

## Correction SQL – Modèle alternatif avec commentaires

Correction détaillée de l'exercice SQL – Modèle alternatif

1. Création des tables avec explication des choix :

```
CREATE TABLE ETUDIANT (  
    id_etudiant INT PRIMARY KEY, -- identifiant unique de l'étudiant  
    nom VARCHAR(50),  
    prenom VARCHAR(50),  
    niveau VARCHAR(10),  
    filiere VARCHAR(50)  
);  
  
CREATE TABLE AUTEUR (  
    id_auteur INT PRIMARY KEY, -- chaque auteur a un identifiant unique  
    nom_auteur VARCHAR(100),  
    nationalite VARCHAR(50)  
);  
  
CREATE TABLE LIVRE (  
    id_livre INT PRIMARY KEY,  
    titre VARCHAR(100),  
    annee_publication INT,  
    id_auteur INT,  
    FOREIGN KEY (id_auteur) REFERENCES AUTEUR(id_auteur) -- clé étrangère vers  
AUTEUR  
);  
  
CREATE TABLE EMPRUNT (  
    id_emprunt INT PRIMARY KEY,  
    id_etudiant INT,  
    id_livre INT,  
    date_emprunt DATE,  
    date_retour DATE,  
    FOREIGN KEY (id_etudiant) REFERENCES ETUDIANT(id_etudiant),  
    FOREIGN KEY (id_livre) REFERENCES LIVRE(id_livre)  
);  
  
2. Insertion de données avec un jeu cohérent :  
-- Etudiants  
INSERT INTO ETUDIANT VALUES (1, 'Lo', 'Awa', 'M1', 'SSI');
```

```
INSERT INTO ETUDIANT VALUES (2, 'Diop', 'Mamadou', 'L3', 'Informatique');
INSERT INTO ETUDIANT VALUES (3, 'Sow', 'Aminata', 'M2', 'Réseaux');
```

-- Auteurs

```
INSERT INTO AUTEUR VALUES (1, 'Mariama Bâ', 'Sénégal');
INSERT INTO AUTEUR VALUES (2, 'Chinua Achebe', 'Nigeria');
INSERT INTO AUTEUR VALUES (3, 'Camara Laye', 'Guinée');
```

-- Livres

```
INSERT INTO LIVRE VALUES (1, 'Une si longue lettre', 1980, 1);
INSERT INTO LIVRE VALUES (2, 'Le regard du roi', 1954, 3);
INSERT INTO LIVRE VALUES (3, 'Things Fall Apart', 1958, 2);
```

-- Emprunts

```
INSERT INTO EMPRUNT VALUES (1, 1, 1, '2023-01-15', '2023-01-25');
INSERT INTO EMPRUNT VALUES (2, 2, 3, '2024-05-10', '2024-05-20');
```

3. Requêtes SQL avec explications :

a. -- Affichage des étudiants

```
SELECT nom, prenom, niveau, filiere FROM ETUDIANT;
```

b. -- Livres empruntés avec dates

```
SELECT L.titre, E.date_emprunt, E.date_retour
FROM EMPRUNT E
JOIN LIVRE L ON E.id_livre = L.id_livre;
```

c. -- Nombre de livres par auteur

```
SELECT A.nom_auteur, COUNT(*) AS nb_livres
FROM LIVRE L
JOIN AUTEUR A ON L.id_auteur = A.id_auteur
GROUP BY A.nom_auteur;
```

d. -- Livres empruntés par des étudiants en informatique

```
SELECT L.titre
FROM ETUDIANT ET
JOIN EMPRUNT EM ON ET.id_etudiant = EM.id_etudiant
JOIN LIVRE L ON EM.id_livre = L.id_livre
WHERE ET.filiere = 'Informatique';
```

e. -- Étudiants sans emprunts

```
SELECT nom, prenom
FROM ETUDIANT
WHERE id_etudiant NOT IN (
```

```
SELECT id_etudiant FROM EMPRUNT  
);
```

```
f. -- Suppression d'emprunts vieux de plus de 2 ans  
DELETE FROM EMPRUNT  
WHERE date_emprunt < DATE_SUB(CURDATE(), INTERVAL 2 YEAR);
```

```
4. Ajout de la colonne 'genre' à LIVRE :  
ALTER TABLE LIVRE ADD genre VARCHAR(50);
```

```
5. Mise à jour des genres :  
UPDATE LIVRE SET genre = 'Roman' WHERE id_livre = 1;  
UPDATE LIVRE SET genre = 'Fiction' WHERE id_livre = 2;  
UPDATE LIVRE SET genre = 'Historique' WHERE id_livre = 3;
```