LE LANGAGE DML: Mise à jour des données

Pour modifier les données d'une base, l'utilisateur dispose de 3 ordres SQL :

- INSERT
- UPDATE
- DELETE

INSERT

L'ajout de lignes dans une table (ou une vue) répond à la syntaxe suivante :

```
INSERT INTO NOM_DE_TABLE (NOMS DE COLONNES)
VALUES (LISTE DE VALEURS)
```

Le mode principal dans l'ordre `INSERT` pour ajouter des lignes est l'insertion directe avec la clause VALUES

L'attribution de valeurs est faite aux colonnes.

Exemples:

```
EMPLOYES et DEPART de structure
EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
DEPART (NODEPT, NOMDEPT)
```

Insérer l'employé 00140, de nom REEVES, de prénom HUBERT dans le département A00, de salaire 2100€

```
INSERT INTO EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
VALUES (00140,'REEVES','HUBERT','A00',2100)
```

On donne une valeur pour chacun des attributs spécifiés dans l'ordre `INSERT` ; les valeurs de la clause VALUES doivent correspondre avec la liste des colonnes , les attributs non spécifiés prennent la valeur NULL.

Insérer l'employé 00140, de nom REEVES, de prénom HUBERT dans le département A00

```
INSERT INTO EMPLOYES (NOEMP, NOM, PRENOM, DEPT)
VALUES (00140,'REEVES','HUBERT','A00')
```

La colonne Salaire prendra la valeur NULL pour cette ligne. Si cette colonne n'a pas été spécifiée comme pouvant être nulle, une erreur sera générée. La liste des colonnes peut être omise à condition que l'ordre d'insertion concerne toutes les colonnes de la table.

Nous pouvons insérer plusieurs lignes dans la table

```
INSERT INTO EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
VALUES (00140,'REEVES','HUBERT','A00', 2100), (00150,'JACQUARD','ALBERT','B00', 1800)
```

Une colonne ayant une propriété AUTO_INCREMENT peux faire partie de la liste des colonnes, dans ce cas, vous spécifier vous-même la valeur. Si vous ne spécifiez pas la colonne, AUTO_INCREMENT fait son travail et un nouveau numéro est inséré.

UPDATE

L'ordre UPDATE est utilisé pour modifier des lignes de tables existantes et est composé de trois clauses :

```
UPDATE <NOM DE TABLE>
SET <NOM COLONNE 1> = <VALEUR 1> [,… <NOM COLONNE n> = <VALEUR n>]
WHERE <condition>
```

- SET Nom des colonnes et leurs valeurs ou expressions mises à jour.
- WHERE Critère de sélection pour la mise à jour d'une ligne (optionnel)

Si la clause WHERE n'est pas codée, la table entière sera mise à jour.

Exemple 1: Augmenter le salaire de 20% de tous les employés

```
UPDATE EMPLOYES
SET SALAIRE = SALAIRE * 1,2
```

Exemple 2: Augmenter le salaire de 20% de l'employé de matricule 00040.

```
UPDATE EMPLOYES
SET SALAIRE = SALAIRE * 1,2
WHERE NOEMP = 00040
```

Exemple 3: Modifier le salaire (augmentation de 20%) de l'employé de matricule 00040, et son affectation dans le service A40

```
UPDATE EMPLOYES
SET SALAIRE = SALAIRE * 1,2, DEPT= 'A40'
WHERE NOEMP = 00040
```

DELETE

L'ordre DELETE utilise trois clauses pour supprimer une ou plusieurs lignes d'une table.

```
DELETE [FROM] <NOM DE TABLE>
FROM <NOM DE TABLE> [, … <NOM DE TABLE n>]
WHERE <PREDICAT>
```

- FROM Spécifie le nom de la table ou les lignes seront supprimées
- FROM Une 2eme clause FROM pour spécifier le nom d'autres tables utilisées pour fournir des critères
- WHERE Spécifie le critère de sélection (optionnel)

Si la clause WHERE n'est pas codée, toutes les lignes seront supprimées.

Exemple 1: Supprimer tous les employés de la table EMPLOYES

```
DELETE FROM EMPLOYES
```

Exemple 2: Supprimer les employés du département 'E21'

```
DELETE FROM EMPLOYES
WHERE WDEPT ='E21'
```