P\_DB

Table des matières

[A. Importer les données et le schéma de base de données 2](#_Toc178595241)

[B. Gestions des utilisateurs 2](#_Toc178595242)

[Admin 2](#_Toc178595243)

[Joueur 2](#_Toc178595244)

[Gestionnaire de la boutique 2](#_Toc178595245)

[C. Requêtes de sélection 3](#_Toc178595246)

[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 3](#_Toc178595247)

[2: Trouver le prix maximum, minimum et moyen des armes. Les colonnes doivent avoir pour nom « PrixMaximum », « PrixMinimum » et « PrixMoyen) 3](#_Toc178595248)

[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" 3](#_Toc178595249)

[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" 4](#_Toc178595250)

[5: Trouver le pseudo du joueur et le nom de l'arme pour chaque commande. 4](#_Toc178595251)

[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" 4](#_Toc178595252)

[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`. 5](#_Toc178595253)

[8: Récupérer toutes les commandes et afficher le pseudo du joueur s’il existe, sinon afficher `NULL` pour le pseudo. 5](#_Toc178595254)

[9: Trouver le nombre total d'armes achetées par chaque joueur (même si ce joueur n'a acheté aucune Arme). 5](#_Toc178595255)

[10: Trouver les joueurs qui ont acheté plus de 3 types d'armes différentes 6](#_Toc178595256)

[D. Création des index 6](#_Toc178595257)

[E. Backup / Restore 6](#_Toc178595258)

[Backup 6](#_Toc178595259)

[Restore 6](#_Toc178595260)

# Importer les données et le schéma de base de données

Docker exec -i db mysql -u root -proot < db\_space\_invaders.sql

# Gestions des utilisateurs

## Admin

CREATE ROLE 'Admin';

GRANT ALL PRIVILEGES

ON \*.\*

TO 'Admin'

WITH GRANT OPTION;

## Joueur

CREATE ROLE 'Joueur';

GRANT SELECT

ON t\_arme

TO 'Joueur';

GRANT INSERT, SELECT

ON t\_commande

TO 'Joueur';

## Gestionnaire de la boutique

CREATE ROLE 'Gestionnaire de la boutique';

GRANT SELECT

ON t\_joueur

TO 'Gestionnaire de la boutique';

GRANT ALL PRIVILEGES

ON t\_arme

TO 'Gestionnaire de la boutique';

GRANT SELECT

ON t\_commande

TO 'Gestionnaire de la boutique';

# Requêtes de sélection

## 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 \*

FROM t\_joueur

ORDER BY (jouNombrePoints) DESC

LIMIT 5;

## 2: Trouver le prix maximum, minimum et moyen des armes. Les colonnes doivent avoir pour nom « PrixMaximum », « PrixMinimum » et « PrixMoyen)

SELECT MAX(armPrix) AS PrixMax,

AVG(armPrix) AS PrixMoyen,

MIN(armPrix) AS PrixMin

FROM t\_arme ;

## 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 fkJoueur AS IdJoueur,

COUNT(\*) AS NombreCommande

FROM t\_commande

GROUP BY(fkJoueur) ;

## 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 fkJoueur AS IdJoueur,

COUNT(\*) AS NombreCommande

FROM t\_commande

GROUP BY (fkJoueur)

HAVING COUNT(\*) > 2;

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

SELECT t\_arme.armNom,

t\_joueur.jouPseudo

FROM t\_detail\_commande, t\_commande, t\_joueur, t\_arme

WHERE t\_detail\_commande.fkArme = t\_arme.idArme

AND t\_detail\_commande.fkCommande = t\_commande.idCommande

AND t\_commande.fkJoueur = t\_joueur.idJoueur ;

## 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.jouPseudo, SUM(t\_arme.armPrix) AS TotalDepense FROM t\_detail\_commande, t\_commande, t\_joueur, t\_arme

WHERE t\_detail\_commande.fkArme = t\_arme.idArme AND t\_detail\_commande.fkCommande = t\_commande.idCommande AND t\_commande.fkJoueur = t\_joueur.idJoueur

GROUP BY t\_joueur.jouPseudo

ORDER BY SUM(t\_arme.armPrix) DESC

LIMIT 10 ;

## 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 DISTINCT jouPseudo, IF(idCommande IS NULL, "NULL", idCommande)

FROM t\_joueur

JOIN t\_commande

ON t\_joueur.idJoueur = t\_commande.fkJoueur ;

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

SELECT DISTINCT idCommande, IF(t\_joueur.jouPseudo IS NULL, "NULL", t\_joueur.jouPseudo)

FROM t\_commande

JOIN t\_joueur

ON t\_joueur.idJoueur = t\_commande.fkJoueur ;

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

SELECT t\_joueur.jouPseudo,

SUM(IF(t\_detail\_commande.detQuantiteCommande IS NULL, "NULL", t\_detail\_commande.detQuantiteCommande)) AS nbArmes

FROM t\_joueur

JOIN t\_commande

ON t\_joueur.idJoueur = t\_commande.fkJoueur

JOIN t\_detail\_commande

ON t\_commande.idCommande = t\_detail\_commande.fkCommande

GROUP BY t\_joueur.jouPseudo ;

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

SELECT t\_joueur.jouPseudo, COUNT(t\_detail\_commande.fkArme)

FROM t\_joueur

JOIN t\_commande

ON t\_joueur.idJoueur = t\_commande.fkJoueur

JOIN t\_detail\_commande

ON t\_commande.idCommande = t\_detail\_commande.fkCommande

GROUP BY t\_joueur.jouPseudo

HAVING COUNT(DISTINCT t\_detail\_commande.fkArme) > 3;

# Création des index

1. Pourtant certains index existent déjà. Pourquoi ?

Parce que c'est des PRIMARY KEY ou des UNIQUE

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

Pour identifier un élément unique de la table

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

Imaginons une course avec 100 participants, chaque coureur aura un numéro unique sur son maillot pour pouvoir l'identifier. Ce numéro sera l'index.

# Backup / Restore

## Backup

docker exec -i db mysqldump -u root -proot --databases db\_space\_invaders > db\_space\_invaders.sql

## Restore

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