

Secteur Tertiaire Informatique Filière étude - développement

Activité « Développer la persistance des données »

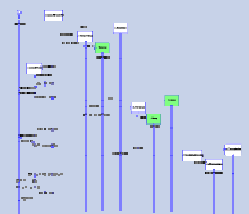
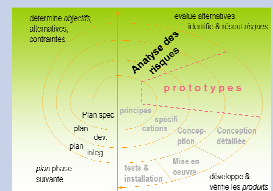
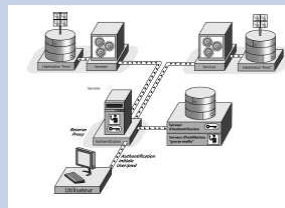
LMD 1 : INSERT UPDATE DELETE

Accueil

Apprentissage

Période en
entreprise

Evaluation



SOMMAIRE

I	L'ORDRE INSERT	4
II	L'ORDRE UPDATE	7
III	L'ORDRE DELETE	9

I L'ORDRE INSERT

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

```
INSERT [INTO] <NOM DE TABLE> [<NOMS DE COLONNES>]
{ DEFAULT VALUES
| VALUES <LISTE DE VALEURS>
| < REQUETE SELECT >
| EXECUTE <NOM DE PROCEDURE STOCKEE>}
```

Les modes principaux disponibles dans l'ordre INSERT pour ajouter des lignes sont :

- **l'insertion directe avec la clause VALUES**

L'attribution de valeurs est faite aux colonnes.

Pour une valeur par défaut, il faut spécifier DEFAULT ou DEFAULT VALUES.

Exemple 1: Insérer l'employé 00140, de nom REEVES, de prénom HUBERT dans le département A00, de salaire 2100€ dans la table EMPLOYES de structure EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)

```
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.

Exemple 2: 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.

Exemple 3: Insérer l'employé 00140, de nom REEVES, de prénom HUBERT dans le département A00, de salaire 2100€

```
INSERT INTO EMPLOYES  
VALUES (00140,'REEVES','HUBERT','A00', 2100)
```

• l'insertion à l'aide de sous requêtes en utilisant SELECT

Les données à insérer sont issues d'une table.

Exemple 4: Insérer dans la table EMPLOYES_A00 préalablement créée de structure EMPLOYES_A00 (NOEMP, NOM, PRENOM, SALAIRE) tous les employés de la table EMPLOYES attachés au département A00.

```
INSERT INTO EMPLOYES_A00 (NOEMP, NOM, PRENOM, SALAIRE)  
SELECT NOEMP, NOM, PRENOM, SALAIRE  
FROM EMPLOYES  
WHERE WDEPT = 'A00'
```

ou

```
INSERT INTO EMPLOYES_A00  
SELECT * FROM EMPLOYES  
WHERE WDEPT = 'A00'
```

- **l'insertion à l'aide d'une procédure stockée**

L'insertion est déléguée à la procédure stockée appelée.

Exemple 5: Insérer dans la table EMPLOYES_A00 préalablement créée de structure EMPLOYES_A00 (NOEMP, NOM, PRENOM, SALAIRE)

tous les employés de la table EMPLOYES attachés au département A00.

```
INSERT INTO EMPLOYES_A00 (NOEMP, NOM, PRENOM, SALAIRE)
```

```
EXECUTE EMP_A00
```

Les employés concernés sont sélectionnés dans la procédure stockée EMP_A00

Une colonne ayant une propriété IDENTITY ne fait pas partie de la liste des colonnes :

On peut toutefois forcer sa valeur en utilisant l'instruction

```
SET IDENTITY_INSERT nom_table{ ON | OFF }
```

On désactivera la propriété IDENTITY, on insérera la ligne et on réactivera la propriété IDENTITY.

La clause DEFAULT VALUES force la nouvelle ligne à prendre les valeurs par défaut définies pour chaque colonne

(Voir l'aide en ligne SQL Server pour plus d'informations).

II L'ORDRE 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>]
FROM <NOM DE TABLE 1>[, ... <NOM DE TABLE n>]
WHERE <PREDICAT>
```

- **SET** Nom des colonnes et leurs valeurs ou expressions mises à jour.
- **FROM** Nom d'autres tables utilisées pour fournir des critères
- **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 pour la table EMPLOYES de structure

EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)

```
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
```

On pourra utiliser l'ordre UPDATE combiné à une sous requête.

Exemple 4: Augmenter le salaire de 20% des employés du service informatique (repéré par son nom dans la table DEPART), pour les tables EMPLOYES et DEPART de structure
EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
DEPART (NODEPT, NOMDEPT)

```
UPDATE EMPLOYES
  SET SALAIRE = SALAIRE * 1,2
  WHERE DEPT IN (SELECT NODEPT FROM DEPART
                 WHERE NOMDEPT LIKE 'SERVICE%')
```

On pourra également utiliser l'ordre UPDATE et sa clause FROM pour utiliser des informations provenant d'une autre table.

Exemple 5: Augmenter le salaire de 20% des employés du service informatique (repéré par son nom dans la table DEPART), pour les tables EMPLOYES et DEPART de structure
EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
DEPART (NODEPT, NOMDEPT)

```
UPDATE EMPLOYES
  SET SALAIRE = SALAIRE * 1, 2
  FROM EMPLOYES JOIN DEPART ON DEPT = NODEPT
  WHERE NOMDEPT LIKE 'SERVICE%'
```

III L'ORDRE 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 de structure EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)

```
DELETE FROM EMPLOYES
```

Exemple 2: Supprimer les employés du département 'E21'

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

On pourra utiliser l'ordre DELETE combiné à une sous requête.

Exemple 3: Supprimer tous les employés du service informatique (repéré par son nom dans la table DEPART), pour les tables EMPLOYES et DEPART de structure EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)
DEPART (NODEPT, NOMDEPT)

```
DELETE FROM EMPLOYES
WHERE DEPT IN (SELECT NODEPT FROM DEPART
               WHERE NOMDEPT LIKE 'SERVICE%')
```


On pourra de la même façon utiliser l'ordre DELETE et sa deuxième clause FROM pour utiliser des informations provenant d'une autre table.

Exemple 4: Supprimer tous les employés du service informatique pour les tables EMPLOYES et DEPART de structure

EMPLOYES (NOEMP, NOM, PRENOM, DEPT, SALAIRE)

DEPART (NODEPT, NOMDEPT)

DELETE FROM EMPLOYES

FROM EMPLOYES JOIN DEPART ON DEPT = NODEPT

WHERE NOMDEPT LIKE 'SERVICE%'

Etablissement référent

Marseille Saint Jérôme

Equipe de conception

Elisabeth Cattaneo

Remerciements :

Aux formateurs de Marseille Saint Jérôme

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction

Date de mise à jour 05/01/2015
afpa © Date de dépôt légal janvier 15



afpa / Direction de l'Ingénierie 13 place du Général de Gaulle / 93108 Montreuil
Cedex
association nationale pour la formation professionnelle des
adultes
Ministère des Affaires sociales du Travail et de la
Solidarité