# DBMS SQL

## Lesson 10: Data Manipulation Language

Capgemini

# Lesson Objectives

- To understand the following topics:
  - Concept of Data Manipulation Language
  - Inserting rows into a table
  - Deleting rows from a table
  - Updating rows in a table

# Data Manipulation Language

- Data Manipulation Language (DML) is used to perform the following routines on database information:
  - Retrieve
  - Insert
  - Modify

- DML changes data in an object. If you insert a row into a table, that is DML.

- All DML statements change data, and must be committed before the change becomes permanent.

# INSERT

- INSERT command:
  - INSERT is a DML command. It is used to add rows to a table.
  - In the simplest form of the command, the values for different columns in the row to be inserted have to be specified.
  - Alternatively, the rows can be generated from some other tables by using a SQL query language command.

# Inserting Rows into a Table

- Inserting by specifying values:

Example: To insert a new record in the DEPT table

```
INSERT INTO table_name[(col_name1,col_name2,...)]
        {VALUES (value1,value2,....) | query};
```

```
INSERT INTO Department_master
VALUES (10, 'Computer Science');
```

# Inserting Rows into a Table

- Inserting rows in a table from another table using Subquery:

Example: The example given below assumes that a new_emp_table exists. You can use a subquery to insert rows from another table.

```
INSERT INTO new_staff_table
SELECT * FROM staff_master
WHERE staff_master.hiredate > '01-jan-82';
```

# Inserting Rows into a Table

- Inserting by using “substitution variables”:

Example: In the example given below, when the command is run, values are prompted every time.

```
INSERT INTO department_master
VALUES (&dept_code, '&dept_name');
Enter a value for dept_code : 20
Enter a value for dept_name : Electricals
```

# DELETE

- The DELETE command is used to delete one or more rows from a table.
  - The DELETE command removes all rows identified by the WHERE clause.

```
DELETE [FROM] {table_name | alias }
        [WHERE condition];
```

# Deleting Rows from Table

Example 1: If the WHERE clause is omitted, all rows will be deleted

from the table.

Example 2: If we want to delete all information about department 10 from the Emp table:

```
DELETE FROM staff_master;
```

```
DELETE FROM student_master WHERE  dept_code=10;
```

# UPDATE

- Use the UPDATE command to change single rows, groups of rows, or all rows in a table.
  - In all data modification statements, you can change the data in only "one table at a time".

```
UPDATE table_name
SET   col_name = value|
      col_name =
SELECT_statement_returning_single_value|
      (col_name,...) = SELECT_statement
[WHERE condition];
```

# Updating Rows from Table

Example 1: To UPDATE the column “dname” of a row, where deptno is 10, give the following command:

```
UPDATE department_master
SET dept_name= ‘Information Technology’
WHERE dept_code=10;
```

# Updating Rows from Table

Example 2: To UPDATE the subject marks details of a particular student, give the following command:

```
UPDATE student_marks
SET subject1= 80 , subject2= 70
            WHERE student_code=1005;
```

# Using a Subquery to do an Update

- For making salary of “Anil” equal to that of staff member 100006, use the following command:

```
UPDATE staff_master
SET staff_sal = (SELECT staff_sal FROM staff_master
                        WHERE staff_code = 100006 )
WHERE staff_name = ‘Anil';
```

# MERGE statement

- The MERGE statement, provides the ability to conditionally update or insert data into a database table.

- The MERGE statement, performs an UPDATE if the row exists, and an INSERT if it is a new row:
  - Increases performance and ease of use
  - Is useful in data warehousing applications
  - Avoids separate updates

# MERGE statement

- You can conditionally insert or update rows in a table by using the MERGE statement

```
MERGE INTO table_name table_alias
  USING (table|view|sub_query) alias
  ON (join condition)   WHEN MATCHED THEN
    UPDATE SET
    col1 = col_val1,
    col2 = col2_val
  WHEN NOT MATCHED THEN
    INSERT (column_list)
    VALUES (column_values);
```

# Example on Merge

Example

```
CREATE table staff_copy as select staff_code,staff_name FROM
staff_master where 1=2;

MERGE into staff_copy using staff_master
    ON (staff_master.deptno=staff_copy.deptno)
    WHEN MATCHED THEN
    UPDATE SET staff_code=staff_master.staff_code,
    staff_name=staff_master.staff_name
    WHEN NOT MATCHED THEN
   INSERT (staff_code,staff_name) values
  (staff_master.staff_code,staff_master.staff_name);
```

# Summary

- The concept of Data Manipulation Language
- Inserting rows into a table
- Deleting rows from a table
- Updating rows in a table
- Using Merge Statement

# Review - Questions

- Question 1: Both TRUNCATE statement and DELETE without condition removes the entire date from a table
  - True/False
- Question 2: All DML statements are auto committed
  - True/False
- Question 3:In a transaction, DDL statement after DML statement commits the changes done by DML.
  - True/False

# Review - Questions

- Question 4: Inserting rows in a table emp1 from another table can be done using ____.
  - Option 1: insert into emp1(t1) as select empno from emp
  - Option 2: insert into emp1(t1)  select empno from emp
  - Option 3: insert into emp1(t1) as select * from emp