

Métropole – 2025 – sujet2 - Correction

Exercice 2 (6 points)

Question 1

On ne peut pas choisir l'attribut nom comme clé primaire de la relation adherent car plusieurs adhérents peuvent avoir le même nom. Une clé primaire doit identifier de manière unique chaque enregistrement.

Commentaire : l'unicité est une propriété indispensable d'une clé primaire.

Question 2

La requête affiche le nom et l'éditeur de tous les jeux présents dans la table jeu, classés par ordre alphabétique croissant du nom du jeu.

Commentaire : la clause ORDER BY permet de trier les résultats en ordre croissant.

Question 3

Requête SQL permettant d'obtenir les jeux en cours d'emprunt :

```
SELECT nomJeu
FROM emprunt
WHERE dateRendu IS NULL;
```

Commentaire : un jeu non rendu possède une valeur NULL pour l'attribut dateRendu.

Question 4

Requête SQL pour afficher les adhérents ayant emprunté le jeu « Catan » :

```
SELECT nom , prenom
FROM adherent
JOIN emprunt ON adherent.idAdherent = emprunt.idAdherent
WHERE nomJeu = 'Catan';
```

Commentaire : la jointure permet de relier les informations des tables adherent et emprunt.

Question 5

Requête SQL permettant de mettre à jour la date de rendu de l'emprunt :

```
UPDATE emprunt
SET dateRendu = '2025-06-03'
WHERE id_emprunt = 1538;
```

Commentaire : la clause WHERE évite de modifier plusieurs enregistrements.

Question 6

Requête SQL demandée :

```
SELECT nomJeu , categorie
FROM jeu
WHERE anneeSortie >= 2010
AND ageMin < 10;
```

Commentaire : les conditions sont combinées à l'aide de l'opérateur logique AND.

Question 7

Les clés étrangères de la table participation sont :

- idEvenement, clé étrangère faisant référence à l'attribut idEvenement de la table evenement.
- idAdherent, clé étrangère faisant référence à l'attribut idAdherent de la table adherent.

Commentaire : une table d'association contient généralement les clés étrangères des deux tables qu'elle relie.

Question 8

Script Python permettant de créer le dictionnaire dict_emprunts :

```
dict_emprunts = {}
for jeu in liste :
    if jeu in dict_emprunts :
        dict_emprunts[jeu] = dict_emprunts[jeu] + 1
    else :
        dict_emprunts[jeu] = 1
```

Commentaire : on compte le nombre d'occurrences de chaque jeu dans la liste.

Question 9

Script Python permettant de générer le podium des jeux les plus empruntés :

```
valeurs = sorted ( set ( dict_emprunts.values ( ) ) , reverse=True )
podium = []
for v in valeurs[ : 3] :
    ligne = []
    for jeu in dict_emprunts :
        if dict_emprunts[jeu] == v :
            ligne.append ( jeu )
    podium.append ( ligne )
```

Commentaire : on trie d'abord les nombres d'emprunts, puis on regroupe les jeux ayant le même nombre d'emprunts.