LE LANGAGE SQL

LDD : Langage de Définition des Données

M. RABI

3. Le langage de définition de données (LDD)

Cette partie présente le langage de définition de données (LDD) qui permet de spécifier le schéma d'une base de données relationnelle.

3.1 Création de base de données

La création d'une base de données est possible en utilisant le langage SQL. Même si les systèmes de gestion de base de données (SGBD) sont souvent utilisés pour la créer.

```
CREATE DATABASE ma_base
```

Syntaxe :

Avec MySQL, si une base de données porte déjà ce nom, la requête retournera une erreur. Pour éviter d'avoir cette erreur, il convient d'utiliser la requête suivante :

```
CREATE DATABASE IF NOT EXISTS ma_base
```

3. Le langage de définition de données (LDD)

3.2 Supprimer une base de données

En SQL, la commande DROP DATABASE permet de supprimer totalement une base de données et tout ce qu'elle contient. Cette commande est à utiliser avec beaucoup d'attention car elle permet de supprimer tout ce qui est inclus dans une base: les tables, les données, les vues, les index …

Syntaxe :

```
DROP DATABASE ma_base
```

Ne pas afficher d'erreur si la base n'existe pas

```
DROP DATABASE IF EXISTS ma_base
```

3. Le langage de définition de données (LDD)

3.3 Création de tables

Une table est un ensemble de lignes et de colonnes. La commande **CREATE TABLE** permet de créer une table en SQL. La création d'une table sert à définir les colonnes et le type de données qui seront contenus dans chacune des colonne (entier, chaine de caractères, date, valeur binaire …).

Syntaxe :

```
CREATE TABLE nom_de_la_table
(
colonne1 type_donnees,
colonne2 type_donnees,
colonne3 type_donnees,
colonne4 type_donnees
)
```

Autres types :

- MONEY
- BOOLEAN
- BLOB (Binary Long Object)
- ...

Les types de données

Pour chaque colonne que l'on crée, il faut préciser le type de données que le champ va contenir.

Les types standard:

- INTEGER ou INT, SMALLINT
- NUMERIC(X)
- DECIMAL(X,Y) ou NUMERIC(X,Y)
- FLOAT(X),
- CHAR(X)
- VARCHAR(X)
- DATE (AAAA-MM-JJ)
- DATETIME (AAAA-MM-JJ HH:MM:SS)

3. Le langage de définition de données (LDD)

3.3 Création de tables

Il est également possible de définir des contraintes telles que :

- **PRIMARY KEY** : indiquer si cette colonne est considérée comme clé primaire
- **AUTO_INCREMENT** : la valeur entière du champ s'incrémente automatiquement.
- **NOT NULL** : empêche d'enregistrer une valeur nulle pour une colonne.
- **DEFAULT** : attribuer une valeur par défaut si aucune données n'est indiquée pour cette colonne lors de l'ajout d'une ligne dans la table.
- **UNIQUE** : interdit que deux tuples (enregistrement) de la table aient la même valeur pour l'attribut.
- **CHECK** : contrôle la validité de la valeur de l'attribut spécifié dans la condition dans le cadre d'une restriction de domaine

Syntaxe :

```
CREATE TABLE nom_de_la_table
(
colonne1 type_donnees PRIMARY KEY NOT NULL
colonne2 type_donnees DEFAULT 'VALEUR'
colonne3 type_donnees,
colonne4 type_donnees
)
```

3. Le langage de définition de données (LDD)

3.3 Création de tables

Imaginons qu'on souhaite créer une table utilisateur, dans laquelle chaque ligne correspond à un utilisateur inscrit sur un site web.

Exemple :

```
CREATE TABLE utilisateur (
Id_User INT PRIMARY KEY AUTO_INCREMENT ,
nom VARCHAR(25),
prenom VARCHAR(25),
date_naissance DATE,
pays VARCHAR(25) DEFAULT 'Maroc',
ville VARCHAR(50),
code_postal VARCHAR(5),
email VARCHAR(50) ,
nombre_achat INT CHECK (nombre_achat >=0)
);
```

3. Le langage de définition de données (LDD)

3.3 Création de tables

- **La clé étrangère :**

Le langage SQL permet d'indiquer quelles sont les clés étrangères dans une table, autrement dit, quels sont les attributs qui font référence à une ligne dans une autre table. On peut spécifier les clés étrangères avec l'option **FOREIGN KEY**.

Exemple :

```
create table Achat
(
id int primary key auto_increment,
id_user int not null,
DateAchat Date,
Montant decimal(9,2),
Foreign key (id_user) references utilisateur (id)
);
```

3. Le langage de définition de données (LDD)

3.4 Modifier la structure d'une table

La commande ALTER TABLE en SQL permet de modifier une table existante. Il est ainsi possible d'ajouter une colonne, d'en supprimer une ou de modifier une colonne existante.

Syntaxe :

```
ALTER TABLE nom_table
instruction
```

Ajouter une colonne :

Syntaxe :

```
ALTER TABLE nom_table
ADD nom_colonne type_donnees
```

Exemple :

```
ALTER TABLE utilisateur
ADD adresse VARCHAR(255)
```

3. Le langage de définition de données (LDD)

3.4 Modifier la structure d'une table

Supprimer une colonne :

Syntaxe :

```
ALTER TABLE nom_table
DROP nom_colonne
```

Ou :

```
ALTER TABLE nom_table
DROP COLUMN nom_colonne
```

Exemple :

```
ALTER TABLE utilisateur
DROP code_postal
```

3. Le langage de définition de données (LDD)

3.4 Modifier la structure d'une table

Modifier le type d'une colonne :

Syntaxe :

```
ALTER TABLE nom_table
MODIFY nom_colonne type_donnees
```

Exemple :

```
ALTER TABLE utilisateur
MODIFY Code_Postal Int
```

Renommer une colonne :

```
ALTER TABLE nom_table
CHANGE colonne_ancien_nom
colonne_nouveau_nom type_donnees
```

```
ALTER TABLE utilisateure
CHANGE Code_postal CP VARCHAR(5)
```

3. Le langage de définition de données (LDD)

3.4 Modifier la structure d'une table

La modification d'une table consiste en plus des actions citées précédemment , d'ajouter des contraintes d'intégrité sur une colonne.

Syntaxe :

```
ALTER TABLE nom_table
add Constraint instruction
```

Exemples :

```
ALTER TABLE utilisateur
ADD Constraint UNIQUE (Telephone)
```

```
ALTER TABLE achat
ADD Constraint Check
(montant >=100)
```

3. Le langage de définition de données (LDD)

3.4 Modifier la structure d'une table

Renommer une table:

Syntaxe :

```
ALTER TABLE Table
RENAME TO new_name
```

Exemple :

```
ALTER TABLE utilisateur
RENAME TO User
```

Supprimer une table :

La commande DROP TABLE en SQL permet de supprimer définitivement une table d'une base de données. Cela supprime en même temps les éventuels index, trigger, contraintes et permissions associées à cette table.

```
DROP Table nom_table
```

```
DROP Table Achat
```