

Le Langage SQL

Le Langage de Manipulation des Données (LMD) INSERT - UPDATE - DELETE

L'instruction INSERT

L'instruction INSERT permet d'insérer un ou plusieurs enregistrements dans une table. Pour pouvoir insérer des données, il faut que la table se situe dans votre schéma, ou que vous ayez les droits requis sur cette table si elle se trouve dans un autre schéma.

La forme simplifiée d'une requête SQL INSERT est la suivante :

```
INSERT INTO [schema.]{table}           -- La table dans laquelle on insert
[(column [, column]...)]               -- Les colonnes concernées
{VALUES(expr [, expr]..) | subquery}    -- Les valeurs insérées
```

Exemples

1. Insertion d'un enregistrement dans la table « exe_departement » en respectant l'ordre des colonnes :

```
INSERT INTO exe_departement VALUES (1, 'Informatique');
```

2. Si la table a une valeur par défaut pour la septième colonne (le numéro du département), alors on peut écrire :

```
INSERT INTO exe_employe VALUES (1, 'Bon', 'Jean', 'Rue xyz', '1200', 'Genève', DEFAULT);
```

3. Insertion d'un enregistrement dans la table « exe_departement » en spécifiant l'ordre des colonnes :

```
INSERT INTO exe_departement (dep_no, dep_nom) VALUES (1, 'Informatique');
INSERT INTO exe_departement (dep_nom, dep_no) VALUES ('Base de données', 2);
```

4. Insertion d'un enregistrement dans une table en laissant des colonnes vides :

```
INSERT INTO exe_employe VALUES (1, 'Bon', 'Jean', NULL, NULL, NULL, 1);
```

Contextes d'utilisations :

1. Pour un développeur côté base de données :

Via un programme comme PL/SQL Developer ou SQL*Plus :

- i. Pour insérer des données de test
- ii. Pour vérifier le fonctionnement correct de la base
- iii. Pour valider les contraintes d'intégrité

2. Pour un développeur d'application

Via un programme Java, .NET, MS Access ou Component Pascal :

- i. Pour accéder à la base depuis un programme « client »

Le Langage SQL Le Langage de Manipulation des Données (LMD) - DELETE

L'instruction DELETE

Pour supprimer des enregistrements d'une table, il faut utiliser la commande DELETE dont la syntaxe simplifiée est :

```
DELETE [FROM] [schema.]{table} [alias]  -- Table concernée
[WHERE condition]                       -- Condition à satisfaire pour
                                         -- que les enregistrements
                                         -- soient supprimés
```

Exemple :

```
DELETE FROM exe_departement ; --Supprime tous les enregistrements de la table departement
```

Le Langage SQL

Le Langage de Manipulation des Données (LMD) - UPDATE

L'instruction UPDATE

L'instruction UPDATE permet de mettre à jour des valeurs contenues dans une ou plusieurs colonnes d'un ou plusieurs enregistrements d'une table.

La forme simplifiée d'une requête SQL UPDATE est la suivante :

```
UPDATE [schema.]{table}      -- La table dans laquelle on « update »
SET column = {expr}          -- Modification du contenu d'une
    [, column = {expr} ] ...-- ou plusieurs colonnes
[WHERE condition]            -- Spécification des enregistrements
                                -- concernés par la mise à jour
```

NB : ***expr** peut être une valeur spécifique, la valeur NULL, une fonction, une expression (un calcul), le mot clé DEFAULT (pour assigner la valeur par défaut de la colonne), ou encore une requête SQL retournant une valeur*

Exemples

1. Modification d'une colonne d'un enregistrement :

```
-- Seul le departement 3 est modifié
UPDATE exe_departement
    SET dep_nom = 'ADMINISTRATION'
    WHERE dep_no = 3;
```

2. Modification d'une colonne de plusieurs enregistrements :

```
-- Les départements 3, 4 (et suivants) seront modifiés
UPDATE exe_departement
    SET dep_localite = 'GENEVE'
    WHERE dep_no > 2;

-- Suppression des commissions de tous les employés de la table
  (pas de clause WHERE)
UPDATE exe_employe
    SET emp_commission = NULL;
```

3. Modification de plusieurs colonnes :

-- Les employés de type 'MANAGER' deviennent 'GESTIONNAIRE' et ont une augmentation de leur commission de 500.-

```
UPDATE exe_employe
SET emp_fonction = 'GESTIONNAIRE',
    emp_commission = emp_commission + 500
WHERE emp_fonction = 'MANAGER';
```

-- Pour que cela fonctionne aussi pour les employés n'ayant pas de commission, il faut traiter spécifiquement la valeur NULL (fonction NVL : si NULL alors 0)

```
UPDATE exe_employe
SET emp_fonction = 'GESTIONNAIRE',
    emp_commission = NVL(emp_commission,0) + 500
WHERE emp_fonction = 'MANAGER';
```

Le Langage SQL

Le Langage de Contrôle des Données (LCD/LCT) COMMIT - ROLLBACK

Validation des transactions

Les instructions LMD (INSERT, UPDATE, DELETE, ...) peuvent soit être validées soit être annulées.

La validation permet de confirmer les modifications en cours (INSERT, UPDATE, DELETE). Ceci permet notamment de rendre visible ces modifications à l'ensemble des sessions (utilisateurs).

L'annulation permet à l'inverse de retrouver l'état des données avant ces modifications (INSERT, UPDATE, DELETE).

Attention, seules les instructions émises depuis la dernière validation sont annulées.

La commande de validation des transactions est :

```
COMMIT ;
```

La commande d'annulation des transactions (jusqu'au dernier COMMIT) est :

```
ROLLBACK ;
```