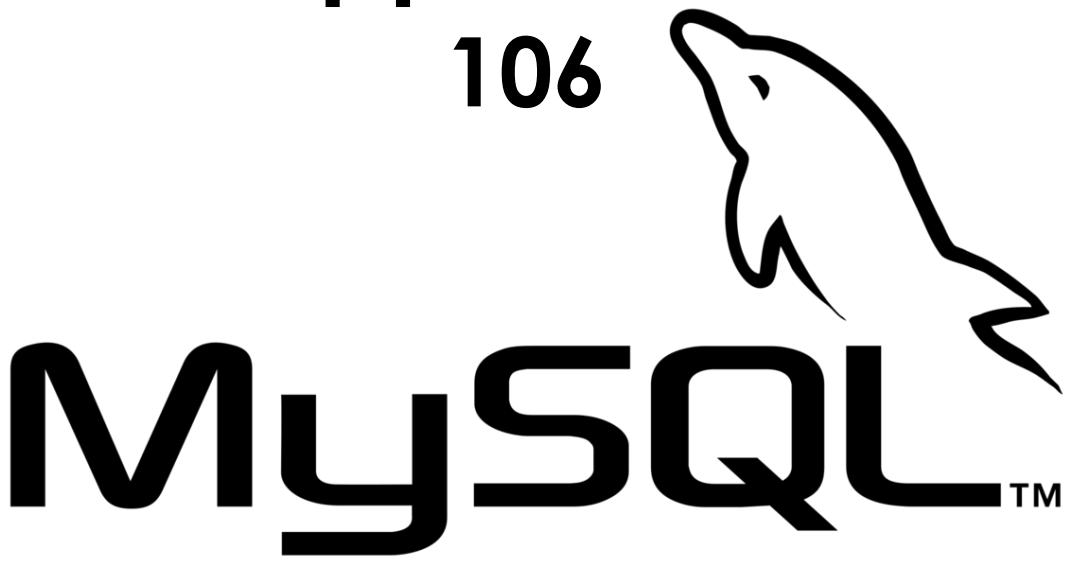
Rapport P-DB
106



Albion Bllaca
ETML - CIN2A
DGEP-ETAT DE VAUD

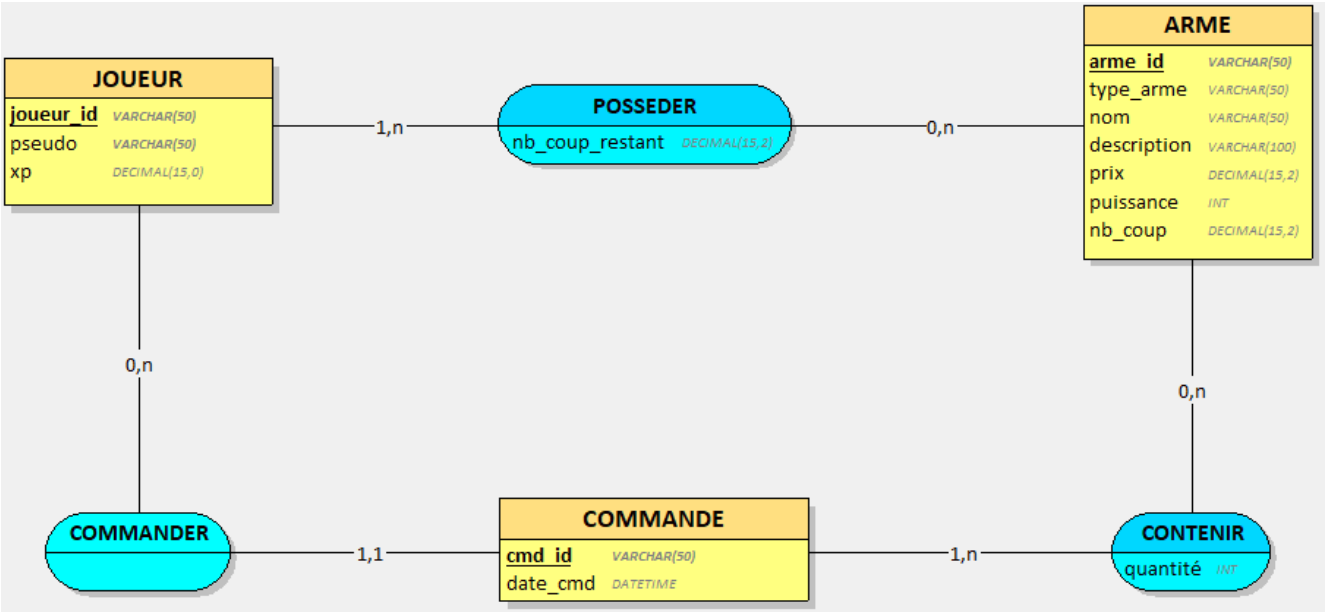
Table des matières

1 Introduction

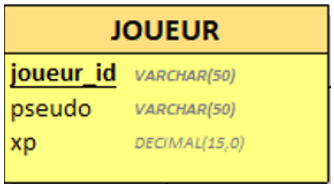
1.1 Description

Le but du projet est de créer une base de données tout en suivant le cahier des charges avec ces fonctionnalités et contraintes. Le projet se base sur un jeux fictif appelé SpaceInvaders.

2 Modèle conceptuel de données

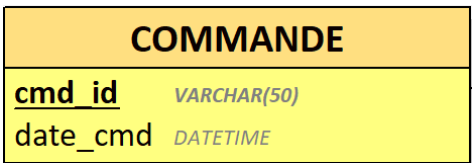


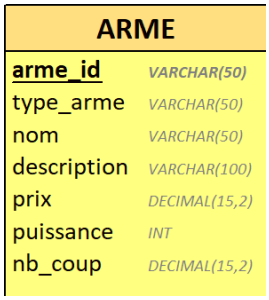
2.1 Explications des tables



La table joueur permet de stocker l'ID du joueur, son pseudonyme et ses points d'expériences (XP)

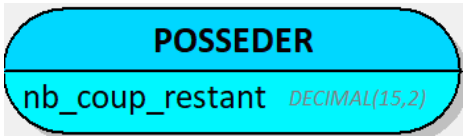
La table commande permet de mettre une identification a une commande et de stocker une date grâce à ces liaisons entre la table joueur et arme.





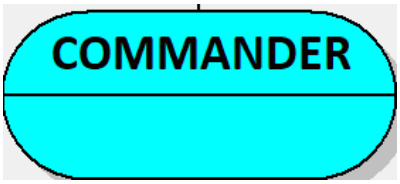
Cette table permet de stocker les armes et leur attribut comme le type, le nom, une description, un prix, une puissance de feu et le nombre d'utilisation restante dans l'arme, en partant du principe que l'arme ne peut plus être utilisée après le nombre de coup tiré

2.2 Description des tables intermédiaire



La table posséder permet de faire le lien entre la table arme et la table joueur. Elle permet aussi de déterminer le nombre de coup restant au joueur pour ses armes

La table commander permet de faire le lien entre une commande contenant id, arme et quantité et un joueur en incluant l'id du joueur dans la table commande (en tant que clé étrangère).



La table contenir permet de stocker le détail de chaque commande. Elle fait le lien entre la table Commandes et la table Armes, associant ainsi une arme commandée à un identifiant de commande, une date de commande, le joueur qui a passé la commande et la quantité commandée.

3 Exécution des script SQL

3.1 Création de la base de données

Le script SQL utilisé pour créer la base de données se nomme SpaceInvaders.sql.

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0665 seconde(s).)

```
CREATE TABLE JOUEUR( joueur_id VARCHAR(50), pseudo VARCHAR(50) NOT NULL, xp DECIMAL(15,0), PRIMARY KEY(joueur_id) );
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0610 seconde(s).)

```
CREATE TABLE ARME( arme_id VARCHAR(50), type_arme VARCHAR(50), nom VARCHAR(50), description VARCHAR(100), prix DECIMAL(15,2), puissance INT, nb_coup DECIMAL(15,2), PRIMARY KEY(arme_id) );
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0638 seconde(s).)

```
CREATE TABLE COMMANDE( cmd_id VARCHAR(50), date_cmd DATETIME, joueur_id VARCHAR(50) NOT NULL, PRIMARY KEY(cmd_id), FOREIGN KEY(joueur_id) REFERENCES JOUEUR(joueur_id) );
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0746 seconde(s).)

```
CREATE TABLE POSSEDER( joueur_id VARCHAR(50), arme_id VARCHAR(50), nb_coup_restant DECIMAL(15,2), PRIMARY KEY(joueur_id, arme_id), FOREIGN KEY(joueur_id) REFERENCES JOUEUR(joueur_id), FOREIGN KEY(arme_id) REFERENCES ARME(arme_id) );
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0717 seconde(s).)

```
CREATE TABLE CONTENIR( arme_id VARCHAR(50), cmd_id VARCHAR(50), quantité INT, PRIMARY KEY(arme_id, cmd_id), FOREIGN KEY(arme_id) REFERENCES ARME(arme_id), FOREIGN KEY(cmd_id) REFERENCES COMMANDE(cmd_id) );
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Le script s'exécute correctement sans erreurs.

3.2 Insertion des données

Le script SQL utilisé pour créer la base de données se nomme data_SpaceInvaders.sql.

100 lignes insérées. (traitement en 0.0119 seconde(s).)

```
-- Data pour JOUEUR insert into JOUEUR (joueur_id, pseudo, xp) values (1, 'aborrington0', 4490), (2, 'kchristophers1', 6972), (3, 'kcareswell2', 4004), (4, 'btrumble3', 3414), (5, 'kboulds4', 4094), (6, 'dlaste5', 5643), (7, 'lstayt6', 5933), (8, 'egregs7', 2340), (9, 'gtoor8', 4194), (10, 'lyeardley9', 1365), (11, 'wbeebya', 8448), (12, 'kcarslawb', 5718), (13, 'eheinrichc', 3843), (14, 'baseld', 5975), (15, 'btiffneye', 4117), (16, 'jhourihanf', 1866), (17, 'mburfootg', 7205), (18, 'sscopynh', 7745), (19, 'jailmeri', 2970), (20, 'lmayell3', 6889), (21, 'lassadk', 2578), (22, 'adeinhardtl', 8427), (23, 'etraytonm', 4947), (24, 'escrivinn', 1811), (25, 'jbunkleo', 6716), (26, 'bdreganp', 7174), (27, 'epundyq', 2883), (28, 'eandrysr', 5268), (29, 'hsullers', 8631), (30, 'ymckibbint', 6623), (31, 'atinceyu', 5397), (32, 'hmccarterv', 3232), (33, 'fboldtw', 6748), (34, 'sgantzmanx', 1328), (35, 'lhaycrofty', 4579), (36, 'lfantinz', 3551), (37, 'athu[...]
```

[Éditer]

50 lignes insérées. (traitement en 0.0150 seconde(s).)

```
-- Data pour ARME INSERT INTO ARME (arme_id, type_arme, nom, description, prix, puissance, nb_coup) VALUES ('a001', 'Sword', 'Excalibur', 'Legendary sword', 1000.00, 90, 100), ('a002', 'Bow', 'Longbow', 'Long-range bow', 500.00, 50, 200), ('a003', 'Axe', 'Battle Axe', 'Powerful axe for melee', 750.00, 80, 90), ('a004', 'Spear', 'Javelin', 'Throwing spear', 300.00, 45, 150), ('a005', 'Dagger', 'Shadow Blade', 'Short-range weapon', 250.00, 40, 250), ('a006', 'Crossbow', 'Repeater', 'Rapid fire crossbow', 600.00, 70, 180), ('a007', 'Staff', 'Fire Staff', 'Magic weapon with fire spells', 900.00, 85, 100), ('a008', 'Hammer', 'War Hammer', 'Heavy damage hammer', 850.00, 95, 80), ('a009', 'Mace', 'Morning Star', 'Mace with high impact', 700.00, 75, 110), ('a010', 'Bow', 'Compound Bow', 'Advanced bow for range', 550.00, 60, 200), ('a011', 'Sword', 'Dragon Slayer', 'Sword forged to hunt dragons', 1200.00, 95, 110), ('a012', 'Bow', 'Hawk Eye', 'Precision bow for sharp shooters', 600[...]
```

[Éditer]

50 lignes insérées. (traitement en 0.0146 seconde(s).)

```
-- Data pour COMMANDE INSERT INTO COMMANDE (cmd_id, date_cmd, joueur_id) VALUES ('c001', '2023-01-01 12:34:56', 1), ('c002', '2023-02-02 13:45:57', 2), ('c003', '2023-03-03 14:56:58', 3), ('c004', '2023-04-04 15:07:59', 4), ('c005', '2023-05-05 16:18:50', 5), ('c006', '2023-06-06 17:29:51', 6), ('c007', '2023-07-07 18:40:52', 7), ('c008', '2023-08-08 19:51:53', 8), ('c009', '2023-09-09 20:02:54', 9), ('c010', '2023-10-10 21:13:55', 10), ('c011', '2023-11-11 12:00:00', 11), ('c012', '2023-11-12 13:10:01', 12), ('c013', '2023-11-13 14:20:02', 13), ('c014', '2023-11-14 15:30:03', 14), ('c015', '2023-11-15 16:40:04', 15), ('c016', '2023-11-16 17:50:05', 16), ('c017', '2023-11-17 18:01:06', 17), ('c018', '2023-11-18 19:11:07', 18), ('c019', '2023-11-19 20:21:08', 19), ('c020', '2023-11-20 21:31:09', 20), ('c021', '2023-11-21 22:41:10', 21), ('c022', '2023-11-22 23:51:11', 22), ('c023', '2023-11-23 00:01:12', 23), ('c024', '2023-11-24 01:11:13', 24), ('c025', '2023-[...]
```

[Éditer]

50 lignes insérées. (traitement en 0.0144 seconde(s).)

```
-- Data pour POSSEDER INSERT INTO POSSEDER (joueur_id, arme_id, nb_coup_restant) VALUES ( 1, 'a001', 95), ( 2, 'a002', 180), ( 3 , 'a003', 85), ( 4 , 'a004', 140), ( 5 , 'a005', 235), ( 6 , 'a006', 160), ( 7 , 'a007', 80), ( 8 , 'a008', 75), ( 9 , 'a009', 105), ( 10 , 'a010', 195), ( 11 , 'a011', 125), ( 12 , 'a012', 150), ( 13 , 'a013', 90), ( 14 , 'a014', 200), ( 15 , 'a015', 175), ( 16 , 'a016', 85), ( 17 , 'a017', 100), ( 18 , 'a018', 70), ( 19 , 'a019', 60), ( 20 , 'a020', 195), ( 21 , 'a021', 110), ( 22 , 'a022', 180), ( 23 , 'a023', 130), ( 24 , 'a024', 105), ( 25 , 'a025', 90), ( 26 , 'a026', 115), ( 27 , 'a027', 170), ( 28 , 'a028', 140), ( 29 , 'a029', 75), ( 30 , 'a030', 155), ( 31 , 'a031', 190), ( 32 , 'a032', 65), ( 33 , 'a033', 120), ( 34 , 'a034', 95), ( 35 , 'a035', 135), ( 36 , 'a036', 100), ( 37 , 'a037', 155), ( 38 , 'a038', 170), ( 39 , 'a039', 80), ( 40 , 'a040', 145), ( 41 , 'a041', 160), ( 42 , 'a042', 115), ( 43 , 'a0[...]
```

[Éditer]

50 lignes insérées. (traitement en 0.0149 seconde(s).)

```
-- Data pour COMMANDER INSERT INTO CONTENIR (arme_id, cmd_id, quantité) VALUES ('a001', 'c001', 1), ('a002', 'c002', 2), ('a003', 'c003', 3), ('a004', 'c004', 1), ('a005', 'c005', 2), ('a006', 'c006', 1), ('a007', 'c007', 4), ('a008', 'c008', 2), ('a009', 'c009', 1), ('a010', 'c010', 3), ('a011', 'c011', 1), ('a012', 'c012', 2), ('a013', 'c013', 1), ('a014', 'c014', 3), ('a015', 'c015', 2), ('a016', 'c016', 4), ('a017', 'c017', 1), ('a018', 'c018', 2), ('a019', 'c019', 1), ('a020', 'c020', 3), ('a021', 'c021', 2), ('a022', 'c022', 1), ('a023', 'c023', 4), ('a024', 'c024', 1), ('a025', 'c025', 2), ('a026', 'c026', 3), ('a027', 'c027', 1), ('a028', 'c028', 2), ('a029', 'c029', 3), ('a030', 'c030', 1), ('a031', 'c031', 2), ('a032', 'c032', 3), ('a033', 'c033', 4), ('a034', 'c034', 1), ('a035', 'c035', 2), ('a036', 'c036', 3), ('a037', 'c037', 1), ('a038', 'c038', 2), ('a039', 'c039', 4), ('a040', 'c040', 1), ('a041', 'c041', 2), ('a042', 'c042', [...]
```

[Éditer]

Les données ont été insérée sans erreurs dans la base de données.

3.3 Requête

Les requêtes utilisées sont dans le fichier nommé requetes.sql.

<u>Requête N°1</u> : Sélectionner les 5 joueurs avec le meilleur score, classés dans l'ordre décroissant

Affichage des lignes 0 - 4 (total de 5, traitement en 0.0003 seconde(s).) [xp: 8960... - 8715...]

```
SELECT joueur_id, pseudo, xp FROM JOUEUR ORDER BY xp DESC LIMIT 5;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

	joueur_id	pseudo	xp 1
Éditer Copier Supprimer	90	kwawer2h	8960
Éditer Copier Supprimer	68	dhayto1v	8819
Éditer Copier Supprimer	87	lmickleburgh2e	8800
Éditer Copier Supprimer	57	lklicher1k	8796
Éditer Copier Supprimer	47	apowdrell1a	8715

Requête N°2 : Trouver le prix maximum, minimum et moyen des armes

Affichage des lignes 0 - 0 (total de 1, traitement en 0.0003 seconde(s).)

```
SELECT MAX(prix) AS PrixMaximum, MIN(prix) AS PrixMinimum, AVG(prix) AS PrixMoyen FROM ARME;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

PrixMaximum	PrixMinimum	PrixMoyen
1200.00	250.00	682.000000

Requête N°3 : Nombre total de commandes par joueur, trié par ordre décroissant

Affichage des lignes 0 - 24 (total de 50, traitement en 0.0003 seconde(s).)

```
SELECT joueur_id AS IdJoueur, COUNT(cmd_id) AS NombreCommandes FROM COMMANDE GROUP BY joueur_id ORDER BY NombreCommandes DESC;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> | Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

IdJoueur	NombreCommandes ▼ 1
44	1
33	1
34	1
35	1
36	1
37	1

Requête N°4 : Trouver les joueurs ayant passé plus de 2 commandes

La requête SQL a été exécutée avec succès.

```
SELECT joueur_id AS IdJoueur, COUNT(cmd_id) AS NombreCommandes FROM COMMANDE GROUP BY joueur_id HAVING COUNT(cmd_id) > 2;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

IdJoueur	NombreCommandes
1	3

Requête N°5 : Trouver le pseudo du joueur et le nom de l'arme pour chaque commande

Affichage des lignes 0 - 24 (total de 52, traitement en 0.0004 seconde(s).)

```
SELECT J.pseudo, A.nom AS NomArme FROM CONTENIR C JOIN ARME A ON C.arme_id = A.arme_id JOIN COMMANDE CMD ON C.cmd_id = CMD.cmd_id JOIN JOUEUR J
ON CMD.joueur_id = J.joueur_id;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

1 > >> | Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

pseudo	NomArme
aborrington0	Excalibur
kchristophers1	Longbow
kcareswell2	Battle Axe
btrumble3	Javelin
kboulds4	Shadow Blade
dlaste5	Repeater
lstayt6	Fire Staff
egregs7	War Hammer
gtoor8	Morning Star
lyeardley9	Compound Bow
wbeebya	Dragon Slayer

Requête N°6 : Total dépensé par chaque joueur, ordonné par montant décroissant, limité aux 10 premiers joueurs

Affichage des lignes 0 - 9 (total de 10, traitement en 0.0006 seconde(s).)

```
SELECT CMD.joueur_id AS IdJoueur, SUM(A.prix * C.quantité) AS TotalDepense FROM CONTENIR C JOIN ARME A ON C.arme_id = A.arme_id JOIN COMMANDE CMD
ON C.cmd_id = CMD.cmd_id GROUP BY CMD.joueur_id ORDER BY TotalDepense DESC LIMIT 10;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

IdJoueur	TotalDepense ▼ 1
1	5760.00
7	3600.00
23	3280.00
33	3080.00
49	2960.00
39	2920.00
47	2790.00
16	2600.00
50	2360.00
3	2250.00

<u>Requête N°7 :</u> Récupérer tous les joueurs et leurs commandes, même s'ils n'ont pas passé de commande

Affichage des lignes 0 - 24 (total de 103, traitement en 0.0002 seconde(s).)

```
SELECT J.joueur_id, J.pseudo, CMD.cmd_id, CMD.date_cmd FROM JOUEUR J LEFT JOIN COMMANDE CMD ON J.joueur_id = CMD.joueur_id;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP]

1 > >> | Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

joueur_id	pseudo	cmd_id	date_cmd
1	aborrington0	c001	2023-01-01 12:34:56
1	aborrington0	c051	2023-02-04 08:39:45
1	aborrington0	c052	2023-02-04 08:39:45
10	lyeardley9	c010	2023-10-10 21:13:55
100	lspellward2r	*NULL*	*NULL*
1011	andrewTate92i	*NULL*	*NULL*

<u>Requête N°8 :</u> Récupérer toutes les commandes et afficher le pseudo du joueur ou NULL si le joueur n'existe pas

Affichage des lignes 0 - 24 (total de 52, traitement en 0.0003 seconde(s).)

```
SELECT CMD.cmd_id, CMD.date_cmd, J.pseudo FROM COMMANDE CMD LEFT JOIN JOUEUR J ON CMD.joueur_id = J.joueur_id;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Ac

1 > >> | Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

cmd_id	date_cmd	pseudo
c001	2023-01-01 12:34:56	aborrington0
c002	2023-02-02 13:45:57	kchristophers1
c003	2023-03-03 14:56:58	kcareswell2
c004	2023-04-04 15:07:59	btrumble3
c005	2023-05-05 16:18:50	kboulds4
c006	2023-06-06 17:29:51	dlaste5
c007	2023-07-07 18:40:52	lstayt6
c008	2023-08-08 19:51:53	egregs7
c009	2023-09-09 20:02:54	gtoor8
c010	2023-10-10 21:13:55	lyeardley9
c011	2023-11-11 12:00:00	wbeebya

Requête N°9 : Nombre total d'armes achetées par chaque joueur, y compris ceux qui n'ont acheté aucune arme

Affichage des lignes 0 - 24 (total de 101, traitement en 0.0006 seconde(s).)

```
SELECT J.joueur_id, J.pseudo, COALESCE(SUM(C.quantité), 0) AS TotalArmes FROM JOUEUR J LEFT JOIN COMMANDE CMD ON
J.joueur_id = CMD.joueur_id LEFT JOIN CONTENIR C ON CMD.cmd_id = C.cmd_id GROUP BY J.joueur_id, J.pseudo;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP

1 > >> | Tout afficher | Nombre de lignes : 25 Filtrer les lignes: Chercher dans cette table Trier pa

Options supplémentaires

joueur_id	pseudo	TotalArmes
1	aborrington0	9
10	lyeardley9	3
100	lspellward2r	0
1011	andrewTate92i	0
11	wbeebya	1
12	kcarslawb	2
13	eheinrichc	1
14	baseld	3
15	btiffneye	2
16	jhourihanf	4
17	mburfootg	1
18	sscopynh	2

Requête N°10 : Trouver les joueurs ayant acheté plus de 3 types d'armes différentes

MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0.0004 seconde(s).)

```
SELECT CMD.joueur_id AS IdJoueur, COUNT(DISTINCT C.arme_id) AS TypesArmes FROM CONTENIR C JOIN COMMANDE CMD ON C.cmd_id =
CMD.cmd_id GROUP BY CMD.joueur_id HAVING COUNT(DISTINCT C.arme_id) > 3;
```

Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

IdJoueur	TypesArmes

4 Création des utilisateurs

Tout d'abord nous commençons par créer des rôles. Ces rôles permettent d'avoir des droits assignés à des zones spécifiques, comme le droit à la lecture (SELECT), insertion (INSERT), suppression (DELETE), etc… Ces rôles seront par la suite attribués à des utilisateurs qui sont créer par la suite.

4.1 Création des rôles

```
-- Suppression des rôles existants (si nécessaires)
DROP ROLE IF EXISTS 'Role_Administrateur';
DROP ROLE IF EXISTS 'Role_Joueur';
DROP ROLE IF EXISTS 'Role_Gestionnaire';

-- Suppression des utilisateurs existants (si nécessaires)
DROP USER IF EXISTS 'Administrateur'@'localhost';
DROP USER IF EXISTS 'Joueur'@'localhost';
DROP USER IF EXISTS 'Gestionnaire'@'localhost';

-- Création des rôles
CREATE ROLE 'Role_Administrateur';
CREATE ROLE 'Role_Joueur';
CREATE ROLE 'Role_Gestionnaire';
```

4.2 Attribution des privilèges aux rôles

```
-- Attribution des privilèges au rôle Administrateur
GRANT SELECT, INSERT, UPDATE, DELETE ON *.* TO 'Role_Administrateur' WITH GRANT
OPTION;

-- Attribution des privilèges au rôle Joueur
GRANT SELECT ON T_CONTENIR TO 'Role_Joueur';
GRANT SELECT, INSERT ON T_COMMANDE TO 'Role_Joueur';
GRANT SELECT ON T_ARME TO 'Role_Joueur';

-- Attribution des privilèges au rôle Gestionnaire
GRANT SELECT ON T_JOUEUR TO 'Role_Gestionnaire';
GRANT SELECT, UPDATE, DELETE ON T_ARME TO 'Role_Gestionnaire';
GRANT SELECT ON T_COMMANDE TO 'Role_Gestionnaire';
GRANT SELECT, UPDATE ON T_CONTENIR TO 'Role_Gestionnaire';
```

4.3 Création des utilisateurs et attribution des rôles

```
-- Création des utilisateurs et attribution des rôles
CREATE USER 'Administrateur'@'localhost' IDENTIFIED BY 'root';
GRANT 'Role_Administrateur' TO 'Administrateur'@'localhost';
SET DEFAULT ROLE 'Role_Administrateur' TO 'Administrateur'@'localhost';

CREATE USER 'Joueur'@'localhost' IDENTIFIED BY 'root';
GRANT 'Role_Joueur' TO 'Joueur'@'localhost';
SET DEFAULT ROLE 'Role_Joueur' TO 'Joueur'@'localhost';

CREATE USER 'Gestionnaire'@'localhost' IDENTIFIED BY 'root';
GRANT 'Role_Gestionnaire' TO 'Gestionnaire'@'localhost';
SET DEFAULT ROLE 'Role_Gestionnaire' TO 'Gestionnaire'@'localhost';

-- Test des utilisateurs
SHOW GRANTS FOR 'Administrateur'@'localhost';
SHOW GRANTS FOR 'Joueur'@'localhost';
SHOW GRANTS FOR 'Gestionnaire'@'localhost';
```

5 Index

Créer un index sur les points d'expérience (xp) des joueurs dans une table JOUEUR permet d'améliorer les performances des requêtes qui filtrent ou trient les données en fonction de cette colonne.

```
CREATE INDEX point_exp ON JOUEUR (xp);
```

Certain index est déjà créé par défaut par MySQL pour optimiser les performances. Par exemple les clés primaires génèrent automatiquement des indexes pour accélérer les recherches.

Quelle sont les bénéfices et les contraintes des indexes ?

Bénéfices :

Une amélioration des performances : accélèrent les requêtes en réduisant le temp de recherche en utilisant un SELECT par exemple, lors des tris avec des ORDER BY ou des GROUP BY.

Contraintes :

La création d’indexe occupe de l’espace, c’est pour cela que nous ne pouvons pas indexer toutes les données de la base. Elle peut causer des ralentissements des insertions de données si celle-ci sont indexée car les indexe doivent être mis à jour.

6 Backup / restore

Voici la commande permettant de faire un backup (sauvegarde)

```
mysqldump -u root -proot SpaceInvader > SpaceInvader_back.sql
```

La première commande utilisée est mysqldump, car c'est une application dédiée à effectuer les sauvegardes de bases de données. Ensuite, on spécifie le nom d'utilisateur et le mot de passe avec les options -u root et -proot. Après cela, il faut indiquer la base de données à sauvegarder, dans ce cas, "SpaceInvaders". Enfin, le symbole > est utilisé pour rediriger la sortie et spécifier le nom du fichier dans lequel on souhaite enregistrer la base de données.

Le fichier de backup est disponible sous le nom de SpaceInvader_back.sql.

Pour l'export c'est presque pareil, la différence c'est que cette fois-ci on utilise mysql comme application et on inverse le jalon. Bien sur comme cette fois-ci nous restaurons une base il ne faut pas spécifier de base de données.

Voici la commande permettant de faire un restore (restauration)

```
mysql -u root -proot < SpaceInvader_back.sql
```