

Manipulation de données avec SQL

Dr N. BAME

SQL : Langage de Manipulation de données

- **Insertion** : ajout de **nouvelles** données
- **Mise en jour** : modification de données **existantes**
- **Suppression** : enlever des données **existantes**

Manipulation de données

- Un langage de manipulation de données (LMD) est un langage permettant de **modifier les informations contenues** dans la base.
- Il existe trois commandes SQL permettant d'effectuer les trois types de modification des données :
 - ***INSERT*** : **ajout** de lignes
 - ***UPDATE*** : **mise à jour** de lignes
 - ***DELETE*** : **suppression** de lignes

Insertion de tuples

- La commande **INSERT** permet d'insérer une ligne dans une table en spécifiant les valeurs à insérer.

Syntaxe :

```
INSERT INTO nom_table(nom_col_1, ..., nom_col_N)  
VALUES (val_1, ... , val_N) ;
```

- La liste des noms de colonne est **optionnelle**.
 - Si elle est omise, la liste des colonnes sera par défaut la liste de **l'ensemble des colonnes** de la table dans **l'ordre de la création** de la table.
- Si une **liste de colonnes est spécifiée**, les colonnes ne **figurant** pas dans la liste **auront la valeur NULL**.

Insertion de tuples (2)

- Il est possible d'insérer dans une table des lignes *provenant d'une autre table*.

```
INSERT INTO nom_table(nom_col1, nom_col2, ...)  
SELECT ...
```

- Il est possible d'insérer dans une table plusieurs lignes à la fois:

```
INSERT INTO nom_table(nom_col_1, nom_col_2, ...)  
VALUES (val_11, val_21, ...),  
        (val_12, val_22, ...),  
        ...  
        (val_1N, val_2N, ...);
```

Exemple

- **Emp** (Eno, Ename, Title, City)
 - **Project**(Pno, Pname, Budget, City)
 - **Pay**(Title, Salary)
 - **Works**(Eno, Pno, Resp, Dur)
-
- Ajout de l'employé informaticien Ndiaga de numéro 15 habitant à Thies.
 - Ajout du projet Accacia localisé à Kaolack avec le numéro 26.
 - Ajouter à la fois les enregistrements suivants dans la table Pay :
 - Technicien, 300000
 - Comptable 400000
 - Medecin 1200000

Exemple

- **Emp** (Eno, Ename, Title, City)
 - **Project**(Pno, Pname, Budget, City)
 - **Pay**(Title, Salary)
 - **Works**(Eno, Pno, Resp, Dur)
-
- Ajout de l'employé informaticien Krépin de numéro 15 habitant à Thies.

```
INSERT INTO Emp VALUES (15, 'Krépin', 'Informaticien', 'Thies');
```

- Ajout du projet Accacia localisé à Kaolack avec le numéro 26.

```
INSERT INTO Project(pno, Pname, City) VALUES (26, 'Accacia', 'Kaolack');
```

- Ajouter à la fois les enregistrements suivants dans la table Pay :
 - Technicien, 300000
 - Comptable 400000
 - Medecin 1200000

```
INSERT INTO Pay values ( 'Technicien', 300000),  
                        ( 'Comptable', 400000),  
                        ( 'Medecien', 1200000);
```

Mises à jour de tuples

- La commande UPDATE permet de modifier les valeurs d'une ou plusieurs colonnes, dans une ou plusieurs lignes existantes d'une table.

Syntaxe

```
UPDATE nom_table  
SET    nom_col_1 = expression_1,  
        nom_col_2 = expression_2,  
        ...  
        nom_col_n = expression_n  
[WHERE cond];
```

- Les valeurs des colonnes *nom_col_1, nom_col_2, ..., nom_col_n* sont modifiées dans **toutes les lignes** qui **satisfont** le prédicat **cond**.
- En **l'absence** d'une clause **WHERE**, **toutes** les lignes sont **mises à jour**.
- Les expressions *expression_1, expression_2, ..., expression_n* peuvent faire référence aux anciennes valeurs de la ligne.

Exemple

- **Emp** (Eno, Ename, Title, City)
- **Project**(Pno, Pname, Budget, City)
- **Pay**(Title, Salary)
- **Works**(Eno, Pno, Resp, Dur)
- Augmenter de 20% les budgets des projets.

UPDATE Project

SET budget = budget*1.2 ;

Exemple

- **Emp** (Eno, Ename, Title, City)
 - **Project**(Pno, Pname, Budget, City)
 - **Pay**(Title, Salary)
 - **Works**(Eno, Pno, Resp, Dur)
-
- Augmenter de 40% les budgets des projets de Fatick.

```
UPDATE Project
SET budget = budget*1.4
WHERE city= 'Fatick' ;
```

Exemple

- **Emp** (Eno, Ename, Title, City)
 - **Project**(Pno, Pname, Budget, City)
 - **Pay**(Title, Salary)
 - **Works**(Eno, Pno, Resp, Dur)
-
- Changer le nom et la ville de l'employé de numéro 15 en Samba Ndiaye et Louga

```
UPDATE Emp
SET     ename= 'Samba Ndiaye',
        city= 'Louga'
WHERE eno= 15;
```

Exemple

- **Emp** (Eno, Ename, Title, City)
 - **Project**(Pno, Pname, Budget, City)
 - **Pay**(Title, Salary)
 - **Works**(Eno, Pno, Resp, Dur)
-
- Changer les villes des employés de noms Coumba Sow en Matam

```
UPDATE Emp  
SET city= 'Matam'  
WHERE ename= 'Coumba Sow';
```

Exemple

- **Emp** (Eno, Ename, Title, City)
 - **Project**(Pno, Pname, Budget, City)
 - **Pay**(Title, Salary)
 - **Works**(Eno, Pno, Resp, Dur)
-
- Augmenter de 40% les budgets des projets de Fatick et diminuer de 10% les budgets des projets de Dakar.

```
UPDATE Project
```

```
SET budget = budget*1.4
```

```
WHERE city= 'Fatick' ;
```

```
UPDATE Project
```

```
SET budget = budget*0.9
```

```
WHERE city= 'Dakar' ;
```

Suppression de tuples

- La commande **DELETE** permet de supprimer des lignes d'une table.
- La syntaxe est la suivante :

```
DELETE FROM nom_table  
[WHERE cond ];
```
- Toutes les lignes pour lesquelles **cond** est évalué à *vrai* sont supprimées.
- En l'absence de clause **WHERE**, toutes les lignes de la table sont **supprimées**.