

LE LANGAGE DML: Mise à jour des données

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

- INSERT
- UPDATE
- DELETE

INSERT

L'instruction `insert` s'utilise de la façon suivante :

```
insert into nom_de_table (noms de colonnes)
values (liste de valeurs)
```

Exemples:

```
insert into employes (noemp, nom, prenom, dept, salaire)
values (00140, 'reeves', 'hubert', 'a00', 2100)
```

On donne une valeur pour chacune des colonnes spécifiées dans l'ordre `INSERT` ; les valeurs de la clause `VALUES` doivent correspondre à 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` peut faire partie de la liste des colonnes, dans ce cas, vous spécifiez vous-même la valeur. Si vous ne spécifiez pas la colonne, `AUTO_INCREMENT` fait son travail et un nouveau numéro est inséré.

UPDATE

L'instruction `UPDATE` est utilisée pour modifier des lignes de tables existantes:

```
update nom_de_la_table
set colonne_1 = valeur_1, colonne_2=valeur_2
where condition
```

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

Si la clause `WHERE` n'est pas renseignée, 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
where condition
```

- FROM Spécifie le nom de la table où les lignes seront supprimées
- 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'
```