



Chapitre 2 : Langage de Manipulation des Données (LMD) dans SQL

Tarek HAMROUNI
Maître de Conférences en Informatique
hamrouni.tarek@gmail.com

Langage de Manipulation des Données (LMD)

- ▶ Langage de Manipulation des Données (LMD) : ensemble des commandes concernant la manipulation des données dans une base de données.
- ▶ Le LMD permet l'ajout, la suppression et la modification de lignes dans une table. Toute modification effectuée (ajout d'une ligne, modification d'une ligne ou suppression d'une ligne) doit respecter les contraintes d'intégrité (clé primaire, clé étrangère, etc.) ainsi que les contraintes de domaine et de présence dans le cas d'ajout et de modification.
- ▶ **LMD : INSERT, UPDATE, DELETE**

Insertion d'une ligne : INSERT ... VALUES

```
INSERT INTO nomTable [(col_1, ..., col_n)]  
VALUES (val_1, ..., val_n) ;
```

- La liste des noms de colonne est optionnelle. Si elle est omise, la liste des colonnes sera par défaut la liste de l'ensemble des colonnes de la table dans l'ordre de la création de la table.
- Si une liste de colonnes est spécifiée, les colonnes ne figurant pas dans la liste auront la valeur **NULL** ou la valeur **par défaut si elle est définie**.

Insertion du résultat d'une sous-requête : INSERT ... SELECT

Il est possible d'insérer dans une table des lignes provenant d'une autre table.

```
INSERT INTO nomTable [(col_1,..., col_n )]  
SELECT col_1', col_2', ..., col_n'  
FROM ...  
[WHERE ...];
```

→ avec col_i' de type compatible avec col_i

Application 1

Soient les relations :

Emp(num, nom, age, ville)

Archive(numA, nomA, dateEmbauche, dateFinEmbauche)

Requête 1 : Insérer le tuple « 01, "Ali Mabrouk", 20, "Tunis" » dans la relation Emp.

Requête 2 : Insérer dans la relation Emp un employé dont son numéro est égal à 2, son prénom est 'Mohamed' et qui habite à 'Tunis'.

Requête 3 : Insérer dans la relation Archive les numéros et les prénoms des employés de Emp ayant plus de 60 ans.

Application 1

Soient les relations :

Emp(num, nom, age, ville)

Archive(numA, nomA, dateEmbauche, dateFinEmbauche)

Requête 1 : Insérer le tuple « 01, "Ali Mabrouk", "Tunis", 20 » dans la relation Emp.

```
INSERT INTO Emp (num, nom, ville, age)
```

```
VALUES (01, 'Ali Mabrouk', 'Tunis', 20) ;
```

ou bien

```
INSERT INTO Emp
```

```
VALUES (01, 'Ali Mabrouk', 20, 'Tunis');
```

Application 1

Soient les relations :

Emp(num, nom, age, ville)

Archive(numA, nomA, dateEmbauche, dateFinEmbauche)

Requête 2 : Insérer dans la relation Emp un employé dont son numéro est égal à 2, son nom est 'Mohamed Ben Salah' et qui habite à 'Tunis'.

```
INSERT INTO Emp (num, nom, ville)
```

```
VALUES (2, 'Mohamed Ben Salah', 'Tunis') ;
```

Application 1

Soient les relations :

Emp(num, nom, age, ville)

Archive(numA, nomA, dateEmbauche, dateFinEmbauche)

Requête 3 : Insérer dans la relation Archive les numéros et les noms des employés de Emp ayant plus de 60 ans.

```
INSERT INTO Archive(numA, nomA)
SELECT num, nom
FROM Emp
WHERE age > 60 ;
```


Application 2

Soient les relations :

Commande(numCo, dateCo, montantCo)

ChiffreAffaire(anneeCA, moisCA, montantCA)

On considère le montant du chiffre d'affaire (CA) d'un mois comme étant la somme des montants des commandes ayant eu lieu durant ce mois en question.

Insérer le chiffre d'affaire du mois d'Avril de l'année 2020. On vous indique que **extract(year from d)** renvoie l'année d'une date d et **extract(month from d)** renvoie le mois en question.

Application 2

Soient les relations :

Commande(numCo, dateCo, montantCo)

ChiffreAffaire(anneeCA, moisCA, montantCA)

On considère le montant du chiffre d'affaire (CA) d'un mois comme étant la somme des montants des commandes ayant eu lieu durant ce mois en question.

Insérer le chiffre d'affaire du mois d'Avril de l'année 2020. On vous indique que `extract(year from d)` renvoie l'année d'une date d et `extract(month from d)` renvoie le mois en question. **(version 1)**

```
INSERT INTO ChiffreAffaire  
SELECT 2020, 04, SUM(montantCo)  
FROM Commande  
WHERE extract(year from dateCo) = 2020 AND extract(month from dateCo) = 4);
```

Application 2

Soient les relations :

Commande(numCo, dateCo, montantCo)

ChiffreAffaire(anneeCA, moisCA, montantCA)

On considère le montant du chiffre d'affaire (CA) d'un mois comme étant la somme des montants des commandes ayant eu lieu durant ce mois en question.

Insérer le chiffre d'affaire du mois d'Avril de l'année 2020. On vous indique que *extract(year from d)* renvoie l'année d'une date d et *extract(month from d)* renvoie le mois en question. **(version 2 : acceptée par Oracle, MySQL, etc.)**

INSERT INTO ChiffreAffaire

VALUES(2020, 04, (SELECT SUM(montantCo) FROM Commande WHERE extract(year from dateCo) = 2020 AND extract(month from dateCo) = 4));

Modification des données : UPDATE

```
UPDATE nom_table  
SET  
  nom_col_1 = {expression_1 | (SELECT ...)},  
  nom_col_2 = {expression_2 | (SELECT ...)},  
  ...  
  nom_col_n = {expression_n | (SELECT ...)}  
[WHERE predicat];
```

- La modification peut concerner une seule colonne comme elle peut être associée à plusieurs colonnes.
- *expression_i* définit une valeur pour la colonne i. Cette valeur peut résulter **d'une expression (variable, constante, combinaison) ou d'un ordre SELECT renvoyant une seule valeur.**
- Les valeurs des colonnes *nom_col_1*, *nom_col_2*, ..., *nom_col_n* sont modifiées dans toutes les lignes qui **satisfont le prédicat « predicat » s'il est présent** ou **toutes lignes de la table sinon.**
- **Le prédicat s'il existe peut contenir des sous-requêtes.**

Application

Soient les relations :

Emp(num, nom, age, ville, salaire, numDept)

Dept(numDept, nomDept)

Requête 1 : Faire passer "Xavier Dupont" dans le département 10 et enlever son âge.

Requête 2 : Augmenter le salaire de 40 des employés du département 20.

Requête 3 : Augmenter le salaire de 100 des employés du département 'Informatique'.

Requête 4 : Modifier le numéro du département des employés des départements 10 et 15 par le numéro du département le plus élevé de la relation Dept.

Application

Soient les relations :

Emp(num, nom, age, ville, salaire, numDept)

Dept(numDept, nomDept)

Requête 1 : Faire passer "Xavier Dupont" dans le département 10 et enlever son âge.

```
UPDATE Emp  
SET numDept=10,  
age = NULL  
WHERE nom = "Xavier Dupont" ;
```

Application

Soient les relations :

Emp(num, nom, age, ville, salaire, numDept)

Dept(numDept, nomDept)

Requête 2 : Augmenter le salaire de 40 des employés du département 20.

```
UPDATE Emp
SET salaire=salaire+40
WHERE numDept=20 ;
```

Application

Soient les relations :

Emp(num, nom, age, ville, salaire, numDept)

Dept(numDept, nomDept)

Requête 3 : Augmenter le salaire de 100 des employés du département 'Informatique'.

UPDATE Emp

SET salaire=salaire+100

WHERE numDept= (SELECT numDept FROM Dept WHERE nomDept = 'Informatique') ;

Application

Soient les relations :

Emp(num, nom, age, ville, salaire, numDept)

Dept(numDept, nomDept)

Requête 4 : Modifier le numéro du département des employés des départements 10 et 15 par le numéro du département le plus élevé de la relation Dept.

UPDATE Emp

SET numDept = (SELECT MAX(numDept) FROM Dept)

WHERE numDept = 10 OR numDept = 15;

Suppression des données : DELETE

DELETE FROM nomTable
[**WHERE** condition];

- ▶ La condition du prédicat WHERE, **si elle existe**, permet de restreindre les lignes de la table à supprimer,
- ▶ **Si la clause WHERE est absente**, toutes les lignes sont supprimées (en conséquent, la relation existe toujours mais sa population est vide).
- ▶ *La condition du prédicat, s'il existe, peut contenir des sous-requêtes.*

Application 1

Soit la relation :

Emp(num, nom, age, ville, salaire, numDept)

Requête 1 : Supprimer les employés qui travaillent au département 10

```
DELETE FROM Emp  
WHERE numDept=10 ;
```

Application 1

Soit la relation :

Emp(num, nom, age, ville, salaire, numDept)

Requête 2 : Supprimer les employés dont le rapport entre leur salaire et leur age dépasse 40.

```
DELETE FROM Emp  
WHERE salaire/age > 40 ;
```

Application 2

Soit la relation :

Emp(num, nom, age, ville, salaire, numDept)

Dept(numDept, nomDept, ageMax)

Requête : Supprimer les employés qui travaillent dans le département 'RH' ainsi que ceux ayant dépassé l'âge maximal permis dans le département qui leur est associé.

```
DELETE FROM Emp  
WHERE numDept = (SELECT numDept FROM Dept WHERE nomDept = 'RH')  
OR age > (SELECT ageMax FROM Dept WHERE Dept.numDept =  
Emp.numDept);
```