Rapport P\_DB\_106

Table des matières

[1. Création d’un MCD 2](#_Toc185235441)

[2. Création de la base de données et insertion des données 3](#_Toc185235442)

[3. Gestion des utilisateurs 4](#_Toc185235443)

[Requêtes de sélection 6](#_Toc185235444)

[Création des index 9](#_Toc185235445)

[Backup / Restore 10](#_Toc185235446)

Zhuravel Mykola

Lausanne – Vennes

CIN2A

06.11.2024 – 08.01.2025

# Création d’un MCD

Au début du projet il faut créer un modèle **MCD** qui va correspondre aux demandes du CDC. Dans mon MCD j’ai inséré 3 entités suivantes : JOUEUR, COMMANDE, ARME. J’ai ainsi mis en place trois associations suivantes : effectuer, posséder, concerne.

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquementLa logique : Un joueur effectue une commande, cette commande concerne une arme. Un joueur peut aussi posséder une arme.

Chaque **joueur** possède un pseudonyme et un nombre de points accumulés dans le jeu en détruisant ses vaisseaux ennemis.

Chaque **commande** est définie par une date et possède un numéro unique. Une commande ne peut être passée que par un et un seul joueur. Une commande peut inclure plusieurs armes, et pour chaque arme commandée, une quantité est précisée.

Voici le **MLD** :

Une image contenant texte, capture d’écran, diagramme, ligne

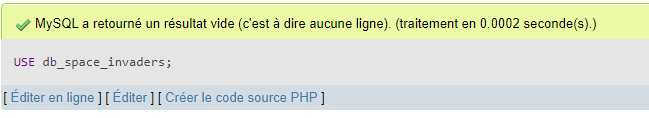
Description générée automatiquement

# Création de la base de données et insertion des données

Ensuite je vais créer une base de données ‘db\_space\_invaders’ en utilisant une requête SQL très simple.

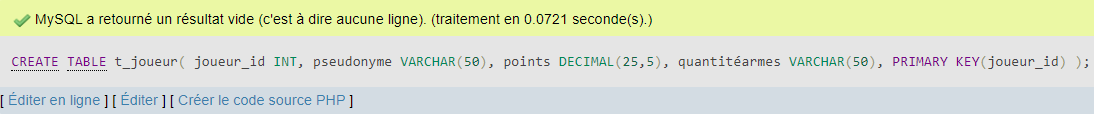
Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Puis j’utilise cette base de données.

Pour commencer à « remplir » cette base de données je dois créer des tables dans lesquels je vais insérer des données concernant les joueurs, leurs armes et leurs commandes.

Et pour finir je crée des tables de ma base de données. Je crée plusierus tables comme joueur par exemple. J’utilise la commande « CREATE TABLE » pour créer des tables.



Pour insérer des données dans les tables j’ai utilisé un générateur de données SQL appelé Mokaroo :

Site Mokaroo : [Mockaroo - Random Data Generator and API Mocking Tool | JSON / CSV / SQL / Excel](https://mockaroo.com/)

Une image contenant capture d’écran, texte

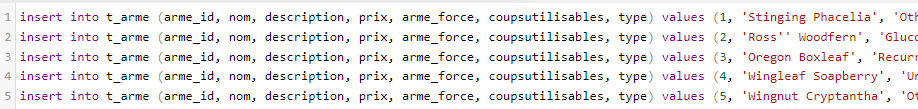
Description générée automatiquement

Le script pour insérer des données dans la table ‘t\_joueur’. En tout il faut insérer 50 unités de data dans chaque table.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Le script SQL pour insérer des données dans la table ‘t\_arme’ :



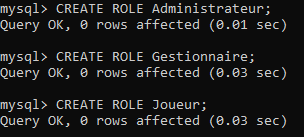
Le script SQL pour insérer des données dans la table ‘t\_commande’ :

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

# Gestion des utilisateurs

Pour commencer je dois créer des rôles. Ce sont l’administrateur du jeu, le joueur et le gestionnaire de la boutique. Voici un exemple d’une commande qui me permet de créer des rôles. (Je l’ai fait depuis CMD)



Ensuite je vais attribuer des privilèges aux rôles que je viens de créer.

Voici ceux d’administrateur :



Gestionnaire : La commande pour attribuer les privilèges demandés à un gestionnaire est la suivante :

**GRANT SELECT ON t\_joueur TO 'gestionnaire';**

**GRANT SELECT, INSERT, UPDATE, DELETE ON t\_arme TO 'gestionnaire';**

**GRANT SELECT ON t\_commande TO 'gestionnaire';**

Ensuite on peut vérifier que les droits ont bien été appropriés en utilisant la commande **SHOW GRANTS FOR ‘gestionnaire’ ;**

Voici le résultat que cette commande nous donne :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Joueur : Pour attribuer les privilèges demandés à un joueur j’ai utilisé la commande suivante :

**GRANT SELECT ON t\_arme TO 'joueur';**

**GRANT INSERT, SELECT ON t\_commande TO 'joueur';**

La vérification que les privilèges ont bien été appropriés :

Une image contenant capture d’écran, texte, Police, ligne

Description générée automatiquement

Ensuite, pour pouvoir attribuer des rôles créés je dois avoir des users spécifiques. Pour cela j’utilise des commandes suivantes :

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

L’attribution des rôles aux users :







# Requêtes de sélection

Voici une liste de requêtes SQL que vous devez réaliser et expliquer en détail.

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

**SELECT joueur\_id, pseudonyme, points**

**FROM t\_joueur**

**ORDER BY points DESC**

**LIMIT 5;**

Requête n°2 :

Trouver le prix maximum, minimum et moyen des armes.

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

**SELECT**

**MAX(prix) AS PrixMaximum,**

**MIN(prix) AS PrixMinimum,**

**AVG(prix) AS PrixMoyen**

**FROM t\_arme;**

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"

**SELECT**

**joueur\_id AS IdJoueur,**

**COUNT(\*) AS NombreCommandes**

**FROM t\_commande**

**GROUP BY joueur\_id**

**ORDER BY NombreCommandes DESC;**

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"

**SELECT**

**joueur\_id AS IdJoueur,**

**COUNT(\*) AS NombreCommandes**

**FROM t\_commande**

**GROUP BY joueur\_id**

**HAVING COUNT(\*) > 2;**

Requête n°5 :

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

**SELECT**

**t\_joueur.pseudonyme,**

**t\_arme.nom**

**FROM t\_commande**

**JOIN concerne ON t\_commande.commande\_id = concerne.commande\_id**

**JOIN t\_arme ON concerne.arme\_id = t\_arme.arme\_id**

**JOIN t\_joueur ON t\_commande.joueur\_id = t\_joueur.joueur\_id;**

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"

**SELECT**

**t\_joueur.joueur\_id AS IdJoueur,**

**SUM(t\_arme.prix \* concerne.quantité) AS TotalDepense**

**FROM t\_commande**

**JOIN concerne ON t\_commande.commande\_id = concerne.commande\_id**

**JOIN t\_arme ON concerne.arme\_id = t\_arme.arme\_id**

**JOIN t\_joueur ON t\_commande.joueur\_id = t\_joueur.joueur\_id**

**GROUP BY t\_joueur.joueur\_id**

**ORDER BY TotalDepense DESC**

**LIMIT 10;**

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`.

**SELECT**

**t\_joueur.joueur\_id,**

**t\_joueur.pseudonyme,**

**t\_commande.commande\_id,**

**t\_commande.datedecommande**

**FROM t\_joueur**

**LEFT JOIN t\_commande ON t\_joueur.joueur\_id = t\_commande.joueur\_id;**

Requête n°8 :

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

**SELECT**

**t\_commande.commande\_id,**

**IFNULL(t\_joueur.pseudonyme, NULL) AS pseudonyme**

**FROM t\_commande**

**LEFT JOIN t\_joueur ON t\_commande.joueur\_id = t\_joueur.joueur\_id;**

Requête n°9 :

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

**SELECT**

**t\_joueur.joueur\_id,**

**IFNULL(SUM(concerne.quantité), 0) AS NombreTotalArmesAchetees**

**FROM t\_joueur**

**LEFT JOIN t\_commande ON t\_joueur.joueur\_id = t\_commande.joueur\_id**

**LEFT JOIN concerne ON t\_commande.commande\_id = concerne.commande\_id**

**GROUP BY t\_joueur.joueur\_id;**

Requête n°10 :

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

**SELECT**

**t\_joueur.joueur\_id,**

**COUNT(DISTINCT concerne.arme\_id) AS TypesArmesAchetees**

**FROM t\_joueur**

**JOIN t\_commande ON t\_joueur.joueur\_id = t\_commande.joueur\_id**

**JOIN concerne ON t\_commande.commande\_id = concerne.commande\_id**

**GROUP BY t\_joueur.joueur\_id**

**HAVING COUNT(DISTINCT concerne.arme\_id) > 3;**

# Création des index

En étudiant le dump MySQL **db\_space\_invaders.sql** vous constaterez que vous ne trouvez pas le mot clé **INDEX**.

1. Pourtant certains index existent déjà. Pourquoi ?

**Certains index existent déjà parce que les clés primaires et les clés étrangères**

**créent automatiquement des index pour garantir l'unicité et améliorer la performance**

**des opérations de jointure et de recherche.**

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

**Avantages :**

**Accélèrent les requêtes SELECT et les recherches.**

**Facilitent le tri et le filtrage des données.**

**Inconvénients :**

**Occupent de l'espace supplémentaire dans la base de données.**

**Ralentissent les opérations d'INSERT, UPDATE, et DELETE car l'index doit être mis à jour.**

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

**Pseudonyme, prix et l’id de joueur**

# Backup / Restore

Nous souhaitons réaliser une sauvegarde (Backup) de la base de données db\_space\_invaders.

Ensuite, nous souhaitons nous assurer que cette sauvegarde est correcte en la rechargeant dans MySQL (opération de restauration).

Donner la commande permettant de faire :

- Un backup de la base de données db\_space\_invaders

**mysqldump -u root -p root > backup\_db\_space\_invaders.sql**

- Un restore de la base de données db\_space\_invaders

**mysql -u root-p root < backup\_db\_space\_invaders.sql**

En expliquant en détail chaque commande utilisée.