

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 :

```
1 | INSERT INTO NOM_DE_TABLE (NOMS DE COLONNES)
2 | 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

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

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

```
1 | INSERT INTO employes (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
2 | 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

```
1 | INSERT INTO employes (NOEMP, NOM, PRENOM, DEPT)
2 | 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 (avec un seul VALUES) :

```
1 | INSERT INTO employes (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
2 | VALUES
3 | (00140, 'REEVES', 'HUBERT', 'A00', 2100),
4 | (00150, 'JACQUARD', 'ALBERT', 'B00', 1800),
5 | (00999, 'LOPER", 'DAVE', 'C00', 900)
6 |
```

Une colonne ayant une propriété AUTO_INCREMENT ne fait pas partie de la liste des colonnes : On peut toutefois forcer sa valeur en spécifiant la colonne.

UPDATE

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

```
1 | UPDATE <NOM DE TABLE>
2 | SET <NOM COLONNE 1> = <VALEUR 1> [,... <NOM COLONNE n> = <VALEUR n>]
3 | 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)

Exemple 1 : Modifier le salaire de l'employé LOPER, qui gagne désormais 1000 € (au lieu de 900 €):

```
1 | UPDATE employes
2 | SET
3 | salaire = 1000
4 | WHERE nom = 'LOPER'
```

Exemple 2 : Modifier plusieurs valeurs d'un coup (on modifie les nom, prénom et adresse de l'employé 3) :

```
1 | UPDATE employes
2 | SET
3 | nom = 'LOPER',
4 | prenom = 'Dave',
5 | adresse = '15 avenue Tella'
6 | WHERE noemp = '00999'
```

Exemple 3 : Augmenter le salaire de 20% de tous les employés

```
1 | UPDATE employes
2 | SET
3 | salaire = salaire * 1.2
```

Attention, l'absence de clause WHERE impacte donc tous les enregistrements de la table

Exemple 3 : Augmenter le salaire de 20% de l'employé de matricule 00040.

```
1 | UPDATE employes
2 | SET salaire = salaire * 1.2
3 | WHERE noemp = 00040
```

Exemple 4 : Modifier le salaire (augmentation de 20%) de l'employé de matricule 00040, et son affectation dans le service A40 :

```
1 | UPDATE employes
2 | SET salaire = salaire * 1.2,
3 | dept = 'A40'
4 | WHERE noemp = 00040
```

DELETE

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

```
1 | DELETE [FROM] <NOM DE TABLE>
2 | WHERE <CLAUSE>
```

- FROM spécifie le nom de la table ou les lignes seront supprimées
- WHERE : spécifie le(s) critère(s) de sélection (optionnel)

Attention, en l'absence de clause WHERE , toutes les lignes seront supprimées (table vide).

Exemple 1 : Supprimer tous les employés de la table EMPLOYES :

```
DELETE FROM employes
```

Exemple 2 : Supprimer les employés du département E21 :

```
1 | DELETE FROM employes
2 | WHERE nodept = 'E21'
```

Exemple 3 : Supprimer les employés du département E21 qui habitent Amiens :

```
1 | DELETE FROM employes
2 | WHERE nodept = 'E21'
3 | AND ville = 'Amiens'
```