

TP N° 4 : Insertions et modifications dans la Base de données

Partie Rappel: Langage de Manipulation des Données (LMD)

La manipulation des données dans SQL se fait à l'aide du LMD (Langage de manipulation des données). La manipulation des données se traduit par l'une des opérations suivantes : l'ajout, la modification ou la suppression des données.

Les commandes du LMD sont :

- **INSERT** : Insertion des données dans une table de la base de données c'est-à-dire ajouter une (plusieurs) ligne (s) à une table de la base de données.
- **UPDATE** : Mise à jour des données d'une table de base de données
- **DELETE** : Suppression des données d'une table de base de données

1. Insertion des données : Elle se fait avec la commande : **INSERT**

- **Première forme** : Ici vous indiquez explicitement le nom des colonnes que vous voulez valoriser, et vous indiquez les valeurs dans le même ordre.

SYNTAXE:

```
INSERT INTO nom_table (nom_colonne_1, nom_colonne_2, ...)  
VALUES (valeur_colonne_1, valeur_colonne_2, ...) ;
```

Exemple : Soit la table Employés (Num_Employé, Nom, Salaire, Fonction, Date_Recrutement, Code_Service, Code_Département)

- Insérer le tuple suivant dans la table Employés (14, 'BEN ALIA', 63000, 'comptable' , NULL, 2 , NULL)

```
SQL > INSERT INTO employés (Num_Employé, Nom, Salaire, Fonction, Code_Service)  
> VALUES (12, 'Ait Ali', 48000, 'comptable', 2) ;
```

Remarque 1 : Vous n'êtes pas obligé de respecter l'ordre initial des colonnes comme vous n'êtes pas obligé de toutes les indiquer. En effet les colonnes qui ont une valeur par défaut ou qui peuvent avoir une valeur nulle, si elles n'apparaissent pas, sont mises soit à NULL, soit à leur valeur par défaut.

Deuxième forme : Insertion de plusieurs lignes à la fois

Il est possible d'ajouter plusieurs lignes à un tableau avec une seule requête. Pour ce faire, il convient d'utiliser la syntaxe suivante :

SYNTAXE:

```
INSERT INTO nom_table (nom_colonne_1, nom_colonne_2, ...)  
VALUES (valeur_colonne_1, valeur_colonne_2, ...),  
       (valeur_colonne_1, valeur_colonne_2, ...)  
       ...  
       (valeur_colonne_1, valeur_colonne_2, ...) ;
```

Exemple : Insérer les tuples suivants dans la table Employés

- (14, 'BEN ALIA', 63000, 'comptable' , NULL, 2 , NULL)
- (15, 'ZEKKAR', 64000, 'Enseignant' , NULL, 3 , NULL)

```
SQL > INSERT INTO employés (Num_Employé, Nom, Salaire, Fonction, Code_Service)  
> VALUES (12, 'Ait Ali', 48000, 'comptable', 2),  
> (15, 'ZEKKAR', 64000, 'Enseignant' , 3 ) ;
```

Troisième forme : Les noms des colonnes ne sont pas indiqués dans la clause INSERT.

SYNTAXE:

```
INSERT INTO nom_table  
VALUES (valeur_colonne_1, valeur_colonne_2, ...) ;
```

Exemple : Insérer le tuple suivant dans la table Employés (14, 'BEN ALIA', 63000, 'comptable' , NULL, 2 , NULL)

```
SQL > INSERT INTO employés  
> VALUES (14, 'BEN ALIA', 63000, 'comptable', NULL, 2, NULL)
```

Remarque : Dans ce cas, il faut *absolument respecter* l'ordre dans lequel les colonnes ont été créées et les valoriser toutes (ou écrire explicitement NULL pour les valeurs des colonnes non renseignées)

2. Suppression des données : Elle se fait avec la commande : **DELETE**

La commande **DELETE** permet de supprimer une ou plusieurs lignes, qui vérifie(nt) une condition

SYNTAXE:

```
DELETE FROM nom_table  
WHERE condition(s) ;
```

Exemples 1: Supprimer de tous les employés comptables du service numéro 3 :

```
SQL > DELETE FROM employés  
> WHERE Code_Service = 3 AND  
> Fonction = 'comptable';
```

Exemples 2: Supprimer de l'employé numéro 10

```
SQL > DELETE FROM employés  
> WHERE NumEmployé= 10;
```

3. Modification des données : Elle se fait avec la commande : **UPDATE**

La commande **UPDATE** permet de modifier la valeur d'une colonne pour une ou plusieurs lignes en particulier (qui vérifie(nt) un critère)

SYNTAXE :

```
UPDATE nom_table  
SET nom_colonne = valeur  
WHERE condition(s);
```

Exemple 1: Augmentation de 10% des salaires des employés Comptable

```
SQL > UPDATE employés  
> SET Salaire = Salaire * 1.1  
> WHERE Fontion = 'Comptable';
```

Remarque 1: Pour modifier la valeur d'une colonne de la même manière pour toutes les lignes, il suffit de ne pas indiquer de condition.

Exemple 2: Augmenter des salaires de tous les employés de 1% :

```
SQL > UPDATE employés  
> SET Salaire = Salaire * 1.01;
```

Remarque 2: Pour modifier plusieurs colonnes en même temps, en séparant les affectations par une virgule.

SYNTAXE:

```
UPDATE nom_table  
SET nom_colonne1 = valeur_1, nom_colonne_2 = valeur_2, ...
```

Exemple 3: Modifier les information des employés du service numéro 13, tel que le code du département de ces employés devient 100 et tout leurs salaires sont égale 30 000.

```
UPDATE employés  
SET Code_departement=1, Salaire =29000  
WHERE code_service=13;
```

Enfin, on peut utiliser le résultat d'une requête pour affecter de nouvelles valeurs :
UPDATE nom_table SET Colonne = Requête