

Cours : Introduction aux bases de données relationnelles et SQL

1. Qu'est-ce qu'une base de données relationnelle ?

Une base de données relationnelle est un système de gestion de bases de données (SGBD) qui utilise un modèle basé sur les relations entre les données, organisées sous forme de tables. Ces bases de données sont structurées pour permettre une gestion facile, rapide et sécurisée des informations. Les données sont stockées dans des tables composées de lignes (enregistrements) et de colonnes (champs).

Les principales caractéristiques d'une base de données relationnelle sont :

- **Tables** : les informations sont organisées en tables.
- **Colonnes** : chaque table est constituée de colonnes qui contiennent des types de données spécifiques.
- **Clé primaire (Primary Key)** : chaque ligne d'une table est identifiée de manière unique par une clé primaire.
- **Clé étrangère (Foreign Key)** : une colonne qui établit une relation avec la clé primaire d'une autre table.
- **SQL (Structured Query Language)** : un langage de programmation utilisé pour interroger, manipuler et gérer les bases de données.

2. Composants d'une base de données relationnelle

Les bases de données relationnelles sont basées sur plusieurs composants clés :

- **Table** : une collection de données organisées en lignes et en colonnes.
- **Enregistrement** : une ligne dans une table, également appelée "tuple".
- **Colonne** : une caractéristique d'une table, également appelée "attribut".
- **Clé primaire (Primary Key)** : une colonne ou un ensemble de colonnes utilisées pour identifier de manière unique chaque enregistrement dans une table.
- **Clé étrangère (Foreign Key)** : une colonne ou un ensemble de colonnes qui permet de créer une relation entre deux tables.

3. SQL (Structured Query Language)

SQL est un langage de programmation standard utilisé pour interagir avec les bases de données relationnelles. Il permet de créer, manipuler et interroger les données stockées dans une base de données. Voici les principaux types de commandes SQL :

- **Commandes de définition des données (DDL)** : ces commandes permettent de définir, modifier et supprimer des structures de base de données.
 - **CREATE** : pour créer des bases de données, des tables, des index, etc.

- **ALTER** : pour modifier une structure existante (par exemple, ajouter une colonne à une table).
 - **DROP** : pour supprimer des structures (tables, bases de données).
- **Commandes de manipulation des données (DML)** : ces commandes permettent de manipuler les données contenues dans les tables.
 - **SELECT** : pour interroger et récupérer des données.
 - **INSERT** : pour insérer des données dans une table.
 - **UPDATE** : pour mettre à jour des données existantes.
 - **DELETE** : pour supprimer des données d'une table.
- **Commandes de contrôle des données (DCL)** : ces commandes sont utilisées pour contrôler l'accès aux données.
 - **GRANT** : pour accorder des privilèges d'accès à une base de données.
 - **REVOKE** : pour révoquer des privilèges d'accès.
- **Commandes de contrôle de transaction (TCL)** : ces commandes permettent de gérer les transactions dans une base de données.
 - **COMMIT** : pour valider une transaction.
 - **ROLLBACK** : pour annuler une transaction.

4. Création d'une base de données et d'une table

Voici un exemple de la syntaxe SQL pour créer une base de données et une table.

- **Créer une base de données :**
- CREATE DATABASE NomDeLaBaseDeDonnées;
- **Créer une table :**
- CREATE TABLE Employes (
- ID INT PRIMARY KEY,
- Nom VARCHAR(100),
- Prenom VARCHAR(100),
- Age INT,
- Salaire DECIMAL(10, 2)
-);

Dans cet exemple, nous créons une table Employes avec cinq colonnes : ID, Nom, Prenom, Age et Salaire. La colonne ID est définie comme clé primaire, ce qui garantit que chaque employé a un identifiant unique.

5. Requêtes SQL de base

- **Sélectionner des données d'une table :**
- SELECT * FROM Employes;

Cette requête retourne toutes les lignes et toutes les colonnes de la table Employes.

- **Sélectionner des données spécifiques :**
- SELECT Nom, Prenom FROM Employes;

Cette requête retourne uniquement les colonnes Nom et Prenom de la table Employes.

- **Ajouter des données dans une table :**
- INSERT INTO Employes (ID, Nom, Prenom, Age, Salaire)
- VALUES (1, 'Dupont', 'Pierre', 30, 2500.00);

Cette requête insère un nouvel enregistrement dans la table Employes.

- **Mettre à jour des données dans une table :**
- UPDATE Employes
- SET Salaire = 2800.00
- WHERE ID = 1;

Cette requête met à jour le salaire de l'employé ayant l'ID 1.

- **Supprimer des données d'une table :**
- DELETE FROM Employes WHERE ID = 1;

Cette requête supprime l'enregistrement de l'employé ayant l'ID 1.

6. Clés primaires et étrangères

- **Clé primaire :** C'est une contrainte qui garantit que chaque enregistrement d'une table a une valeur unique pour la colonne (ou ensemble de colonnes) définie comme clé primaire. Exemple :
- CREATE TABLE Clients (
 - ClientID INT PRIMARY KEY,
 - Nom VARCHAR(100)
-);
- **Clé étrangère :** C'est une contrainte utilisée pour lier deux tables. Elle fait référence à la clé primaire d'une autre table, garantissant l'intégrité référentielle. Exemple :
- CREATE TABLE Commandes (
 - CommandeID INT PRIMARY KEY,
 - ClientID INT,
 - DateCommande DATE,
 - FOREIGN KEY (ClientID) REFERENCES Clients(ClientID)
-);

7. Normalisation des bases de données

La normalisation est le processus de structuration d'une base de données pour minimiser la redondance et améliorer l'intégrité des données. Il existe plusieurs formes normales (1NF, 2NF, 3NF, etc.), chacune ayant des règles spécifiques pour organiser les tables et les relations.

Conclusion

Les bases de données relationnelles et SQL sont des éléments essentiels pour gérer les informations dans les systèmes modernes. La maîtrise de SQL permet de créer, manipuler et sécuriser efficacement les données dans une base de données relationnelle. En suivant ce cours, vous serez en mesure de comprendre et d'appliquer les concepts fondamentaux de SQL pour gérer des bases de données complexes.