Bahria University
Discovering Knowledge

CSL – 220: Database Management System
SEMESTER BS CS 04, BSIT 04

Lab 08: DML Queries

Objective(s):

To learn the Data Manipulation Language

DML

Data Manipulation Language (DML) besides of the **SELECT** statement that retrieves information from databases includes also statements modifying data state. These statements are:

INSERT	Inserts rows into database table
UPDATE	Changes values in columns of database table
DELETE	Deletes rows from database table

INSERT

The **insert** statement adds new rows to a table. In so doing, the column values may be literal constants or be formed from result of subquery execution. In the first case, one insert statement adds only one row; in the second case, one insert statement adds as many rows as the subquery returns.

Syntax

```
INSERT [INTO] <table> [(column_list)] VALUES (data_values)
```

Example 1

Inserting a single row of data

```
INSERT INTO Production.UnitMeasure
VALUES (N'FT', N'Feet', '20080414');
```

Example 2

Inserting multiple rows of data

```
INSERT INTO Production.UnitMeasure
VALUES (N'FT2', N'Square Feet ', '20080923'), (N'Y', N'Yards', '20080923'), (N'Y3', N'Cubic Yards', '20080923');
```

Example 3

Inserting data that is not in the same order as the table columns

```
INSERT INTO Production.UnitMeasure (Name, UnitMeasureCode,ModifiedDate)
VALUES (N'Square Yards', N'Y2', GETDATE());
```

UPDATE

The update statement changes existing data in a table. **Syntax**

```
UPDATE <table name>
SET <column> = <value> [,<column> = <value>]
[FROM <source table(s)>]
[WHERE <restrictive condition>]
```

With a single **update** statement, arbitrary number of columns can change their values. However a column cannot be changed in the same **update** statement more than once. All rows of a table will be affected if a **WHERE** clause is eliminated from an **update** statement.

If the column is defined to allow NULL values, NULL can be specified explicitly. Moreover, existing value can be changed by default value (**DEFAULT**) for a given column.

An *expression* may refer to current values in a table to be changed. For example, we may decrease prices by 10 percent on all the laptops with the following statement:

Example 4

```
UPDATE Production.UnitMeasure
SET UnitMeasureCode = ‘NY3
WHERE stor_id = ‘TEST’
```

DELETE statement

The delete statement delete rows from temporary or basic tables, views and cursors, with the statement action in the two last cases is propagated on those basic tables, from which were extracted data into these views and cursors.

Syntax:

```
DELETE FROM <table name> [WHERE <predicate>];
```

Example 5

The following example deletes all rows from the ProductCostHistory table in which the value in the StandardCost column is more than 1000.00

```
DELETE FROM Production.ProductCostHistory
WHERE StandardCost> 1000.00;
```

Exercise

Using the EMP and DEPT table, apply the following queries:

1. Create a new table EMPLOYEE replica of EMP table with no records in it.

   Create a new table DEPART replica of DEPT table with all the records in it.

   Create a primary foreignkey relationship between EMPLOYEE and DEPART tables.

2. Insert following records into EMPLOYEE using following values:

   EMPNO = 101, ENAME = WASEEEM, SAL = 5000, JOB = CLERK and DEPTNO = 20.

   EMPNO = 102, ENAME = SAJID, SAL = 15000, JOB = ANALYST and DEPTNO = 10.

3. Change the record inserted in question 2 from ENAME ‘WASEEM’ to ‘KHALID’ and SAL 5000 to 10000.

4. Remove all the records from EMPLOYEE table having job = ‘ANALYST’.

5. Try to update the table DEPART and change the value of DEPTNO from 10 to 30. Find the error and give the reason for the error.

6. Insert a new record into EMPLOYEE using following values:

   EMPNO = 103, ENAME = SALEEM, SAL = 60000, JOB = ANALYST and DEPTNO = 60. Find the error and give the reason for the error.