

**BASE DE DONNEES ET SYSTEMES DE GESTION DE BASE DE
DONNEES (SGBD)**

CHAPITRE III : GESTION DES TABLES

Sommaire

INTRODUCTION	3
CREATION D'UNE TABLE	3
MODIFICATION D'UNE TABLE	5
SUPPRESSION D'UNE TABLE	6
COMMANDE COMMENT ON	6

INTRODUCTION

Les trois commandes permettant de manipuler des tables sont :

- CREATE TABLE pour définir une nouvelle table dans la base de données ;
- ALTER TABLE pour modifier la structure d'une colonne ou ajouter une colonne à une table ;
- DROP TABLE pour retirer une table de la base de données.

CREATION D'UNE TABLE

La commande SQL permettant de définir une table est la suivante :

```
CREATE TABLE nom_table  
  
(nom_colonne1 type [UNIQUE] [NOT NULL]  
  
[, nom_colonne2 type [UNIQUE] [NOT NULL],... ] ) ;
```

Cette commande crée une table dont la valeur des colonnes est NULL ou non- définie. Si l'option NOT NULL est indiquée, il est impératif que cette colonne contienne une valeur qui sera introduite lors du remplissage de la table. D'autre part, si l'option UNIQUE est choisie, deux valeurs identiques ne pourront apparaître dans cette colonne.

Les types déterminent la valeur et la taille des données stockées dans la base de données ainsi que le format sous lequel elles seront affichées. Dans MySQL, les colonnes peuvent avoir les types suivants :

SMALLINT	nombre entier de cinq digits de précision, compris entre -99999 et +99999;
INTEGER	nombre entier de dix digits maximum, compris entre -9999999999 et +9999999999;
CHAR(I)	chaîne de caractères de longueur I fixe : $1 \leq I \leq 254$;
VARCHAR(I)	chaîne de caractères de longueur I variable : $1 \leq I \leq 254$;
LONG VARCHAR	chaîne de caractères de longueur variable d'au plus 32767 caractères;
DECIMAL (p, d)	nombre décimal de p digits dont d décimales: $d \leq 18$;
NUMBER ou FLOAT	tout nombre de précision inférieure ou égale à dix-huit digits. Si la précision et le nombre de décimales ne sont pas précisés, l'espace maximum autorisé est réservé en mémoire;
DATETIME	type spécial pour la date et l'heure. Le diagramme syntaxique pour les date et heure est le suivant: < jj-mm-aaaa jj-mm-aa jj/mm/aaaa jj/mm/aa mm/jj/aaaa mm/jj/aa mm-jj-aaaa mm-jj-aa > [00:00:00 hh:mm:ss hh:mm hh][AM PM];

Note

Le format standard d'une date qui constitue le résultat d'une commande est :

jj-mmm-aa hh:mm:ss AM | PM

Exemple : 12-jan-89 07:25:00 AM

En guise d'illustration de la commande de création de tables, nous donnons ci-dessous les instructions qui ont permis de créer les tables de la base de données appliquées à la compagnie aérienne du chapitre suivant :

```
CREATE TABLE passagers
(no_pass smallint NOT NULL,
nom varchar (20),
prenom varchar (20),
tel varchar (15),
rue varchar (20),
boite varchar (8),
localite varchar (20),
code varchar (8),
pays varchar (5),
nopasspt varchar (20)) ;
```

```
CREATE TABLE reservations
(no_pass smallint NOT NULL,
no_vol varchar (13) NOT NULL,
origine varchar (3),
destin varchar (3),
classe varchar (1));
```

```
CREATE TABLE vols
(no_vol char (13) NOT NULL,
no_av varchar (7),
date_dep datetime,
origine varchar (3),
date_arr datetime,
destin varchar (3),
escale varchar (3),
datarresc datetime,
classes varchar (3));
```

```
CREATE TABLE avions
(type varchar (7),
marque varchar (20),
modele varchar (15),
pays varchar (5),
auto smallint,
nb_pass smallint,
vit_croi smallint,
longue decimal (6,2),
enverg decimal (6,2),
moteurs varchar (20));
```

MODIFICATION D'UNE TABLE

Une modification apportée à une table sera réalisée par la commande ALTER ; cette commande permet à un utilisateur d'ajouter, de retirer ou de modifier une colonne, ou encore de renommer une colonne ou une table. La syntaxe générale en est :

```
ALTER TABLE nom_table  
  
    ADD nom_coll type [ (taille) ] [NOT NULL ]  
        [ , nom_col2 type [ (taille) ] [NOT NULL ] ,... ] |
```

```
DROP nom_coll [ , nom_col2 ,... ] |  
  
MODIFY nom_coll [type (longueur) ] [NOT NULL | NULL]  
  
    [ , nom-col2 [type (longueur)] [NOT NULL | NULL ] ,... ] |  
  
RENAME < TABLE nouveau_nom_table |  
  
    Nom_coll nouv_nom1 [ ,nom_col2 nouv_nom2 ,... ] > ;
```

- **ADD**

L'ajout d'une ou de plusieurs colonnes se fait en spécifiant ses caractéristiques, tout comme pour la commande CREATE TABLE.

Il faut noter que le fait d'ajouter une colonne n'affecte en rien les instructions précompilées ou les vues existantes.

Exemple

```
ALTER TABLE passagers  
    ADD (sexe char (1)) ;
```

- **DROP**

Le fait d'éliminer une colonne entraînera l'effacement de toutes les références à cette colonne dans la définition de table. Si des données sont associées à cette colonne, elles seront perdues.

Exemple

```
ALTER TABLE passagers  
    DROP sexe;
```

- **MODIFY**

Le type associé à une colonne ne peut être changé. Le paramètre de type n'est utilisé que quand on désire modifier la longueur d'une colonne de caractères, et cela uniquement pour augmenter cette longueur.

L'attribut NOT NULL pourra être ajouté à une colonne qui acceptait précédemment les valeurs NULL. Cependant, si la colonne contient des valeurs NULL, il ne sera pas possible de lui associer l'attribut NOT NULL.

Exemple

```
ALTER TABLE passagers
  MODIFY nom char(25) NOT NULL,
  MODIFY Tel char(12) NULL;
```

- **RENAME**

Le nom de la table peut être renommé. Il en est de même des noms de toutes ses colonnes ou d'une partie de celles-ci.

Exemple

```
ALTER TABLE passagers
  RENAME clients;

ALTER TABLE avions
  CHANGE marque constructeur varchar(50),
  CHANGE nb_pass nbre_passag integer(5);
```

SUPPRESSION D'UNE TABLE

Pour supprimer une table, on utilise la commande DROP dont la syntaxe est simplement :

```
DROP TABLE nom_table ;
```

La table nommée est éliminée, ainsi que les données incluses. Les instructions précompilées qui font références à la table ne sont pas effacées automatiquement. Par contre, les index et les vues associés à cette table sont éliminés, tout comme les privilèges attribués en relation avec la table.

Exemple

```
DROP TABLE clients ;
```

Remarque

Les opérations réalisées par les trois commandes CREATE, ALTER et DROP n'agissent pas sur les données mais bien sur le schéma de la base de données.

Le fait de renommer une colonne, par exemple, n'altère en rien les valeurs de cette colonne. Pour agir sur les données, les commandes INSERT, UPDATE et DELETE devront être utilisées.

COMMANDE COMMENT ON

Il n'est pas rare de voir à un moment donné des utilisateurs incapables de se souvenir des noms des colonnes d'une table ou d'une vue, ou même de retrouver les tables et vues qui leur appartiennent. SQL se charge de conserver une trace de cette information dans un ensemble de tables inhérentes au système. Entre autres informations, les noms des tables, des vues, de leurs propriétaires ainsi que les noms des colonnes y sont placés et sont accessibles aux utilisateurs par de simples instructions SQL.

La commande permettant d'ajouter ou de modifier un tel commentaire dans MySQL possède la forme générale :

- Ajout de commentaire sur une table :

```
ALTER TABLE `matable` COMMENT='mon commentaire' ;
```

Exemple :

```
ALTER TABLE passagers COMMENT='mon commentaire';
```

- Ajout de commentaire sur un champ d'une table à la création :

CREATE TABLE `matable` (nomCol1 TypeCol1 **COMMENT** "mon commentaire de coll", ...);

Exemple :

[CREATE TABLE](#) essai (nom [VARCHAR](#)(25) **COMMENT** "nom");

- Modifier un commentaire attaché à une colonne d'une table déjà existante :

ALTER TABLE `matable` **MODIFY** nomCol1 TypeCol1 **COMMENT** "mon nouveau commentaire de coll";

Exemple :

[ALTER TABLE](#) essai **MODIFY** nom [varchar](#)(10) **COMMENT** "mon nouveau commentaire";

- Pour la visualisation du commentaire (COMMENT) des tables de la base de données :

SHOW TABLE STATUS FROM 'nomDeLaBase' ;

Exemple :

SHOW TABLE STATUS FROM compaeri2021 ;

- Pour la visualisation des commentaires (COMMENT) des champs de la table :

SHOW CREATE TABLE 'nomTable' ;

Exemple :

[SHOW CREATE TABLE](#) essai ;

- Quelle est la taille (type) de ces (COMMENT) ?

SHOW FULL COLUMNS FROM 'nomTable' ;

Exemple :

[SHOW FULL COLUMNS FROM](#) essai;

- Les commentaires sur les colonnes sont contenus dans la table COLUMNS de la base INFORMATION_SCHEMA. On y accède avec la commande :

SELECT COLUMN_NAME, COLUMN_COMMENT
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='ma_table' AND TABLE_SCHEMA='ma_base' ;

Exemple :

[SELECT](#) COLUMN_NAME, COLUMN_COMMENT **FROM** INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='essai' [AND](#) TABLE_SCHEMA='compaeri2021';

En bref, il est donc possible à tout un chacun de créer des tables, d'en modifier le schéma et de les retirer par l'intermédiaire des commandes **CREATE TABLE**, **ALTER TABLE** et **DROP TABLE**.

L'ajout de commentaires peut constituer une aide parfois précieuse lors de la manipulation de grandes bases de données.