

SQL : Langage de Manipulation de Données



Insertion de données

Insérer une valeur pour chaque colonne

```
INSERT INTO nom_table VALUES (valeur_colonne1, ...,  
    valeur_colonneN);
```

Le SGBD affectera les valeurs aux colonnes dans l'ordre

Exemple :

Enseignant				
<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris

num : clé primaire de la table

Insertion de données

Insérer une valeur pour chaque colonne

```
INSERT INTO enseignant VALUES (5, 'Cooper', 'David', 3000, 'Marseille');
```

Exemple :

Enseignant				
<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris
5	Cooper	David	3000	Marseille

Insertion de données

Insérer une valeur pour chaque colonne

```
INSERT INTO nom_table SET nom_colonne=  
valeur_colonne, ...;
```

Le SGBD affectera les valeurs aux colonnes dans l'ordre

Exemple :

Enseignant				
<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris

num : clé primaire de la table

Insertion de données

Insérer une valeur pour chaque colonne

```
INSERT INTO enseignant Set num=5, nom='Cooper' ,  
    prenom='David', salaire=3000, ville='Marseille';
```

Exemple :

Enseignant				
<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris
5	Cooper	David	3000	Marseille

Insertion de données

Insérer quelques valeurs (pas toutes)

```
INSERT INTO nom_table (valeur_colonneP,...,  
    valeur_colonneQ) VALUES (valeur_colonneP,...,  
    valeur_colonneQ);
```

Exemple :

```
INSERT INTO enseignant (num,nom,prenom) VALUES (6,'  
    Benatia','Sonia');
```

<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris
5	Cooper	David	3000	Marseille
6	Benatia	Sonia		

Ceci n'est possible que si les champs non-renseignés ne sont pas nuls

Insérer des valeurs à partir d'un script

```
SOURCE nom_fichier.sql;
```

Insertion de données

Insérer des valeurs à partir d'un script

```
SOURCE nom_fichier.sql;
```

ou

```
\. nom_fichier.sql;
```


Insertion de données

Insérer des valeurs à partir d'un script

```
SOURCE nom_fichier.sql;
```

ou

```
\. nom_fichier.sql;
```

Attention aux éventuelles erreurs

Insertion de plusieurs données

Insérer une valeur pour chaque colonne

```
INSERT INTO nom_table VALUES  
    (valeur1_colonne1, ..., valeur1_colonneN) ,  
    (valeur2_colonne1, ..., valeur2_colonneN) ,  
    ...  
    (valeurQ_colonne1, ..., valeurQ_colonneN) ;
```

Insertion de plusieurs données

Insérer une valeur pour chaque colonne

```
INSERT INTO nom_table VALUES
    (valeur1_colonne1, ..., valeur1_colonneN) ,
    (valeur2_colonne1, ..., valeur2_colonneN) ,
    ...
    (valeurQ_colonne1, ..., valeurQ_colonneN) ;
```

Pour l'exemple précédent

```
INSERT INTO enseignant VALUES
    (1, Durand, Philippe, 2000, Marseille) ,
    (2, Leberre, Bernard, 1500, Paris) ,
    (3, Benammar, Pierre, 1800, Lyon) ,
    (4, Hadad, Karim, 1500, Paris) ,
    (5, Cooper, David, 3000, Marseille) ;
```

Suppression de données

Supprimer des tuples respectant une ou plusieurs conditions

```
DELETE FROM nom_table  
WHERE condtions;
```

Exemple :

```
DELETE FROM enseignant  
WHERE salaire > 2000 AND ville = 'Marseille';
```

<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris
5	Cooper	David	3000	Marseille
6	Benatia	Sonia		

Suppression de données

Supprimer tous les tuples d'une table

```
DELETE FROM nom_table;
```

Suppression de données

Supprimer tous les tuples d'une table

```
DELETE FROM nom_table;
```

Cette requête supprime toutes les données de la table, mais pas la table. Donc, le résultat est une table vide.

Modification de données

Modifier des tuples respectant une ou plusieurs conditions

```
UPDATE nom_table  
SET nom_colonne = valeur  
WHERE condition;
```

Exemple :

```
UPDATE Enseignant  
SET salaire= 1600, ville = 'Toulouse'  
WHERE nom = 'benatia';
```

<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Paris
3	Benammar	Pierre	1800	Lyon
4	Hadad	Karim	1500	Paris
6	Benatia	Sonia	1600	Toulouse

Modifier tous les tuples

```
UPDATE enseignant  
SET ville = 'Marseille' ;
```


Modification de données

Modifier tous les tuples

```
UPDATE enseignant  
SET ville = 'Marseille' ;
```

Le résultat

<u>num</u>	nom	prenom	salaire	ville
1	Durand	Philippe	2000	Marseille
2	Leberre	Bernard	1500	Marseille
3	Benammar	Pierre	1800	Marseille
4	Hadad	Karim	1500	Marseille
6	Benatia	Sonia	1600	Marseille

Quelques remarques

- Les champs dont les valeurs sont auto_incrementales sont gérées par le SGBD, mais on ne peut forcer le système à leur affecter une valeur de notre choix si elle n'existe pas dans la table.
- Dans ce cas et si la dernière valeur affectée est supérieure à la précédente, le SGBD incrémente cette valeur et l'affecte au prochain tuple.