

TD6 - Le retour du Select

Inès de Courchelle

Durée : 4h30

Consignes :

Durant ce TD l'utilisation d'un papier et d'un crayon est fortement conseillé !

Objectifs :

- Réaliser des tripples requêtes
- Maîtriser les jointures (LEFT,RIGHT,INNER)
- Réfléchir encore un peu plus qu'avant !

Attention :

- L'ensemble des exercices ci-dessous ne seront pas tous corrigés en cours !
- Les éléments de correction seront donnés en TD, EN AUCUN CAS, des corrections toutes faites vous seront données ou distribuées. Vous devez prendre des notes !

Exo 1 - Zelda

Nous RE-considérons le MLD suivant :

```
Joueur(idJoueur, pseudo, niveauEndurance, nbCoeurs)

Plat(idPlat, nom, famille,puissance)

Manger(#idJoueur, #idPlat,horaire)

Ingredient(idIngredient,nom);

Composer(#idIngredient,#idPlat,quantité);

Monde(idMonde, nom)

Sanctuaire(idSanctuaire, nom, niveau, #idMonde)

Visiter(#idJoueur,#idSanctuaire,horaire, vainqueur)

CreatureDivine(idCreatureDivine,nom,#idMonde)

Combattre(#idJoueur,#idCreatureDivine,horairevainqueur)
```

Attention : Si vous avez déjà utilisé la base de données Zelda dans le TP précédent, il n'est pas nécessaire de ré-installer la BDD. Vous pouvez directement lancer MySQL et lancer la commande `use Zelda`. Cependant, il vous faudra créer un nouveau script pour stocker les nouvelles instructions.

1. Télécharger le script [zeldaLDD.sql](#)
2. Lancer MySQL
3. Exécuter le script `zeldaLDD.sql`
4. Créer le nouveau script `zeldaLMDTD6.sql`
5. Ajouter les instructions au script `zeldaLMDTD6.sql` permettant d'afficher :
 - a. Quels sont les joueurs qui n'ont jamais mangé de plat ?

```
SELECT DI STINCT(i dJoueur)
FROM   Joueur
NATURAL LEFT JOI N Manger
WHERE  Manger.i dJoueur i s nul l ;
```

b. Quel est le joueur qui a consommé le plus de plat ?

```
SELECT idJoueur, count(*) as nbPlats
FROM Manger GROUP BY idJoueur
HAVING nbPlats = (SELECT max(total)
FROM (SELECT count(*) as total
FROM Manger GROUP BY idJoueur) as TAB);
```

c. Quel plat contient le plus d'ingrédients différents ?

```
SELECT idPlat, count(*) as nbIngredient
FROM Composer GROUP BY idPlat
HAVING nbIngredient = (SELECT max(total)
FROM (SELECT count(*) as total
FROM Composer GROUP BY idPlat) as TAB);
```

Exo 2 - Harry Potter

Nous RE-considérons le MLD suivant :

Maison(idMaison, nom, couleur)

Eleve(idEleve, nom, prenom, #idMaison)

Professeur(idProfesseur, nom, prenom);

Matiere(idMatiere, intitule, #idProfesseur);

EleveSuitMatiere(#idEleve, #idMatiere)

Points(#idProfesseur, #idEleve, horaire, motif, nbPoints);

Attention : Si vous avez déjà utilisé la base de données HarryPotter dans le TP précédent, il n'est pas nécessaire de ré-installer la BDD. Vous pouvez directement lancer MySQL et lancer la commande `use HarryPotter`. Cependant, il vous faudra créer un nouveau script pour stocker les nouvelles instructions.

1. Télécharger le script [harryPotterLDD.sql](#)

2. Lancer MySQL

3. Exécuter le script `harryPotterLDD.sql`

4. Créer le script `harryPotterLMDTD6.sql`

5. Ajouter les instructions au script `harryPotterLMDTD6.sql` permettant d'afficher :

a. On veut obtenir le nom, le prénom et le nombre de points total obtenu par chaque étudiant, y compris ceux qui n'ont gagné aucun points. Trier le résultat.

```
SELECT El eve. nom, El eve. prenom, sum(Poi nts. nbPoi nts) as
total Poi nts
FROM El eve, Poi nts
WHERE El eve. i dEl eve = Poi nts. i dEl eve
GROUP BY El eve. i dEl eve
ORDER BY total Poi nts DESC;
```

b. Quels sont les élèves dont le nombre de points remporté est supérieur au nombre de points moyen remporté ?
Vous prendrez en compte dans la moyenne les points positifs et négatifs.

```
SELECT El eve. nom, El eve. prenom, sum(Poi nts. nbPoi nts) as total Poi nts
FROM El eve, Poi nts
WHERE El eve. i dEl eve = Poi nts. i dEl eve
GROUP BY El eve. i dEl eve HAVING total Poi nts >= (SELECT avg(total) FROM (SELECT sum(nbPoi nts) as total
FROM Poi nts GROUP BY Poi nts. i dEl eve as TAB) ORDER BY total Poi nts DESC;
```

- c. Donnez la liste des élèves de "Severus Rogue". Dans le résultat, vous afficherez le nom et le prénom des élèves.

```
SELECT nom, prenomFROM El eveWHERE i dEl eve IN (SELECT i dEl eve FROM El eveSui tMati ereWHERE i dMati ere = (SELECT i dMati ere FROM Mati ereWHERE i dProfesseur = (SELECT i dProfesseur FROM ProfesseurWHERE nom = "Rogue" AND prenom = "Severus")));
```

- d. Quel professeur a le plus utilisé le système de point ? Dans le résultat, vous afficherez le nom et le prénom du professeur, ainsi que le nombre de fois où il a utilisé son pouvoir.

```
SELECT nom, prenomFROM ProfesseurWHERE i dProfesseur = (SELECT i dProfesseurFROM Poi ntsGROUP BY i dProfesseur HAVING count(*) = (SELECT max(total)FROM (SELECT count(*) as totalFROM Poi ntsGROUP BY i dProfesseurasTAB)));
```

Exo 3 : Mario Kart

Nous considérons le MLD suivant :

Personnage(id, nom, dateNaissance)

Circuit(id, nom, nbTour)

RéaliserUneCourse(#idPerso, #idCircuit, date, durée)

Voiture(id, nom, vitesse, poids, #idPerso)

1. Télécharger le script [marioKartLDD.sql](#)
2. Lancer mySQL
3. Exécuter le script `marioKartLDD.sql`
4. Créer le script `marioKartLMD.sql`
5. Ajouter les instructions au script `marioKartLMD.sql` permettant d'afficher :
 - a. Quels personnages ont leur nom commençant par un M ? Dans le résultat, vous affichez le nom, et la date de naissance.

```
SELECT nom, dateNai ssanceFROM PersonnageWHERE nom LIKE ' M%';
```

- b. Quels sont les voitures appartenant au personnage appelé "Mario" ? Dans le résultat, vous affichez le nom du personnage, le nom de la voiture et sa vitesse.

```
SELECT Personnage. nom, Voi ture. nom, Voi ture. vi tesseFROM Voi tureINNER JOIN PersonnageWHERE Voi ture. i dPerso = Personnage. i dAND Personnage. i d = (SELECT i d FROM Personnage WHERE nom = Mari o");
```

- c. Quel est le circuit qui a été réalisé le plus récemment ? Dans le résultat, vous affichez le nom du circuit, la date, et le nom du personnage ayant réalisé le circuit.

```
SELECT Personnage.nom, Circuit.nom,  
RealiserUneCourse.date_realisee  
FROM RealiserUneCourse, Personnage, Circuit  
WHERE Circuit.id = RealiserUneCourse.idCircuit  
AND RealiserUneCourse.idPerso = Personnage.id  
AND RealiserUneCourse.date_realisee = (SELECT max  
(date_realisee) FROM RealiserUneCourse as TAB);
```

- d. Quelles sont les voitures dont le nom commence par la lettre B, M ou W ? Dans le résultat, vous affichez le nom des voitures.

```
SELECT nom  
FROM Voiture  
WHERE nom Like 'B%'  
OR nom Like 'M%'  
OR nom Like 'W%';
```

- e. Quels sont les personnages qui n'ont pas de voiture dépassant le 5 km/h ? Dans le résultat, vous affichez le nom des personnages.

```
SELECT Personnage.nom, Voiture.nom, Voiture.vitesse  
FROM Personnage, Voiture  
WHERE Personnage.id = Voiture.idPerso  
AND Voiture.vitesse < 5;
```

- f. Classez les personnages en fonction du nombre de courses réalisées (même les personnages qui n'ont réalisés aucunes courses). Dans le résultat, vous affichez le nom des personnages, leur nombre de courses.

```
SELECT p.nom  
FROM Personnage AS p  
RIGHT JOIN Voiture AS v ON p.id = v.idPerso  
WHERE p.id NOT IN (SELECT idPerso FROM Voiture WHERE  
vitesse > 5 AND idPerso != "NULL");
```

- g. Quels sont les personnages qui ont réalisés tous les circuits ? Dans le résultat, vous affichez le nom des personnages.

- h. Quelles sont les voitures ayant un poids supérieur au poids moyen de toutes les voitures ? Dans le résultat, vous affichez le nom des voitures.

