

Chapitre 1 : Langage de Définition des Données dans SQL

1. Introduction

Le Langage de Définition des Données (LDD) permet la création, la modification et la suppression des tables.

2. Les types de données

Les types de données les plus utilisés sont :

Type	Description
CHAR(n)	Chaîne de caractères de longueur fixe = n (n max 255). Une chaîne plus courte sera complétée par des espaces.
VARCHAR(n)	Chaîne de caractères de longueur variable de taille maximale = n (n max 255)
INT	Entier codé sur 4 octets. Valeurs de : -2147483648 à 2147483647
FLOAT	Réel codé sur 4 octets.
INT(n)	Entier à n chiffres
DATE	Date au format aaaa-mm-jj

3. Création des tables

La commande CREATE TABLE permet de créer une table en définissant le nom et le type de chacune des colonnes de la table. La syntaxe est la suivante :

```
CREATE TABLE
nom_table (Attribut_1
           Type_attribut_1,
           Attribut_2
           Type_attribut_2,
           .....
           Attribut_n Type_attribut_n) ;
```

Exemple :

```
CREATE TABLE emp  
(cin int(8),  
Prenom VARCHAR  
(30), Date_naissance  
DATE);
```

4. Les contraintes d'intégrité (C.I.)

Une C.I. concerne une colonne de la table et peut être aperçue comme étant une condition que doivent respecter les valeurs de cette colonne.

4.1. Motivations de la mise en place des contraintes d'intégrité

On utilise les contraintes pour :

- Appliquer des règles au niveau d'une table chaque fois qu'une ligne est insérée, mise à jour ou supprimée de cette table. La contrainte doit être satisfaite pour que l'opération réussisse.
- Empêcher la suppression d'une table s'il y a des dépendances entre elle et d'autres tables.

Les types de contraintes dans MySQL sont :

- NOT NULL
- UNIQUE
- PRIMARY KEY
- FOREIGN KEY
- CHECK

4.2. Contrainte NOT NULL

Cette contrainte interdit la présence de valeurs NULL dans la colonne à laquelle elle s'applique.

Exemple :

```
CREATE TABLE emp
(cin INT (8) NOT NULL,
Prenom VARCHAR(30),
Date_naissance DATE);
```

4.3. Contrainte UNIQUE

Une contrainte d'intégrité de type UNIQUE appliquée à une colonne exige que chaque valeur non nulle dans cette colonne soit unique (existe une seule fois).

Exemple :

```
CREATE TABLE emp
(cin INT(8),
Prenom VARCHAR(30),
Date_naissance DATE,
CONSTRAINT C2 UNIQUE (prenom));
```

4.4. Contrainte PRIMARY KEY

Une contrainte PRIMARY KEY crée une clé primaire pour la table.

Exemple :

```
CREATE TABLE emp
(cin INT (8),
Prenom VARCHAR(30),
Date_naissance DATE,
CONSTRAINT PK_C3 PRIMARY KEY (cin));
```

Remarque : PRIMARY KEY = UNIQUE + NOT NULL

4.5. Contrainte FOREIGN KEY

La contrainte FOREIGN KEY, ou contrainte d'intégrité référentielle, désigne une colonne ou une combinaison de colonnes comme étant une clé étrangère et établit une relation avec une clé primaire d'une autre table.

Exemple :

```
CREATE TABLE emp
(cin INT (8),
Prenom VARCHAR(30),
Date_naissance DATE,
Num_dept INT(2),
CONSTRAINT PK_C3 PRIMARY KEY (cin),
CONSTRAINT FK_C4 FOREIGN KEY (Num_dept) REFERENCES
Dept(deptno));
```

4.6. Contrainte CHECK

Elle définit une condition que chaque ligne doit obligatoirement satisfaire (peut être assimilée à une contrainte de domaine).

Exemple :

```
CREATE TABLE emp
(cin INT (8),
Prenom VARCHAR (30),
Date_naissance DATE,
Num_dept INT (2),
CONSTRAINT C5 CHECK (num_dept between 10 and 70));
```

5. Modification de la structure d'une table

La modification de la structure d'une table peut concerner :

- L'ajout d'attributs,
- La modification d'attributs,
- La suppression d'attributs.

5.1. Ajout d'un attribut

La syntaxe est comme suit : ALTER TABLE nom_table ADD (Attr₁ Type₁, Attr₂ Type₂, ...);

Exemple :

```
ALTER TABLE emp ADD (tel INT(8));
```

La colonne tel ajoutée est mise à NULL pour toutes les lignes de la table emp. Il en découle qu'on ne peut pas appliquer une contrainte sur cette colonne lui interdisant d'avoir une valeur NULL à moins qu'elle soit affectée à une valeur par défaut ou que la table soit vide. Il est à noter que l'affectation d'une valeur par défaut s'effectue avec le mot clé DEFAULT.

Par exemple, affecter la valeur par défaut 71000000 au numéro de téléphone revient à écrire la commande : ALTER TABLE emp ADD tel INT(8) DEFAULT 71000000;

5.2. Modification d'un attribut

La syntaxe est comme suit : ALTER TABLE nom_table MODIFY attr type ;

Exemple :

```
ALTER TABLE emp MODIFY prenom VARCHAR(50);
```

5.3. Suppression d'un attribut

La syntaxe est comme suit : ALTER TABLE nom_table DROP COLUMN attribut ;

Exemple :

```
ALTER TABLE emp DROP COLUMN date ;
```

6. Mise à jour des contraintes d'intégrité

6.1. Ajout d'une contrainte d'intégrité

La syntaxe est comme suit : ALTER TABLE nom_table ADD CONSTRAINT nom_contrainte ... ;

Exemple :

```
ALTER TABLE emp ADD CONSTRAINT DateUnique UNIQUE (Date_naissance) ;
```

Date_naissance est bien évidemment un des attributs de la relation emp.

6.2. Suppression d'une contrainte d'intégrité

La syntaxe est comme suit : ALTER TABLE nom_table DROP CONSTRAINT nom_contrainte ;

Exemple :

Rappelons que la contrainte C5 correspond à la contrainte suivante sur l'attribut num_dept de la table emp : CONSTRAINT C5 CHECK (num_dept between 10 and 70)). Pour supprimer cette contrainte, on écrit la commande suivante :

```
ALTER TABLE emp DROP CONSTRAINT C5 ;
```

7. Suppression d'une table

La suppression d'une table est effectuée par la requête DROP TABLE, celle-ci détruirait les données et la structure de la table.

```
DROP TABLE nom_table ;
```

Exemple :

```
DROP TABLE emp ;
```

supprimera la table emp (structure et données).

Attention : « DROP TABLE » supprime la structure en plus des données si elles existent, contrairement à « DELETE FROM » qui ne supprime que les données.