

Module BDD - LDD (Langage de Définition des Données)

La définition de données dans SQL permet la définition des objets manipulés par le SGBD. Les objets : Table, Vue, Index
Les commandes du LDD sont :

CREATE : Création des objets.

ALTER : Modification de la structure des objets.

DROP : Suppression des objets.

1. Création des tables :

Elle se fait avec la commande : **CREATE TABLE**

```
CREATE TABLE nom_table  
(  
  col_1 type [(taille)] [DEFAULT ValeurParDefaut] [NULL/NOT NULL] [contrainte de colonne] ,  
  col_2 type [(taille)] [DEFAULT ValeurParDefaut] [NULL/NOT NULL] [contrainte de colonne],  
  ...  
  col_n type [(taille)] [DEFAULT ValeurParDefaut] [NULL/NOT NULL] [contrainte de colonne]  
  [contrainte de tables]  
);
```

NB. Les parties mises entre crochets [] sont optionnelles (facultatives)

Avec :

Types : indique le type de données de l'attribut (colonne)

Taille : indique la valeur maximale de la longueur de l'attribut

Principaux types de données SQL

CHAR(n) : Chaîne de longueur fixe égale à n caractères.

INTEGER : Entier.

VARCHAR(n) : Chaîne de longueur maximale n.

DECIMAL(m,n) : Numérique sur m chiffres avec n décimales.

DATE : Date avec le jour, le mois, l'année.

TIME : Horaire avec heures, minutes, secondes.

DATETIME : Date et horaire réunis (non ANSI)

TEXT : Texte de longueur quelconque (non ANSI)

Exemple

Produit (Numprod, Desprod, Couleur, Poids, Qte_stk, qte_seuil, Prix)

Avec :

Numprod : de type numérique de taille 6,

Desprod : de type caractère variable de taille 15,

Couleur : de type caractère sur une position,

Poids : numérique sur huit positions dont trois chiffres après la virgule,

Qte_stk : numérique sur sept positions dont trois chiffres après la virgule,

Qte_seuil : numérique sur sept positions dont trois chiffres après la virgule,

Prix : numérique sur dix positions dont sept chiffres avant la virgule.

```
CREATE TABLE Produit  
(  
  Numprod integer NOT NULL ,  
  Desprod varchar (15),  
  Couleur char,  
  Poids numeric (8,3),  
  Qte_stk numeric (7,3),  
  Qte_seuil numeric (7,3),  
  Prix numeric (10,3)  
);
```

2. Définition des contraintes d'intégrité :

a) Contrainte de clé primaire : Primary key

Elle peut être définie comme contrainte de table ou comme contrainte de colonne.

Clé primaire comme contrainte de table selon la syntaxe :

```
CREATE TABLE Produit  
(  
  Numprod integer NOT NULL ,  
  Desprod varchar(15),  
  Couleur char,  
  Poids numeric (8,3),  
  Qte_stk numeric (7,3),  
  Qte_seuil numeric (7,3),  
  Prix numeric (10,3),  
  CONSTRAINT PK_Produit Primary Key (NumProd)  
);
```

Clé primaire comme contrainte de colonne :

```
CREATE TABLE Produit  
(  
  numprod integer Primary Key,  
  Desprod varchar(15),  
  couleur char,  
  Poids numeric (8,3),  
  Qte_stk numeric (7,3),  
  Qte_seuil numeric (7,3),  
  prix numeric (10,3)  
);
```

b) Contrainte : UNIQUE et NOT NULL

```
CREATE TABLE Produit  
(  
  Numprod integer Primary Key,  
  Desprod varchar (15) UNIQUE , // l'attribut desprod doit être unique pour chaque produit  
  Couleur char NOT NULL, // l'attribut couleur doit avoir obligatoirement une valeur  
  ....  
);
```

c) Contrainte de valeur avec la clause CHECK

Exemple : On suppose que le poids d'un produit doit être positif. La commande de création de la table Produit devient :

```
CREATE TABLE Produit
(
  Numprod integer primary key,
  Desprod varchar(15),
  Couleur char,
  Poids numeric (8,3),
  Qte_stk numeric (7,3),
  Qte_seuil numeric (7,3),
  Prix numeric (10,3),
  CONSTRAINT Ck1_Produit CHECK (Poids >=0)
);
```

Comme contrainte sur la colonne

```
CREATE TABLE Produit
(
  Numprod integer primary key,
  Desprod varchar(15),
  Couleur char,
  Poids numeric (8,3) CHECK (Poids >=0),
  Qte_stk numeric (7,3),
  Qte_seuil numeric (7,3),
  Prix numeric (10,3)
);
```

d) Contrainte de clé étrangère : Foreign Key

Soit le schéma suivant : **MAGASIN**(NumMag, Adresse, Surface) , **PRODUIT**(NumProd, DesProd, Couleur, Poids, Qte_Stk, CodMag)

La commande pour la création de la table Magasin étant :

```
CREATE TABLE Magasin
(
  NumMag numeric(6) primary Key,
  Adresse varchar(30),
  Surface numeric(7,3)
);
```

La commande pour la création de la table Produit peut être écrite de deux façons:

```
CREATE TABLE Produit
(
  Numprod integer primary key,
  Desprod varchar(15),
  Couleur char,
  Poids numeric ( 8,3),
  Qte_stk numeric (7,3),
  Qte_seuil numeric (7,3),
  Prix numeric (10,3),
  CodMag numeric (6),
  CONSTRAINT FK_Produit FOREIGN KEY (CodMag)
REFERENCES Magasin (NumMag)
);
```

```
CREATE TABLE Produit
(
  Numprod numeric(6) primary key,
  Desprod varchar(15),
  Couleur char,
  Poids numeric(8,3),
  Qte_stk numeric(7,3),
  Qte_seuil numeric(7,3),
  Prix numeric(10,3),
  CodMag numeric(6) REFERENCES Magasin(NumMag)
);
```

3. Modification de la structure d'une table : Elle se fait avec la commande **ALTER TABLE**

Plusieurs possibilités de modification de la structure de table

a) 1ère forme : Ajout de nouvelles colonnes à une table

```
ALTER TABLE nom_table
ADD (col1 type [(taille)] [null / not null] );
```

Exemple : ajouter une colonne type_clt

```
ALTER TABLE CLIENT
ADD type_clt char(3) ;
```

b) 2ème forme : modification des colonnes d'une table

```
ALTER TABLE nom_table
MODIFY (col1 type [(taille)] [null / not null];
```

Remarque : Pour modifier le nom d'une colonne :

```
RENAME COLUMN nom_table.ancien_nom TO
nom_table.nouveau_nom ;
```

Exemple : Changer le type_clt de char(3) en char(5) :

```
ALTER TABLE CLIENT
MODIFY type_clt char(5) ;
```

c) 3ème forme : Suppression de colonnes existantes

```
ALTER TABLE nom_table
DROP COLUMN ( col1 , col2 ,..., coln ) ;
```

Exemple : supprimer le champ ville de la table Magasin :

```
ALTER TABLE Magasin
DROP COLUMN ville ;
```

d) 4ème forme: ajout d'une contrainte

```
ALTER TABLE nom_table
ADD Constraint Def_de_contrainte ;
```

Exemple : Ajouter à la table « Magasin » la contrainte suivante : la surface doit être comprise entre 10 et 100 m2

```
ALTER TABLE Magasin
ADD CONSTRAINT ck1_magasin Check (surface between 10 and 100) ;
```

e) 5ème forme : Suppression d'une contrainte:

On peut effacer une clé primaire. La commande est :

```
ALTER TABLE nom_table DROP PRIMARY KEY
[CASCADE] ;
```

Remarque : L'option cascade est ajoutée pour pouvoir supprimer une clé primaire référencée.

```
ALTER TABLE magasin DROP PRIMARY KEY;
```

Suppression d'une contrainte autre que la clé primaire :

```
ALTER TABLE nom_table DROP CONSTRAINT nom_contrainte ;
```

Où Le nom de la contrainte c'est celui de la contrainte à supprimer

Exemple :

```
ALTER TABLE produit DROP CONSTRAINT Ck4_Produit ;
```

On peut supprimer une contrainte de clé étrangère . La

commande (sous MySQL) est :

```
ALTER TABLE nom_table DROP FOREIGN KEY
nom_contrainte ;
```

4. Suppression d'une table

Elle se fait avec la commande : **DROP TABLE**

Exemple :

```
DROP TABLE client ;
```

Pour changer le nom d'une table existante la commande est :

```
RENAME TABLE ancien_nom TO nouveau_nom
```