exemple les deux tables connectées sont : ‘’t\_joueur’’ et ‘’t\_arsenal’’

*Requête n06*: La commande suivante permet d’afficher la dépense totale de chaque joueur, ainsi que de regrouper les résultats par joueur, les tries par ordre décroissant et limite les résultats à 10.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

*Requête n07*: la commande ci-bas affiche tous les pseudonymes des joueurs associés aux numéros de commande, même les joueurs qui n’ont pas passé de commande seront afficher en revanche s’ils n’ont pas passé de commandes, la colonne affichera null.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’LEFT JOIN t\_commande ON t\_commande.fkJoueur = t\_joueur.idJoueur’’ Comme la commande JOIN mais au lieu de lier et d’afficher toutes les informations des deux colonnes des deux tables, toutes les informations de la table qui se situe à gauche du ‘’LEFT JOIN’’ seront afficher même si il n’a pas de correspondance avec la deuxième table.

*Requête n08*: la commande suivante permet d’afficher toutes les commandes ainsi que les joueurs qui ont passé ces commandes, les commandes seront toutes afficher même si aucune id de joueur y corresponds

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. ‘’RIGHT JOIN t\_commande ON t\_commande.fkJoueur = t\_joueur.idJoueur’’ Comme la commande JOIN LEFT, mais à la place d’afficher toutes les informations de la colonne qui est spécifier à gauche du Join, ici c’est à droite du Join que toute les informations seront afficher

*Requête n09*: La requête ci-dessous, affichera

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

1. L’expression ‘’SUM (t\_detail\_commande.detQuantiteCommande)’’ permet de calculer la somme totale des colonnes qui se trouvent dans les parenthèses qui suis le ‘’SUM’’

*Requête n010*: la requête suivante affichera une liste de pseudo de joueur qui acheter au moins une arme.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## Création des index

1. Certains index existent déjà car, MySQL crée automatiquement des index pour les colonnes suivantes : clé primaire, clé étrangère et colonne uniques, pour améliorer les performances des requêtes et garantir l’incorruptibilité des données

**Avantage**

- Performances des requêtes (les index accélèrent les recherches, ainsi que les filtrages, jointure et le tri des données)

- Intégrité des données (les index permettent d’être sûr que les données stocker dans les colonnes uniques sont uniques)

- Facilité d’utilisation (si MySQL crée des index automatiquement pas besoin d’en crée)

**Inconvénients**

- Surutilisation espace de stockage (si un nombre excessif d’index automatique est créé cela peut devenir problématique)

- Maintenance complexe (gérer un nombre important d’index leur d’une maintenance peux être plus complexe)

- Index inutiles (parfois des index peuvent être crée sur des colonnes qui ne sont pas fréquemment utilisées)

1. Un index pertinent à ajouter serait sur la colonne de la colonne ‘’t\_joueur’’, car le pseudonyme des joueurs est une information importante et de ce fait il est très probable que de nombreuses requêtes comporterons l’informations des pseudonymes des joueurs (comme le prouve les commandes de sélections précédentes, la colonne ‘’jouPseudo’’ apparait très souvent).

## Backup / Restore

### Backup

Voici la commande pour réaliser un backup de la base de données db\_space\_invaders :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Au préalable de cette commande, il faut crée un fichier qui contiendra le backup (dans cet exemple : backup\_db\_space\_invaders.sql)

Également ‘’root’’ est le nom d’utilisateur de l’utilisateur qui est utiliser pour accéder à la base de données ‘’db\_space\_invaders’’, il faut donc le remplacer en conséquence.

### Restore

Voici la commande pour réaliser un restore de la base de données db\_space\_invaders :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Il est important de préciser qu’il faut que la base de données ‘’db\_space\_invaders’’ soit préalablement crée pour que le restore fonctionne.

Et tout comme la commande de Backup précédente, dans cette exemple ‘’root’’ est le nom d’utilisateur de l’utilisateur qui vas se connecter à la base de données.

# POO – Programmation orientée objet

## Introduction

## Analyse fonctionnelle

## Analyse technique

## Tests Unitaire

## Utilisation ChatGPT

## Conclusion