

Chapitre IV: Langage SQL

1. Introduction

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que l'application Access de Microsoft ou par les produits plus professionnels tels qu'Oracle. Il a fait l'objet de plusieurs normes ANSI/ISO dont la plus répandue aujourd'hui est la norme SQL2 qui a été définie en 1992.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution. Le langage SQL propose un langage de requêtes ensembliste et assertionnel. Néanmoins, le langage SQL ne possède pas la puissance d'un langage de programmation : entrées/sorties, instructions conditionnelles, boucles et affectations. Pour certains traitements il est donc nécessaire de coupler le langage SQL avec un langage de programmation plus complet.

De manière synthétique, on peut dire que SQL est un langage relationnel, il manipule donc des tables (i.e. des relations, c'est-à-dire des ensembles) par l'intermédiaire de requêtes qui produisent également des tables.

2. Catégories d'instructions

Les instructions SQL sont regroupées en catégories en fonction de leur utilité et des entités manipulées. Nous pouvons distinguer cinq catégories, qui permettent :

1. La définition des éléments d'une base de données (tables, colonnes, clés, index, contraintes...),
2. La manipulation des données (insertion, suppression, modification, extraction...),
3. La gestion des droits d'accès aux données (acquisition et révocation des droits),
4. La gestion des transactions,

2.1. Langage de définition de données

Le langage de définition de données (LDD, ou *Data Definition Language*, soit DDL en anglais) est un langage orienté au niveau de la structure de la base de données. Le LDD permet de créer, modifier, supprimer des objets. Il permet également de définir le domaine des données (nombre, chaîne de caractères, date, booléen...) et d'ajouter des contraintes de valeur sur les données. Il permet enfin d'autoriser ou d'interdire l'accès aux données et d'activer ou de désactiver l'audit pour un utilisateur donné.

Les instructions du LDD sont : **CREATE, ALTER, DROP...**

a. Création d'un schéma

Une base de données est définie par son schéma. Pour créer un schéma vide (sans tables), le langage SQL propose l'instruction suivante :

```
CREATE SCHEMA nom_schema
```

Où est `nom_schema` est remplacé par la nouvelle base de données.

b. Création d'une table

Une table est un ensemble de lignes et de colonnes. La création consiste à définir (en fonction de l'analyse) le nom de ces colonnes, leur format (type), la valeur par défaut à la création de la ligne (DEFAULT) et les règles de gestion s'appliquant à la colonne (CONSTRAINT).

La commande de création de tables la plus simple ne comportera que le nom et le type de chaque colonne de la table. À la création, la table sera vide, mais un certain espace lui sera alloué. La syntaxe est la suivante :

```
CREATE TABLE nom_table (  
nom_col1 TYPE1,  
nom_col2 TYPE2,  
... ) ;
```

Quand on crée une table, il faut définir les contraintes d'intégrité que devront respecter les données que l'on mettra dans la table.

▪ Types de données

Les principaux types de données peuvent être :

INTEGER : Ce type permet de stocker des entiers signés codés sur 4 octets.

BIGINT : Ce type permet de stocker des entiers signés codés sur 8 octets.

REAL : Ce type permet de stocker des réels comportant 6 chiffres significatifs codés sur 4 octets.

NUMERIC [(précision, [longueur])] : Ce type de données permet de stocker des données numériques à la fois entières et réelles avec une précision de 1000 chiffres significatifs.

CHAR (longueur) : Ce type de données permet de stocker des chaînes de caractères de longueur fixe.

DATE : Ce type de données permet de stocker des données constituées d'une date.

BOOLEAN : Ce type de données permet de stocker des valeurs Booléenne.

▪ Contraintes d'intégrité

NOT NULL : valeur null impossible.

UNIQUE : unicité d'un attribut.

PRIMARY KEY: clé primaire.

FOREIGN KEY: clé étrangère.

Une contrainte qui ne fait référence qu'à une seule colonne de la table peut faire partie intégrante de la définition de colonne.

Exemple : Pour créer dans la base de données **client_commande** la table **client**, la commande SQL est la suivante :

```
CREATE TABLE Client (  
NCLI          CHAR(8)      NOT NULL,  
NOM           CHAR(18)     NOT NULL,  
ADRESSE       CHAR(8)      NOT NULL,  
LOCALITE      CHAR(20)     NOT NULL,  
CAT           CHAR(2)      DEFAULT NULL,  
COMPTE        DECIMAL(2)  NOT NULL  
PRIMARY KEY (NCLI)      );
```

c. Suppression d'une table

Pour supprimer une table, le langage SQL propose l'instruction suivante :

```
DROP nom_table
```

Attention : Toutes les données ainsi que la structure de la table seront perdues à la suite de cette opération.

2.2. Langage de manipulation de données

Le langage de manipulation de données (LMD, ou *Data Manipulation Language*, soit DML en anglais) est l'ensemble des commandes concernant la manipulation des données dans une base de données. Le LMD permet l'ajout, la suppression et la modification de lignes, la visualisation du contenu des tables et leur verrouillage.

Les instructions du LMD sont : **INSERT**, **UPDATE**, **SELECT**, **DELETE**.

Ces éléments doivent être validés par une transaction pour qu'ils soient pris en compte.

a. Insertion de lignes dans une table

Pour ajouter une ligne dans une table on utilise l'instruction **INSERT**

```
INSERT INTO nom_table (Listes_coulannes)  
VALUES (liste_valeurs);
```

Exemple : Pour créer dans la base de données **client_commande** la table **client**, la commande SQL est la suivante :

```
INSERT INTO Client (NCLI, NOM, ADRESSE)  
VALUES ('ABD', 'AHMED', 'SKIKDA');
```

b. Modification de lignes d'une table

L'instruction pour modifier des valeurs dans les lignes d'une table est **UPDATE** :

```
UPDATE nom_table  
SET nom_colonne = nouvelle_valeur;
```

Exemple : Pour la base de données **client_commande**, la requête :

```
UPDATE produit
SET prix = prix*1.5;
```

c. Extraction de données d'une table

L'extraction de données fait l'objet d'une seule commande : la requête **SELECT**

▪ Extraction simple

La requête select la plus simple et permet l'affichage de toutes les lignes d'une table, mais en ne montrant que certaines colonne. Sa forme générale est :

```
SELECT liste_colonnes
FROM nom_table;
```

Exemple : Pour la base de données **client_commande**, la requête

```
SELECT NCLI, NOM, LOCALITE
FROM CLIENT
```

Affiche pour toutes les lignes de la table **CLIENT** seulement les valeurs des trois colonnes **NCLI**, **NOM**, et **LOCATE**.

Pour obtenir les valeurs de toutes les colonnes, la requête est :

```
SELECT *
FROM CLIENT
```

▪ Extraction de lignes sélectionnées (conditionné)

Une requête de sélection contient dans la clause **WHERE** des conditions qui permettent de ne sélectionner que certaines lignes d'une table. Sa forme générale est :

```
SELECT liste_colonnes
FROM nom_table
WHERE condition
```

Exemple : Pour la base de données **client_commande**, la requête

```
SELECT NCLI, NOM  
FROM CLIENT  
WHERE LOCALITE = 'SKIKDA'
```

N'affiche que les lignes de la table client dont la valeur de la colonne **LOCALITE** est égale à la chaîne de caractères 'SKIKDA'. De plus, seules les valeurs des colonnes **NCLI** et **NOM** seront affichées.

d. Suppression de lignes d'une table

L'instruction pour supprimer des lignes dans une table est **DELETE** :

```
SELECT FROM nom_table  
WHERE condition
```

où le format de la condition de la clause **WHERE** est le même que celui de l'instruction **SELECT**.

2.3. Langage de protections d'accès

Le langage de protections d'accès (ou *Data Control Language*, soit DCL en anglais) s'occupe de gérer les droits d'accès aux tables.

Les instructions du DCL sont : **GRANT**, **REVOKE**.

2.4. Langage de contrôle de transaction

Le langage de contrôle de transaction (ou *Transaction Control Language*, soit TCL en anglais) gère les modifications faites par le LMD, c'est-à-dire les caractéristiques des transactions et la validation et l'annulation des modifications.

Les instructions du TCL sont : **COMMIT**, **SAVEPOINT**, **ROLLBACK**, **SET TRANSACTION**