

TP Bases de données L2-ACAD Section B TP- 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 base.
- **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**

Le langage SQL fournit l'instruction « **INSERT** » pour ajouter des nouveaux tuples (enregistrements) à une relation i.e nouvelles lignes pour une table, munie de nouvelles informations.

Syntaxe:

```
INSERT INTO nom_table VALUES (valeur_attribut1, valeur_attribut2,...);  
  
COMMIT ;
```

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

SYNTAXE:

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

Exemple : Soit la table Employes (NumEmploye, Nom, salaire, fonction, date_recrutement, Code_Service, Code_Departement)

```
SQL > INSERT INTO employés (NumEmploye, Nom, salaire, fonction, Code_service)  
2   > VALUES (12, 'Ait Ali', 48000, 'comptable', 2) ;
```

Remarque : 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.**

```
SQL > INSERT INTO Employes (NumEmploye, Nom, salaire, fonction, date_recrutement,  
2   > Code_Service, Code_Departement) VALUES (12, 'AIT ALI', 48000, 'Comptable',  
3   > to_date('1991-04-02','dd-mm-yyyy', 14, 02)
```

- Il faut mettre les chaines de caractères entre côtes exemple 'AIT ALI'
- Les valeurs d'attribut de type numérique on les met pas entre ' '
- Utiliser le mot clé **NULL** pour laisser un attribut vide.
- Il ne faut pas oublier de valider les insertions en exécutant la commande « *commit* »
- Après chaque exécution de l'instruction INSERT, une réponse de SQL est affichée :
 - **SQL> 1 Ligne créée**
- Après exécution de l'instruction COMMIT, la réponse de SQL affichée sera :
 - **SQL> validation effectuée**

Remarque: il faut absolument respecter l'ordre par lequel les colonnes ont été créées et les toutes valoriser (ou écrire explicitement NULL pour les valeurs des colonnes non renseignées).

```
INSERT INTO Employes (NumEmploye, Nom, salaire, fonction ) VALUES (12, 'AIT ALI', 48000, 'Comptable')
```

- Dans ce cas, les attributs date_recrutement, Code_Service, Code_Departement ne doivent pas avoir la contrainte **NOT NULL** ni par *défait* pour que l'insertion soit acceptée.

➤ **Deuxième forme** : Les noms de colonnes ne sont pas indiqués dans la clause INSERT.

SYNTAXE:

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

Exemple :

```
SQL > INSERT INTO employés  
> VALUES (12, 'Ait Ali', 4800, '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:

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

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

```
SQL > DELETE FROM employés  
      > WHERE code_service = 3 AND  
      > function = 'comptable';
```

Exemples 2: Suppression de l'employé numéro 10

```
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 informaticiens

```
UPDATE employés  
SET salaire = salaire * 1.1  
WHERE fontion = 'informaticien';
```

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% :

```
UPDATE employés  
SET salaire = salaire * 1.01;
```

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

```
UPDATE nom_table  
SET nom_colonneà1 = valeur_1, nom_colonne_2 = valeur_2, ...
```

Exemple 3: augmenter des salaires de tous les employés de 1% :

```
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.**