**FERGAME**

**by Abdou**



Fer\_REQUETES 😎😎

1. Mis à jour des achats et paiement
2. 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');

1. Faire en sorte que les joueurs Edgar, Alvin et Junior qui n’ont pas encore régler 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 personnage.pseudo\_joueur='junior')

-> ELSE 0

-> END)

-> WHERE T2.pseudo\_joueur IN('alvin','edgar','junior');

1. 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;

1. 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';

1. Mis à jour sur les points gagnés en tournoi (terminé et en cours)
2. 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');

1. 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');

1. 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;

1. Afficher le classement général après update

mysql> SELECT pseudo\_joueur pseudo,pays, points\_j points

FROM joueur J

ORDER BY points\_j DESC;

1. 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') ;

1. 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';

1. 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;

1. Afficher l’ensemble des personnage non encore incarnés

mysql> SELECT nom\_perso Non\_incarnés

FROM personnage P

WHERE pseudo\_joueur IS NULL;

1. Faire un classement des armes en fonction de leur portée ou dégâts (xps\_coup)
2. 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';

mysql> SELECT A.nom\_A, A.type, A.xps\_coup

FROM arme A

ORDER BY xps\_coup DESC;

1. 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);

1. 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;

1. 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);

1. 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';

1. 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);

1. Renvoyer le prix moyen d’une arme

mysql> SELECT AVG(prix\_A) prix\_moyen\_arme

-> FROM arme;

1. 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;

1. 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;

1. 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;

1. 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);

1. Afficher les joueurs qui se sont inscrits à Fergame entre les dates « 2019-01-01 00:00 » et « 2019-10-30 00:00 »
2. 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';

1. 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;

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';

1. 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;

1. Afficher l’equipement 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;