

Contexte d'information	<i>cours SQL</i>
Titre du document	Gestion des objets de base de données
Nom de fichier	sql02.pdf

1] La manipulation des tables

Un objet de base de données sert à stocker ou à référencer des données. Nous avons les tables, les vues, les regroupements, les index, les séquences et les synonymes.

La table représente la forme la plus simple de stockage des données dans une base de données relationnelle.

Une table se compose de colonnes (une au minimum) représentant des données d'un certain type et de lignes. La table peut se voir attribuer des contraintes.

- *La création d'une table*

On utilise la commande CREATE TABLE. Supposons que nous avons la table suivante recensant les prix de divers légumes issus de l'agriculture biologique :

Table **legumes**

num	nom	prix au kilo (en €)
1	Pommes de terre	1,5
2	Carotte	1,6
3	Oignon	4,1
4	Céleri	4,8
5	Poireau	4,6
6	Chou blanc	
7	Chou rouge	3,1

```
CREATE TABLE legumes
( num INTEGER PRIMARY KEY,
  nom VARCHAR(20) NOT NULL,
  prix FLOAT NULL);
```

Remarque : L'option PRIMARY KEY permet d'avoir une valeur unique et non NULL.

Ce qui s'explique par le fait que la donnée num est de type entier, non nul et servant de clé primaire ; la donnée nom de type chaîne de caractères de longueur 20 et non nulle ; la donnée prix de type décimal et éventuellement nul. Null indique une absence de valeur et non une valeur zéro.

- *La destruction d'une table*

L'inverse de la commande de création est celle (simple) de destruction DROP TABLE :

```
DROP TABLE legumes;
```

2] La manipulation des données dans une table

- *L'insertion*

Il existe trois formes d'insertion de données dans une table : l'insertion littérale, l'insertion de type complément et l'insertion par requête. Cette dernière ne sera pas vue dans cette fiche.

- **L'insertion littérale** : On l'appelle ainsi car elle respecte à la lettre la syntaxe de création de la table. Autrement dit dans notre exemple nous avons déclaré les champs suivants et dans l'ordre : num, nom, prix ; il faudra respecter à la fois le nombre et l'ordre de ces champs :

```
INSERT INTO legumes VALUES (8, 'Chou fris ', 2);
```

- **L'insertion compl mentaire** : Dans ce cas de figure, nous pouvons d cider d'un nombre et d'un ordre pr cis :

```
INSERT INTO legumes (num, nom) VALUES (9, 'Courge');
```

L'important est de respecter au niveau des valeurs, l'ordre d clar  des noms de colonnes. Remplir une table avec beaucoup de donn es se r v le vite fastidieux !

Une variante d'insertion consiste   passer par la commande LOAD DATA et qui consiste   r cup rer les donn es directement dans un fichier texte (donn es au format CSV : 3:Oignon:4.1). Notez le back quote pour le nom de la table

```
LOAD DATA INFILE 'legumes.txt' INTO TABLE `legumes` FIELDS TERMINATED BY ':';
```

- **La modification de valeur**

Cela passe par l'instruction UPDATE :

```
UPDATE legumes SET prix = 2.2 WHERE nom = "poireaux";
```

Ce qui s'explique par le fait que l'on met le champ prix   2,2 dans la ligne o  le champ est "poireaux". Notez que vous pouvez mettre plusieurs valeurs (du genre SET prix = 2.2, nom = "poireaux" WHERE num = 5) dans la partie SET.

- **La suppression de valeur**

Cela concerne la suppression d'une ligne enti re (bien  videmment, sinon on passe par UPDATE...) identifi e par une clause WHERE :

```
DELETE FROM legumes WHERE num = 5;
```

3] La modification de la structure d'une table

Apr s la cr ation d'une table il est possible d'ajouter ou de supprimer des colonnes. On comprend ais ment que dans le cas d'une suppression de colonnes les donn es contenues sont elles aussi supprim es. Dans le cas d'ajout, par contre elles n'existent pas pour les lignes d j   existantes.

Ces commandes sont donc   utiliser en dernier recours :   vous de bien r fl chir   la structure de votre table !

- **L'ajout d'une colonne**

Exemple de l'ajout de la colonne fournisseur dans la table l gumes :

```
ALTER TABLE legumes ADD COLUMN quantite INT;
```

Notez le type `INT` raccourci du type `INTEGER`.

- ***La suppression d'une colonne***

Le même exemple mais en suppression :

```
ALTER TABLE legumes DROP COLUMN quantite;
```