

TD 3 : SQL – Création des tables et insertion des données

L'objectif de ce travail dirigé est de vous aider à :

- Comprendre comment concevoir et créer une base de données relationnelle en fonction d'un modèle logique de données (MLD) donné.
- Utiliser les instructions SQL CREATE TABLE pour définir la structure des tables de la base de données, y compris les clés primaires, les clés étrangères et les contraintes.
- Se familiariser avec la syntaxe SQL utilisée pour insérer des données fictives dans les tables de la base de données pour simuler des scénarios réels.
- Utiliser les instructions ALTER TABLE pour apporter des modifications à la structure de la base de données, telles que l'ajout, la modification ou la suppression de colonnes, ainsi que l'ajout de contraintes.

Exercice 1 :

Soit le modèle relationnel suivant relatif à la gestion des notes annuelles d'une promotion d'étudiants :

- ETUDIANT (CodeEtudiant, Nom, Prenom)
 - MATIERE (CodeMat, LibelleMat, CoeffMat)
 - EVALUER (#CodeEtudiant, #CodeMat, Date, Note)
1. Créer la base de données correspondante à ce MLD en respectant les règles de gestion suivantes :
 - L'attribut nom de l'étudiant doit être non null.
 - Le coefficient de la matière a comme valeur par défaut 1.
 - La note a une valeur entre 0 et 20.
 2. Remplir la table ETUDIANT avec les enregistrements suivants :
 - 1, Bennis, Ahmad
 - 2, Dupont, Marie
 - 3, Nadin, Aya
 3. Remplir les autres tables sachant que :
 - On a deux matières Mathématique (avec un coefficient qui vaut 1 et date d'examen est 2023-01-10) et Informatique (avec un coefficient qui vaut 2 et date d'examen est 2023-01-15).
 - L'étudiant Bennis a eu 15 en Mathématique et 16 en Informatique.
 - L'étudiant Dupont a eu 14 en Mathématique et 17 en Informatique.
 - L'étudiant Nadin a eu 12 en Mathématique et 13 en Informatique.

===== SOLUTION =====

1. La création de la base de données :

```
CREATE TABLE ETUDIANT (  
  CodeEtudiant INT PRIMARY KEY,  
  Nom VARCHAR(20) NOT NULL,  
  Prenom VARCHAR(20)  
);
```

```
CREATE TABLE MATIERE (  
  CodeMat INT PRIMARY KEY,  
  LibelleMat VARCHAR(20),  
  CoeffMat INT DEFAULT 1  
);
```

```
CREATE TABLE EVALUER (  
  CodeEtudiant INT,  
  CodeMat INT,  
  Date DATE,  
  Note FLOAT,  
  PRIMARY KEY (CodeEtudiant, CodeMat),  
  FOREIGN KEY (CodeEtudiant) REFERENCES ETUDIANT(CodeEtudiant),  
  FOREIGN KEY (CodeMat) REFERENCES MATIERE(CodeMat),  
  CONSTRAINT ck_note CHECK (Note >= 0 AND Note <= 20)  
);
```

2. Remplir la table ETUDIANT avec les enregistrements suivants :

```
INSERT INTO ETUDIANT (CodeEtudiant, Nom, Prenom) VALUES  
(1, 'Bennis', 'Ahmad'),  
(2, 'Dupont', 'Marie'),  
(3, 'Nadin', 'Aya');
```

3. Remplir les autres tables :

```
-- Remplissage de la table MATIERE  
INSERT INTO MATIERE (CodeMat, LibelleMat, CoeffMat) VALUES  
(1, 'Mathématiques', 1),  
(2, 'Informatique', 2);
```

```
-- Remplissage de la table EVALUER  
INSERT INTO EVALUER (CodeEtudiant, CodeMat, Date, Note) VALUES  
(1, 1, '2023-01-10', 15),  
(1, 2, '2023-01-15', 16),  
(2, 1, '2023-01-10', 14),  
(2, 2, '2023-01-15', 17),
```

(3, 1, '2023-01-10', 12),
(3, 2, '2023-01-15', 13);

Exercice 2 :

Soit le MLD suivant d'une base de données de pharmacie :

- Medicament (ID_Medicament, Nom_Medicament, Fabricant_Medicament, Prix_Unitaire, Quantite_en_Stock)
 - Fournisseur (ID_Fournisseur, Nom_Fournisseur, Adresse_Fournisseur, Tele_Fournisseur)
 - Commande_Fournisseur (ID_Commande, Date_Commande, #ID_Fournisseur)
 - Detail_Commande_Fournisseur (ID_Detail_Commande, #ID_Commande, #ID_Medicament, Quantite)
1. Écrivez les requêtes SQL pour créer les tables pour ce modèle de données.
 2. Ajouter dans la table Medicament les données suivantes :
Aspirine, Bayer, 5.99, 100
Paracétamol, Sanofi, 3.49, 150
Amoxicilline, Pfizer, 8.75, 80
 3. Ajouter dans la table Fournisseur les données suivantes :
Pharma Distrib, 10 rue Soumaya, 0123456789
Médicaments SA, 15 avenue des FAR, 0234567890
Pharma Express, 5 rue Rome, 0345678901
 4. Insérer les 3 commandes suivantes dans la table Commande_Fournisseur :
2023-01-10, 1
2023-12-15, 2
2024-02-20, 3
 5. Insérer les détails des 3 commandes précédemment insérées :
1, 1, 100
2, 2, 150
3, 3, 80
 6. Modifier le type de données de la colonne "Tele_Fournisseur" dans la table Fournisseur pour le changer en VARCHAR(15).
 7. Renommer la colonne "Tele_Fournisseur" en "Numero_de_Telephone".
 8. Ajouter une contrainte UNIQUE sur la colonne "Nom_Medicament" dans la table Médicament pour s'assurer qu'aucun médicament n'a le même nom.
 9. Supprimer la colonne "Quantite_en_Stock" de la table Medicament.
 10. Supprimer la table Detail_Commande_Fournisseur.

===== SOLUTION =====

1. Écrivez les requêtes SQL pour créer les tables pour ce modèle de données.

-- Table Medicament

```
CREATE TABLE Medicament (  
  ID_Medicament INT AUTO_INCREMENT PRIMARY KEY,  
  Nom_Medicament VARCHAR(255),  
  Fabricant_Medicament VARCHAR(255),  
  Prix_Unitaire DECIMAL(10,2),  
  Quantite_en_Stock INT  
);
```

-- Table Fournisseur

```
CREATE TABLE Fournisseur (  
  ID_Fournisseur INT AUTO_INCREMENT PRIMARY KEY,  
  Nom_Fournisseur VARCHAR(100),  
  Adresse_Fournisseur VARCHAR(255),  
  Tele_Fournisseur VARCHAR(20)  
);
```

-- Table Commande_Fournisseur

```
CREATE TABLE Commande_Fournisseur (  
  ID_Commande INT AUTO_INCREMENT PRIMARY KEY,  
  Date_Commande DATE,  
  ID_Fournisseur INT,  
  FOREIGN KEY (ID_Fournisseur) REFERENCES Fournisseur(ID_Fournisseur)  
);
```

-- Table Detail_Commande_Fournisseur

```
CREATE TABLE Detail_Commande_Fournisseur (  
  ID_Detail_Commande INT AUTO_INCREMENT PRIMARY KEY,  
  ID_Commande INT,  
  ID_Medicament INT,  
  Quantite INT,  
  FOREIGN KEY (ID_Commande) REFERENCES Commande_Fournisseur(ID_Commande),  
  FOREIGN KEY (ID_Medicament) REFERENCES Medicament(ID_Medicament)  
);
```

2. Ajouter dans la table Medicament les données suivantes :

```
INSERT INTO Medicament (Nom_Medicament, Fabricant_Medicament, Prix_Unitaire,  
  Quantite_en_Stock) VALUES  
  ('Aspirine', 'Bayer', 5.99, 100),  
  ('Paracétamol', 'Sanofi', 3.49, 150),  
  ('Amoxicilline', 'Pfizer', 8.75, 80);
```

3. Ajouter dans la table Fournisseur les données suivantes :

```
INSERT INTO Fournisseur (Nom_Fournisseur, Adresse_Fournisseur, Tele_Fournisseur) VALUES  
( 'Pharma Distrib', '10 rue Soumaya', '0123456789'),  
( 'Médicaments SA', '15 avenue des FAR', '0234567890'),  
( 'Pharma Express', '5 rue Rome', '0345678901');
```

4. Insérer les 3 commandes suivantes dans la table Commande_Fournisseur :

```
INSERT INTO Commande_Fournisseur (Date_Commande, ID_Fournisseur) VALUES  
( '2023-01-10', 1),  
( '2023-12-15', 2),  
( '2024-02-20', 3);
```

5. Insérer les détails des 3 commandes précédemment insérées :

```
INSERT INTO Detail_Commande_Fournisseur (ID_Commande,  
ID_Medicament, Quantite) VALUES  
(1, 1, 100),  
(2, 2, 150),  
(3, 3, 80);
```

6. Modifier le type de données de la colonne "Tele_Fournisseur" dans la table Fournisseur pour le changer en VARCHAR(15).

```
ALTER TABLE Fournisseur  
MODIFY Tele_Fournisseur VARCHAR(15);
```

7. Renommer la colonne "Tele_Fournisseur" en "Numero_de_Telephone".

```
ALTER TABLE Fournisseur  
CHANGE COLUMN Tele_Fournisseur Num_Tele_Fournisseur VARCHAR(15);
```

8. Ajouter une contrainte UNIQUE sur la colonne "Nom_Medicament" dans la table Médicament pour s'assurer qu'aucun médicament n'a le même nom.

```
ALTER TABLE Medicament  
ADD CONSTRAINT uc_nom UNIQUE (Nom_Medicament);
```

9. Supprimer la colonne "Quantite_en_Stock" de la table Medicament.

```
ALTER TABLE Medicament  
DROP COLUMN Quantite_en_Stock;
```

10. Supprimer la table Detail_Commande_Fournisseur.

DROP TABLE Detail_Commande_Fournisseur;