



# Database System

Lecture 12

SQL Sub Languages

**DML Data Manipulation Language**

*DML commands*

*INSERT Statement*

*UPDATE Statement*

*DELETE Statement*

# SQL Components Or SQL Sub Languages

**DCL:** Data Control Language

Example: Grant, Revoke.

**DDL:** Data Definition Language.

Example: Create, Alter, Drop, Rename and Truncate.

**DML:** Data Manipulation Language

Example: Insert, Update, Delete

**DRL:** Data Retrieval Language

Example: Select

**TCL:** Transaction Control Language

Example : Rollback, Commit, Savepoint

# *DML-Data Manipulation Language*

## *DML Data Manipulation Language*

### *DML commands:*

- *INSERT*
- *UPDATE*
- *DELETE*

# *DML- INSERT Statement*

## *Inserting values into a table*

*The syntax:*

```
INSERT INTO <table_name> VALUES  
(value1,value2,.....);
```

***Suppose*** : you have create employees table with four columns as below:

(Empid , Empname, EmpJob,EmpSal)

```
CREATE TABLE Employees
```

```
(  
Empid        NUMBER(10) NOT NULL,  
Empname     VARCHAR2(20) NOT NULL,  
EmpJob      VARCHAR2(20),  
EmpSal      NUMBER(7,2)
```

```
);
```

# *DML- INSERT Statement*

## **Examples :**

```
INSERT INTO Employees
```

```
VALUES (101,'Dhafer','manager',7866.67);
```

```
INSERT INTO Employees
```

```
VALUES (102,'Mohammed','Supervisor',6000);
```

```
INSERT INTO Employees
```

```
VALUES (103,'Ahmed','Teamleader ',4000.43);
```

## **HW:**

Insert 5 to 10 records to Employees table.

# ***DML- INSERT Statement***

## ***Displaying the data of the table***

***The syntax:***

**To display specific column**

```
SELECT [col1,col2,col3,...col_N]  
FROM <table_name>;
```

**To display all the column and rows**

```
SELECT * FROM <table_name>;
```

***Examples:***

```
SELECT Empid ,Empname FROM Employees;
```

```
SELECT * FROM employees;
```

# ***DML- INSERT Statement***

## ***More Examples for insert statement :***

**Suppose** you have create Dept table with three columns  
(DeptNo , DeptName, DeptPos)

```
CREATE TABLE Dept  
(  
DeptNo          NUMBER(3)    NOT NULL,  
DeptName       VARCHAR2(30) NOT NULL,  
DeptPos        VARCHAR2(30)  
);
```

# ***DML- INSERT Statement***

## ***Inserting data into Dept table***

```
INSERT INTO dept  
VALUES(10,'Administration','Iraq');  
INSERT INTO dept  
VALUES(20,'Accounting','Chicago');  
INSERT INTO dept  
VALUES(30,'HR','Dallas');  
INSERT INTO dept  
VALUES(40,'Sales','London');
```



# ***DML- INSERT Statement***

## ***More Examples for insert statement :***

**Suppose** you have create employees2 table with below columns

(Empid , Empname, EmpJob, EmpSal, EmpBouns, EmpMangid, EmpDoB, EmpDept)

```
CREATE TABLE Employees2  
( Empid          NUMBER(6)    NOT NULL,  
Empname         VARCHAR2(10) NOT NULL,  
EmpJob          VARCHAR2(10),  
EmpSal          NUMBER(7,2),  
EmpBouns        NUMBER(7,2),  
EmpMangid       NUMBER(6) ,  
EmpDoB          Date,  
EmpDeptNo       NUMBER(3)    NOT NULL  
);
```

# ***DML- INSERT Statement***

***Inserting data into employees2 table***

```
INSERT INTO employees2 VALUES  
(1235,'Dhafer','manager',1500,500,  
NULL,'27-JUN-80',10);
```

```
INSERT INTO employees2 VALUES  
(1234,'Manhel','salesman',800,150,1235,'1  
6-MAY-80',20);
```

```
INSERT INTO employees2 VALUES  
(1236,'Layth','analyst',11000,350,1239,'12-  
DEC-97',30);
```

# *DML- INSERT Statement*

*Insert values into a Table dynamically*

*OR*

*Inserting values into a table using '&'*

*The syntax:*

```
INSERT INTO <table_name>  
VALUES (&col1,&col2,.....) ;
```

# ***DML- INSERT Statement***

## ***Example:***

**Suppose** you have create employees table with four columns as below:

(Empid , Empname, EmpJob,EmpSal)

```
INSERT INTO employees  
VALUES(&Empid,'&Empname','&Empjob',&  
Empsal);
```

**Note:** Press / after entering a row to add more no. of rows.

# ***DML- INSERT Statement***

## ***HW- Exercise:***

1. Create a table student to store and retrieve the following columns.

(Std\_name,Std\_course,Std\_fees,Std\_address,Std\_phone\_NO)

2. Create a Bankacc table to store and retrieve the following columns.

(C\_Acc\_No, C\_Name,C\_Acc\_Type ,C\_Acc\_Balance)

# ***DML- INSERT Statement***

***Inserting values into specific column of a table***

***The Syntax:***

```
INSERT INTO <table_name>  
(col1,col2,col3,...)  
VALUES  
(val1,val2,val3,...) ;
```

# ***DML- INSERT Statement***

## **Example :**

**Suppose** you have create employees table with four columns as below:

**(Empid , Empname, EmpJob,EmpSal)**

```
INSERT INTO employees(Empid,Empname) VALUES  
(106,'Jasim');
```

```
INSERT INTO employees (Empid,Empname, Empjob)  
VALUES (107, 'ali', 'clerk');
```

```
INSERT INTO employees (Empid,Empname, EmpSal)  
VALUES (107,'Katrain', 3500)
```

**Note:** Values in other columns are treated as NULL value means missing information or unknown information or inapplicable information.

# ***DML- INSERT Statement***

*Inserting values into specific column dynamically.*

*The Syntax:*

**INSERT INTO <table\_name>**

**(col1,col2,....)**

**VALUES**

**(&col1,'&col2',....);**

**Example :**

**Suppose** you have create employees table with four columns as below:

(Empid , Empname, EmpJob,EmpSal)

**INSERT INTO employees (Empid,Empname)**

**VALUES (&empid,'&empname');**



# ***DML- UPDATE Statement***

## ***UPDATE Statement:***

Used to modify the existing data of a table.

### ***The syntax:***

```
UPDATE <table_name>  
SET col1=value ,col2=value,.....  
WHERE <condition>;
```

# ***DML- UPDATE Statement***

## ***Example :***

**Suppose** you have create employees table with four columns

**(Empid , Empname, EmpJob,EmpSal)**

```
UPDATE employees  
SET EmpSal =8000  
WHERE empid=101;
```

## ***Important Note :***

update statement without where clause , will modify all rows of the table.

# ***DML- DELETE Statement***

## ***DELETE Statement :***

Used to delete all the records or selected records using where clause.

### ***The syntax:***

```
DELETE  
FROM <table_name>  
WHERE <condition>;
```

# ***DML- DELETE Statement***

## ***Example:***

**Suppose** you create table employees with four columns

(Empid , Empname, EmpJob,EmpSal)

DELETE FROM employees where Empid =101;

DELETE FROM employees;

## ***Important Note:***

- DELETE statement without WHERE clause , will delete all rows of the table.
- WHERE condition is not exist in DDL command (TRUNCATE) unlike the in DML command (DELETE) that use Where Condition

# ***DML- DELETE Statement***

## **Differences between (TRUNCATE) and (DELETE):**

**DELETE:**

**VS**

**TRUNCATE:**

-----

-----

1.DML

1.DDL

2.Can delete selected records.

2.All records are deleted.

3.Getback records (rollback)

3.deleted permanently.



A close-up photograph of a right hand holding a silver pen, writing the words "Thank you" in a fluid, cursive script on a white surface. The pen is positioned at the end of the word "you".

Thank you

*Dhafar Sabah Yaseen*