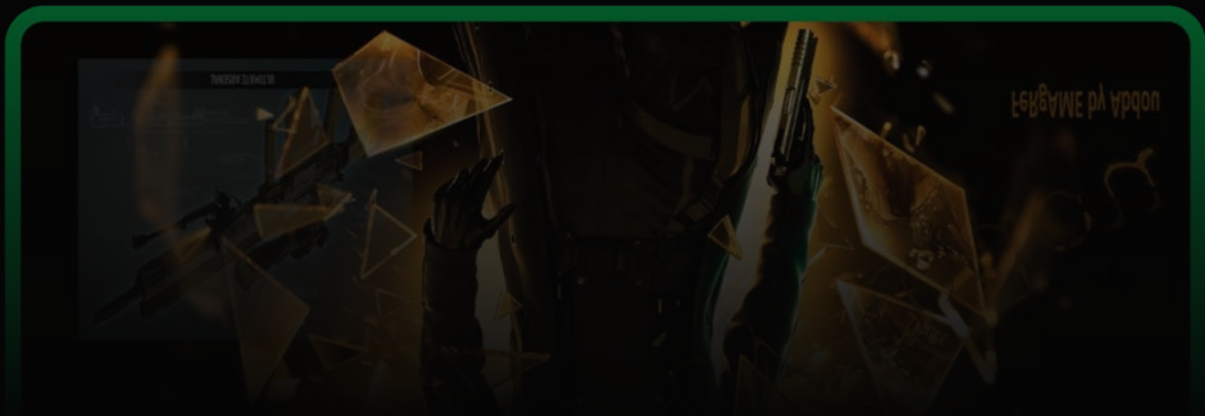


---

*FERGAME*

*by Abdou*

---



# Fer\_REQUETES



## A. Mis à jour des achats et paiement

1. Pour chaque joueur retirer de son fond personnel le prix du personnage qu'il incarne.

```
mysql> UPDATE joueur T1 INNER JOIN personnage T2 on  
T1.pseudo_joueur=T2.pseudo_joueur  
-> SET T1.fonds=T1.fonds-(CASE T2.prix_perso  
-> WHEN '10000' THEN 10000  
-> WHEN '100000' THEN 100000  
-> WHEN '10000000' THEN 10000000  
-> WHEN '50000000' THEN 50000000  
-> WHEN '75000000' THEN 75000000  
-> ELSE 0  
-> END)  
-> WHERE T2.prix_perso IN('10000','100000','10000000','50000000','75000000');
```

2. Faire en sorte que les joueurs Edgar, Alvin et Junior qui n'ont pas encore réglé la somme qu'ils doivent au vendeur d'armes, puisse s'acquitter de leur dette.

```
mysql> UPDATE joueur T1 INNER JOIN personnage T2 USING (pseudo_joueur)
INNER JOIN possede_arme USING (id_perso) INNER JOIN arme T3 USING
(code_arme)
-> SET T1.fonds=T1.fonds-(CASE T2.pseudo_joueur
-> WHEN 'alvin' THEN (SELECT sum(prix_A) FROM personnage INNER JOIN
possede_arme USING(id_perso) INNER JOIN arme USING(code_arme) WHERE
personnage.pseudo_joueur='alvin')
-> WHEN 'edgar' THEN (SELECT sum(prix_A) FROM personnage INNER JOIN
possede_arme USING(id_perso) INNER JOIN arme USING(code_arme) WHERE
personnage.pseudo_joueur='edgar')
-> WHEN 'junior' THEN (SELECT sum(prix_A) FROM personnage INNER JOIN
possede_arme USING(id_perso) INNER JOIN arme USING(code_arme) WHERE
```

3. Annoncer le vainqueur du dernier tournoi achevé et le payer.

```
mysql> SELECT pseudo_joueur vainqueur, points_t, prix_vainqueur
FROM tournoi_cree
INNER JOIN tournoi t
USING(id_tournoi)
WHERE t.etat_t='termine'
AND points_t=(SELECT MAX(points_t)
FROM tournoi_cree
INNER JOIN tournoi
USING (id_tournoi)
WHERE etat_t='termine')
ORDER BY points_t DESC;
```

4. Faire en sorte que les personnages de l'élite **La SECTE** s'acquittent de leur dette auprès du vendeur d'équipement.

```
mysql> UPDATE joueur J
-> SET J.fonds=J.fonds-(SELECT SUM(prix) FROM equipement INNER JOIN
dispose_equipement USING (id_equipement) INNER JOIN personnage P
USING(id_perso) WHERE P.pseudo_joueur='fode' )
-> WHERE pseudo_joueur='fode';
```

**B.** Mis à jour sur les points gagnés en tournoi (terminé et en cours)

1. Mettre à jour les points des joueurs participants au tournoi terminé et afficher le classement général

```
mysql> UPDATE joueur T1 INNER JOIN tournoi_cree T2 USING(pseudo_joueur)
INNER JOIN tournoi T3 USING(id_tournoi)
-> SET T1.points_j=T1.points_j+(CASE T2.pseudo_joueur
-> WHEN 'allan' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='termine' AND
T2.pseudo_joueur='allan' )
-> WHEN 'fode' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='termine' AND
T2.pseudo_joueur='fode' )
-> WHEN 'abdou' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='termine' AND
T2.pseudo_joueur='abdou' )
-> ELSE 0
-> END)
-> WHERE T2.pseudo_joueur IN ('abdou','fode','allan');
```

2. Mettre à jour les points des joueurs participants aux tournois en cours et afficher le classement en live.

```
mysql> UPDATE joueur T1 INNER JOIN tournoi_cree T2 USING(pseudo_joueur)
INNER JOIN tournoi T3 USING(id_tournoi)
-> SET T1.points_j=T1.points_j+(CASE T2.pseudo_joueur
-> WHEN 'abdou' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='abdou' )
-> WHEN 'cristiano' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER
JOIN tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='cristiano' )
-> WHEN 'messi' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='messi' )
-> WHEN 'edgar' THEN (SELECT SUM(T2.points_t) FROM tournoi_cree T2 INNER
JOIN tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='edgar' )
-> WHEN 'adja' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='adja' )
-> WHEN 'junior' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='junior' )
-> WHEN 'baidy' THEN (SELECT T2.points_t FROM tournoi_cree T2 INNER JOIN
tournoi T3 USING(id_tournoi) WHERE T3.etat_t='en cours' AND
T2.pseudo_joueur='baidy' )
-> ELSE 0
-> END)
-> WHERE T2.pseudo_joueur IN ('abdou','cristiano','baidy','messi','edgar','junior','adja');
```

### C. Sélection sur FERGAME

NB : le classement si rien n'est précisé se fait sur les points et non les xps ou le fond

1. Afficher le classement local pour le Sénégal après update

```
mysql> SELECT pseudo_joueur pseudo,pays, points_j points
FROM joueur J
WHERE J.pays='Senegal'
ORDER BY points_j DESC;
```

2. Afficher le classement général après update

```
mysql> SELECT pseudo_joueur pseudo,pays, points_j points
FROM joueur J
ORDER BY points_j DESC;
```

3. Afficher les joueurs ayant déjà gagné un tournoi

```
mysql> SELECT J.pseudo_joueur pseudo,J.pays pays, J.points_j points
FROM joueur J
INNER JOIN tournoi_cree T_C
USING (pseudo_joueur)
INNER JOIN tournoi
USING (id_tournoi) WHERE T_C.points_t=(SELECT MAX(points_t)
FROM tournoi_cree INNER JOIN tournoi USING (id_tournoi) WHERE etat_t='termine')
;
```

4. Afficher les joueurs qui possèdent un missile

```
mysql> SELECT J.pseudo_joueur, A.type, A.nom_A  
FROM joueur J INNER JOIN personnage  
USING (pseudo_joueur) INNER JOIN possede_arme  
USING (id_perso) INNER JOIN arme A  
USING (code_arme) WHERE A.type='missile';
```

5. Afficher tous les personnages de niveau supérieur à 3 qui sont déjà incarnés par des joueurs

```
mysql> SELECT nom_perso personnage_incarnés  
FROM joueur J  
CROSS JOIN personnage P  
ON J.pseudo_joueur=P.pseudo_joueur;
```

6. Afficher l'ensemble des personnage non encore incarnés

```
mysql> SELECT nom_perso Non_incarnés  
FROM personnage P  
WHERE pseudo_joueur IS NULL;
```

7. Faire un classement des armes en fonction de leur portée ou dégâts (xps\_coup)



```
mysql> SELECT A.nom_A, A.type, A.xps_coup  
FROM arme A  
ORDER BY xps_coup DESC;
```

8. Afficher les personnages équipés d'un bunker

```
mysql> SELECT P.nom_perso, E.outils  
FROM personnage P  
INNER JOIN dispose_equipement  
USING (id_perso)  
INNER JOIN equipement E  
USING (id_equipement)  
WHERE E.outils='Bunker';
```

9. Afficher le joueur ayant dépensé le plus en armement puis en équipement.

```
mysql> SELECT NT.joueur ,NT.somme  
FROM (SELECT J.pseudo_joueur joueur ,SUM(A.prix_A) somme  
      FROM joueur J  
      INNER JOIN personnage  
      USING(pseudo_joueur)  
      INNER JOIN possede_arme  
      USING (id_perso)  
      INNER JOIN arme A  
      USING (code_arme)  
      GROUP BY J.pseudo_joueur) NT  
WHERE NT.somme=(SELECT MAX(K.somme)  
      FROM (SELECT J.pseudo_joueur joueur ,SUM(A.prix_A) somme  
            FROM joueur J  
            INNER JOIN personnage  
            USING(pseudo_joueur)  
            INNER JOIN possede_arme  
            USING (id_perso)  
            INNER JOIN arme A  
            USING (code_arme)  
            GROUP BY J.pseudo_joueur) K) ;
```

## ➔ En équipement

```
mysql> SELECT NT.joueur ,NT.somme somme_depensée
FROM (SELECT J.pseudo_joueur joueur ,SUM(E.prix) somme
      FROM joueur J
      INNER JOIN personnage
      USING(pseudo_joueur)
      INNER JOIN dispose_equipement
      USING (id_perso)
      INNER JOIN equipement E
      USING (id_equipement)
      GROUP BY J.pseudo_joueur) NT
WHERE NT.somme=(SELECT MAX(K.somme)
      FROM (SELECT J.pseudo_joueur joueur ,SUM(E.prix) somme
            FROM joueur J
            INNER JOIN personnage
            USING(pseudo_joueur)
            INNER JOIN dispose_equipement
            USING (id_perso)
            INNER JOIN equipement E USING (id_equipement)
            GROUP BY J.pseudo_joueur) K);
```

## 10. Classer les joueurs en fonction de leurs achats en armes + équipement

```
mysql> SELECT T1.joueur,T1.sommeA depenses_equipement, T2.somme
depenses_armes, SUM(T1.sommeA+T2.somme) depenses_TOTAL
FROM (SELECT J.pseudo_joueur joueur ,SUM(E.prix) sommeA
      FROM joueur J
      INNER JOIN personnage
      USING(pseudo_joueur)
      INNER JOIN dispose_equipement
      USING (id_perso)
      INNER JOIN equipement E
      USING (id_equipement)
      GROUP BY J.pseudo_joueur) T1
CROSS JOIN (SELECT J.pseudo_joueur joueur ,SUM(A.prix_A) somme
            FROM joueur J
            INNER JOIN personnage
            USING(pseudo_joueur)
            INNER JOIN possede_arme
            USING (id_perso)
            INNER JOIN arme A
            USING (code_arme)
            GROUP BY J.pseudo_joueur) T2
ON T1.joueur=T2.joueur
GROUP BY T1.joueur
ORDER BY T1.sommeA+T2.somme DESC;
```

## 11. Afficher les joueurs qui ont un personnage appartenant à une élite.

```
mysql> SELECT pseudo_joueur
-> FROM joueur
-> INNER JOIN personnage
-> USING (pseudo_joueur)
-> INNER JOIN appartient_elite
-> USING(id_perso);
```

12. Afficher les personnages de niveau supérieur ou égal à 4 appartenant à une élite.

```
mysql> SELECT nom_perso personnage, niveau_p niveau  
-> FROM personnage P  
-> INNER JOIN appartient_elite  
-> USING(id_perso)  
-> WHERE P.niveau_p>='4';
```

13. Faites la somme totale des dépenses en armement.

```
mysql> SELECT SUM(prix_A) somme_totale_depenses  
-> FROM arme  
-> INNER JOIN possede_arme  
-> USING(code_arme);
```

14. Renvoyer le prix moyen d'une arme

```
mysql> SELECT AVG(prix_A) prix_moyen_arme  
-> FROM arme;
```

15. Classer les armes par prix et par type dans un ordre décroissant.

```
mysql> SELECT A.type, A.nom_A nom_arme, A.prix_A prix_arme  
-> FROM arme A  
-> GROUP BY A.type , A.prix_A  
-> ORDER BY A.prix_A DESC;
```

16. Afficher dans l'ordre décroissant le classement des « Légende » de Fergame.

```
mysql> SELECT pseudo_joueur nom_legende, points_j points  
-> FROM joueur  
-> WHERE niveau_j='Legende'  
-> ORDER BY points_j DESC;
```

17. Afficher les noms des personnages avec le nombre d'armes et d'équipement qu'ils possèdent.

```
mysql> SELECT nom_perso nom, COUNT(id_perso) nombre_arme  
-> FROM personnage  
-> INNER JOIN possede_arme  
-> USING (id_perso)  
-> GROUP BY id_perso;
```

18. Afficher le joueur qui à le plus de points (après m-à-j).

```
mysql> SELECT pseudo_joueur , points_j  
-> FROM joueur  
-> WHERE points_j=(SELECT MAX(K.points_j)  
-> FROM joueur K);
```

19. Afficher les joueurs qui se sont inscrits à Fergame entre les dates « 2019-01-01 00:00 » et « 2019-10-30 00:00 »

```
mysql> SELECT pseudo_joueur joueur, date_inscription  
-> FROM joueur  
-> WHERE date_inscription > '2019-01-01 00:00'  
-> AND date_inscription < '2019-10-30 00:00';
```

20. Afficher les joueurs qui ont des personnages possédant un pistolet.

```
mysql> SELECT J.pseudo_joueur joueur, A.type ,A.nom_A nom_arme  
-> FROM joueur J  
-> INNER JOIN personnage  
-> USING (pseudo_joueur)  
-> INNER JOIN possede_arme  
-> USING(id_perso)  
-> INNER JOIN arme A  
-> USING(code_arme)  
-> WHERE A.type='pistolet';
```

21. Afficher le joueur mort dans une partie.

NB : est considéré mort dans une partie de Fergame le joueur dont les points vie restants sont nuls.

```
mysql> SELECT P.pseudo_joueur joueur , P.point_vie_restant  
-> FROM personnage P  
-> WHERE P.point_vie_restant=0;
```

22. Afficher les joueurs ayant participé à plus d'un tournoi.

```
mysql> SELECT NT.pseudo_joueur ,NT.nombre_de_tournoi  
->FROM (SELECT T.pseudo_joueur, COUNT(T.pseudo_joueur) nombre_de_tournoi  
->FROM tournoi_cree T GROUP BY T.pseudo_joueur) NT  
->WHERE NT.nombre_de_tournoi >1;
```

23. Afficher l'équipement que tous les joueurs ont en commun.

```
mysql> SELECT NT.outils, NT.nbre  
-> FROM (SELECT outils, COUNT(outils) nbre FROM equipement INNER JOIN  
dispose_equipement  
-> USING(id_equipement) GROUP BY outils) NT  
-> WHERE NT.nbre=11;
```