

Corrigé d'Examen Base de données**Exercice 1**

1. Citez deux avantages majeurs de l'utilisation d'un SGBD (Système de Gestion de Bases de Données) par rapport à une gestion manuelle sur papier.
2. Parmi les propositions suivantes, laquelle définit le mieux une Clé Étrangère ?
 - a. Un attribut qui doit obligatoirement être unique dans sa table.
 - b. Un attribut qui établit un lien entre deux tables en faisant référence à une clé primaire.
 - c. Un attribut qui ne peut jamais être laissé vide (NOT NULL).
3. On suppose qu'on a une table Etudiant qui contient des données sur les étudiants. Prévoyez le résultat de l'exécution de chacune des séquences suivantes :
 - a. DELETE FROM Etudiant;
 - b. SELECT * FROM etudiant;
 - c. DROP TABLE Etudiant;
 - d. SELECT * FROM etudiant;
4. Pour trier les résultats d'une requête par ordre décroissant, on utilise :
 - a. SORT BY DESC
 - b. ORDER BY DESC
 - c. GROUP BY DOWN

Corrigé :

1.

Deux avantages majeurs :

- *Intégrité des données : le SGBD empêche les doublons et les incohérences grâce aux contraintes (clé primaire, clé étrangère, etc.).*
- *Recherche rapide : retrouver une information est quasi instantané grâce aux index, contre plusieurs minutes/heures sur papier.*

2.

Réponse : b. Un attribut qui établit un lien entre deux tables en faisant référence à une clé primaire.

3.

- a. *DELETE FROM Etudiant; → supprime toutes les lignes de la table, mais la table (structure, colonnes, contraintes) reste intacte. Résultat : table vide.*
- b. *SELECT * FROM etudiant; → la table existe toujours mais est vide. Résultat : 0 ligne retournée (pas d'erreur, juste un résultat vide).*
- c. *DROP TABLE Etudiant; → supprime la table entière : les données ET la structure disparaissent. La table n'existe plus du tout dans la base.*
- d. *SELECT * FROM etudiant; → ERREUR du type Table 'etudiant' doesn't exist(ou équivalent selon le SGBD)*

4.

*Réponse : b. ORDER BY DESC**Exemple : SELECT * FROM Etudiant ORDER BY note DESC;*

Exercice 2

On considère la base de données suivante pour la gestion des prêts de livres :

Auteur (idAuteur, nom, nationalite)

Livre (idLivre, titre, genre, idAuteur#)

Adherent (idAdh, nom, ville, anneeInscription, pointsFidelite)

Emprunt (idLivre#, idAdh#, date Sortie, dateRetourPrevue)

Écrivez les requêtes SQL correspondant aux demandes suivantes :

1. Créer la table Adherent avec les types de données appropriés.
2. Insérer un nouvel auteur: (10, 'Sidi', 'Mauritanienne').
3. Afficher le titre de tous les livres ainsi que le nom de leur auteur respectif.
4. Lister les noms des adhérents habitant à 'Nouadhibou' ou à 'Zouerat'.
5. Afficher les titres des livres dont le genre est 'Roman' et qui ont été écrits par des auteurs de nationalité 'Mauritanienne'.
6. Compter le nombre total de livres enregistrés pour chaque genre.
7. Trouver le nom des adhérents qui ont emprunté le livre intitulé 'Mon rêve pour la Mauritanie de demain'.
8. Afficher les adhérents inscrits en 2024 ayant plus de 50 points de fidélité.
9. Augmenter de 10 points le score de fidélité de tous les adhérents habitant à 'Rosso'.
10. Écrire une requête pour supprimer tous les emprunts effectués par les adhérents ayant les identifiants 111, et 333. 222

Corrigé :

1.

```
CREATE TABLE Adherent (  
    idAdh          INT          PRIMARY KEY,  
    nom            VARCHAR(50)  NOT NULL,  
    ville          VARCHAR(50),  
    anneeInscription INT,  
    pointsFidelite INT          DEFAULT 0  
);
```

2.

```
INSERT INTO Auteur (idAuteur, nom, nationalite) VALUES (10, 'Sidi',  
'Mauritanienne');
```

3.

```
SELECT L.titre, A.nom FROM Livre L  
JOIN Auteur A ON L.idAuteur = A.idAuteur;
```

4.

```
SELECT nom FROM Adherent WHERE ville IN ('Nouadhibou', 'Zouerat');
```

5.

```
SELECT L.titre FROM Livre L  
JOIN Auteur A ON L.idAuteur = A.idAuteur  
WHERE L.genre = 'Roman' AND A.nationalite = 'Mauritanienne';
```

6.

```
SELECT genre, COUNT(*) AS nombre FROM Livre GROUP BY genre;
```

7.

```
SELECT DISTINCT A.nom FROM Adherent A  
JOIN Emprunt E ON A.idAdh = E.idAdh  
JOIN Livre L ON E.idLivre = L.idLivre  
WHERE L.titre = 'Mon rêve pour la Mauritanie de demain';
```

8.

```
SELECT * FROM Adherent WHERE anneeInscription = 2024 AND pointsFidelite > 50;
```

9.

```
UPDATE Adherent SET pointsFidelite = pointsFidelite + 10 WHERE ville = 'Rosso';
```

10.

```
DELETE FROM Emprunt WHERE idAdh IN (111, 222, 333);
```

Exercice 3

Le ministère de la Santé souhaite informatiser la gestion des Centres de Vaccination au niveau national.

- Chaque centre de vaccination est identifié par un code unique, un nom et une adresse. Un centre est situé dans une seule Moughataa.
- Une Moughataa est caractérisée par son nom et sa population.
- Les Citoyens sont identifiés par leur NNI (Numéro National), leur nom, prénom et téléphone. Un citoyen appartient à une seule Moughataa de résidence.
- Plusieurs types de Vaccins sont disponibles (ex: AstraZeneca, Sinopharm, Pfizer). Chaque vaccin a un nom et un nombre de doses nécessaires (ex: 1 ou 2).
- Un Infirmier est défini par son matricule, son nom et sa spécialité. Un infirmier travaille dans un seul centre de vaccination.
- Lorsqu'un citoyen reçoit une Injection, on doit enregistrer la date de l'acte, le type de vaccin administré, l'infirmier qui a pratiqué l'injection et le centre où cela s'est déroulé. Sachant qu'un citoyen peut recevoir zéro ou plusieurs injections (selon le nombre de doses nécessaires), un type de vaccin peut être utilisé pour plusieurs injections, un infirmier peut réaliser plusieurs injections et dans un centre de vaccination on peut faire plusieurs injections.

Travail demandé :

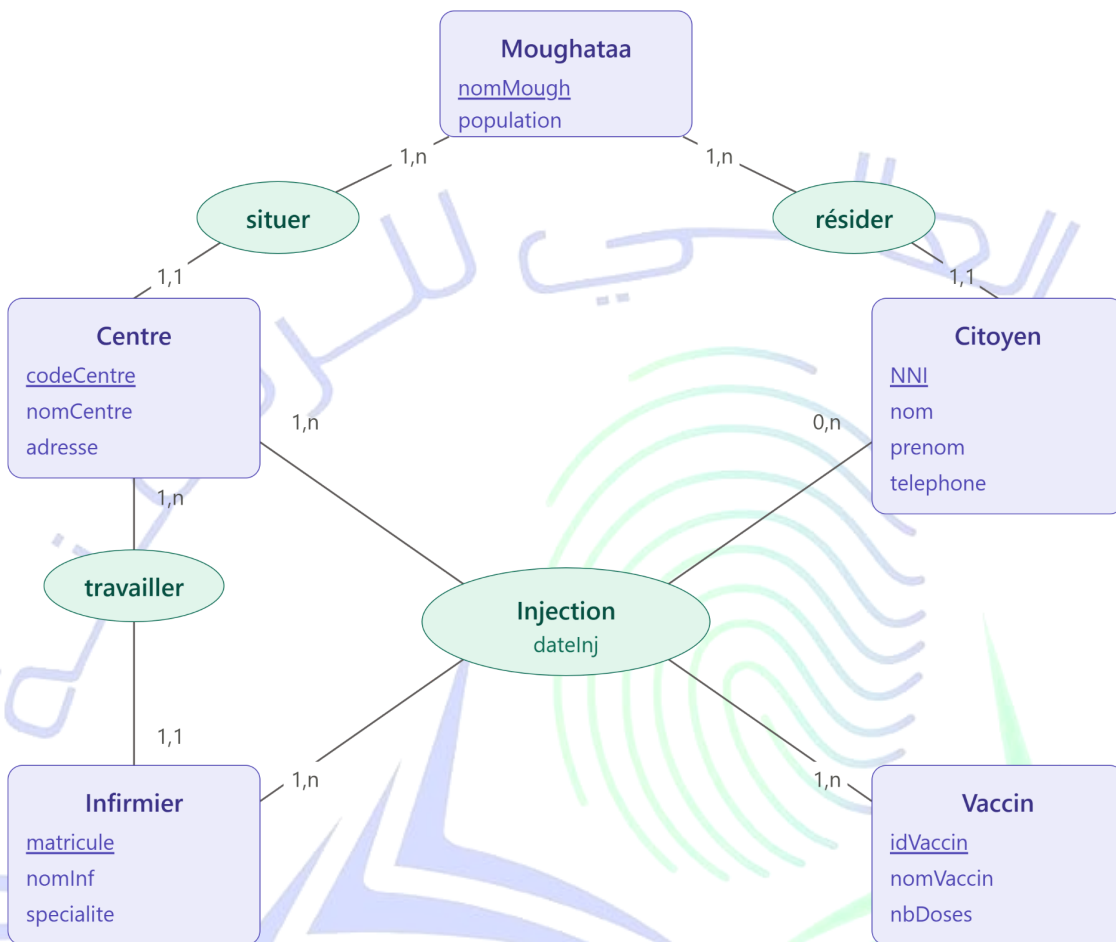
1. Donner le modèle conceptuel de données (MCD) en indiquant les attributs et les cardinalités.
2. Donner le schéma relationnel (MLD) correspondant.

Corrigé :

1.

Les 5 entités identifiées :

- *Moughataa* (nomMough, population)
- *Centre* (codeCentre, nomCentre, adresse)
- *Citoyen* (NNI, nom, prenom, telephone)
- *Infirmier* (matricule, nomInf, specialite)
- *Vaccin* (idVaccin, nomVaccin, nbDoses)



2.

MLD (Modèle Logique de Données)

- *Moughataa* (nomMough, population)
- *Centre* (codeCentre, nomCentre, adresse, #nomMough)
- *Citoyen* (NNI, nom, prenom, telephone, #nomMough)
- *Infirmier* (matricule, nomInf, specialite, #codeCentre)
- *Vaccin* (idVaccin, nomVaccin, nbDoses)
- *Injection* (#NNI, #idVaccin, #matricule, #codeCentre, dateInj)