

**Government College of Engineering, Jalgaon**  
**(An Autonomous Institute of Government of Maharashtra)**

<b>Name :</b> Janhavi Ramesh Wankhede	<b>Semester :</b> V	<b>PRN :</b> 2141027
<b>Class :</b> T. Y. B.Tech Computer	<b>Academic Year :</b> 2023-24	<b>Subject :</b> CO307U
<b>Course Teacher : Mr. Vinit Kakde</b>		
<b>Date of Performance :</b>	<b>Date of Completion :</b>	

**Practical no. 2**

**Aim :** Design at least 10 SQL queries for suitable database application using SQL DML statements to retrieve, insert, delete and update data and queries which involves DML Features like Set Operation, Set Comparisons, Aggregation, group by, having, order by, queries to demonstrate Transaction control language(TCL): commit, savepoint, Rollback and Data Control Language (DCL): Grant/Revoke.

**Requirements:**

1. Computer System with Open Source Operating System.
2. Mysql

**Theory:**

**Data Manipulation Language (DML) Statements :**

Data manipulation language (DML) statements access and manipulate data in existing schema objects. These statements do not implicitly commit the current transaction. The SELECT statement is a limited form of DML statement in that it can only access data in the database. It cannot manipulate data in the database, although it can operate on the accessed data before returning the results of the query. The data manipulation language statements are: INSERT, UPDATE, DELETE, SELECT

The SELECT statement is a limited form of DML statement in that it can only access data in the database. It cannot manipulate data in the database, although it can operate on the accessed data before returning the results of the query.

The SQL INSERT INTO Statement is used to add new rows of data to a table in the database.

**Syntax:** There are two basic syntax of INSERT INTO statement as follows:

INSERT INTO TABLE\_NAME (column1, column2, column3,... columnN)] VALUES (value1, value2, value3,...valueN);

Here, column1, column2,...columnN are the names of the columns in the table into which want to insert data.

It's not required to specify the column(s) name in the SQL query if values for all the columns of the table are provided. But make sure the order of the values is in the same order as the columns in the table.

The SQL INSERT INTO syntax would be as follows:

```
INSERT INTO TABLE_NAME VALUES (value1,value2,value3,...valueN);
```

Following statements would create records in CUSTOMERS table:

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY)VALUES (1, "Ramesh", 32,  
"Ahmedabad", 2000.00 );
```

```
INSERT INTO CUSTOMERS VALUES (7, "Muffy", 24, "Indore",10000.00 );
```

#### **Populate one table using another table:**

You can populate data into a table through select statement over another table provided another table has a set of fields, which are required to populate first table. Here is the syntax:

```
INSERT INTO first_table_name [(column1, column2, ...columnN)]  
SELECT column1, column2, ...columnN FROM second_table_name [WHERE condition];
```

SQL SELECT statement is used to fetch the data from a database table which returns data in the form of result table. These result tables are called result-sets.

**Syntax:** The basic syntax of SELECT statement is as follows:

```
SELECT column1, column2, columnN FROM table_name;
```

Here, column1, column2...are the fields of a table whose values you want to fetch. If you want to fetch all the fields available in the field, then you can use the following syntax:

```
SELECT * FROM table_name;  
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS;
```

The SQL WHERE clause is used to specify a condition while fetching the data from single table or joining with multiple tables. If the given condition is satisfied then only it returns specific value from the table.

You would use WHERE clause to filter the records and fetching only necessary records. The WHERE is also used in UPDATE, DELETE statement

**Syntax:** The basic syntax of SELECT statement with WHERE clause is as follows:

```
SELECT column1, column2, columnN FROM table_name WHERE  
[condition]
```

You can specify a condition using comparison or logical operators like >, <, =, LIKE, NOT, etc.

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE  
SALARY > 2000;  
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE  
NAME = „Hardik“;
```

The SQL AND and OR operators are used to combine multiple conditions to narrow data in an SQL statement. These two operators are called conjunctive operators. These operators provide a means to make multiple comparisons with different operators in the same SQL statement.

The AND Operator: The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.

**Syntax:** The basic syntax of AND operator with WHERE clause is as follows:

```
SELECT column1, column2, columnN FROM table_name WHERE [condition1] AND  
[condition2]...AND [conditionN];
```

You can combine N number of conditions using AND operator. For an action to be taken by the SQL statement, whether it be a transaction or query, all conditions separated by the AND must be TRUE.

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000 AND age < 25;
```

The OR Operator: The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.

**Syntax:** The basic syntax of OR operator with WHERE clause is as follows:

```
SELECT column1, column2, columnN FROM table_name WHERE[condition1] OR  
[condition2]...OR [conditionN]
```

You can combine N number of conditions using OR operator. For an action to be taken by the SQL statement, whether it be a transaction or query, only any ONE of the conditions separated by the OR must be TRUE.

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000 OR age < 25;
```

The SQL UPDATE Query is used to modify the existing records in a table. You can use WHERE clause with UPDATE query to update selected rows otherwise all the rows would be affected.

**Syntax:** The basic syntax of UPDATE query with WHERE clause is as follows:

```
UPDATE table_name SET column1 = value1, column2 = value2...,columnN = valueN WHERE  
[condition];
```

You can combine N number of conditions using AND or OR operators.

```
SQL> UPDATE CUSTOMERS SET ADDRESS = "Pune" WHERE ID = 6;
```

```
SQL> UPDATE CUSTOMERS SET ADDRESS = "Pune", SALARY =1000.00;
```

The SQL DELETE Query is used to delete the existing records from a table. You can use WHERE clause with DELETE query to delete selected rows, otherwise all the records would be deleted.

**Syntax:** The basic syntax of DELETE query with WHERE clause is as follows:

```
DELETE FROM table_name WHERE [condition];
```

You can combine N number of conditions using AND or OR operators.

```
SQL> DELETE FROM CUSTOMERS WHERE ID = 6;
```

If you want to DELETE all the records from CUSTOMERS table, you do not need to use WHERE clause and DELETE query would be as follows:

```
SQL> DELETE FROM CUSTOMERS;
```

The SQL DISTINCT keyword is used in conjunction with SELECT statement to eliminate all the duplicate records and fetching only unique records. There may be a situation when you have multiple duplicate records in a table. While fetching such records, it makes more sense to fetch only unique records instead of fetching duplicate records.

**Syntax:** The basic syntax of DISTINCT keyword to eliminate duplicate records is as follows:

```
SELECT DISTINCT column1, column2,.....columnN FROM table_name WHERE [condition]
SELECT DISTINCT SALARY FROM CUSTOMERS ORDER BY SALARY;
```

The SQL GROUP BY clause is used in collaboration with the SELECT statement to arrange identical data into groups. The GROUP BY clause follows the WHERE clause in a SELECT statement and precedes the ORDER BY clause.

**Syntax:** The basic syntax of GROUP BY clause is given below. The GROUP BY clause must follow the conditions in the WHERE clause and must precede the ORDER BY clause if one is used.

```
SELECT column1, column2 FROM table_name WHERE [ conditions ] GROUP BY column1,
column2
```

```
SQL> SELECT NAME, SUM(SALARY) FROM CUSTOMERS GROUP BY NAME;
```

Following is an example, which would sort the result in descending order by NAME:

```
SQL> SELECT * FROM CUSTOMERS ORDER BY NAME DESC;
```

SQL has many built-in functions for performing processing on string or numeric data. Following is the list of all useful SQL built-in functions:

**SQL SUM Function** – The SQL SUM aggregate function allows selecting the total for a numeric column.

**SQL COUNT Function** – The SQL COUNT aggregate function is used to count the number of rows in a database table.

**Syntax :** SELECT COUNT(column\_name) FROM table\_name WHERE CONDITION;

SQL MAX Function – The SQL MAX aggregate function allows us to select the highest (maximum) value for a certain column.

SQL MIN Function – The SQL MIN aggregate function allows us to select the lowest (minimum) value for a certain column.

SQL AVG Function – The SQL AVG aggregate function selects the average value for certain table column.

### **Aggregate Functions:**

Aggregate functions are functions that take a collection (a set or multiset) of values as input and return a single value. SQL offers five built-in aggregate functions:

- Average: avg
- Minimum: min
- Maximum: max
- Total: sum
- Count: count

The input to sum and avg must be a collection of numbers, but the other operators can operate on collections of nonnumeric data types, such as strings, as well.

### **Basic Aggregation**

Consider the query “Find the average salary of employees in the Computer Science department.”

We write this query as follows:

```
select avg (salary) from employee  
where dept name= “Comp. Sci.”;
```

The result of this query is a relation with a single attribute, containing a single tuple with a numerical value corresponding to the average salary of employees in the Computer Science department.

We use the aggregate function count frequently to count the number of tuples in a relation. The notation for this function in SQL is count(\*). Thus, to find the number of tuples in the employee relation, we write

```
select count (*)  
from employee;
```

There are circumstances where we would like to apply the aggregate function not only to a single set of tuples, but also to a group of sets of tuples; we specify this wish in SQL using the group by clause. The attribute or attributes given in the group by clause are used to form groups. Tuples with the same value on all attributes in the group by clause are placed in one group.

As an illustration, consider the query “Find the average salary in each department.”

We write this query as follows:

```
select dept name, avg (salary) as avg salary from employee group by dept name;
```

### **Set Operations:**

The SQL operations union, intersect, and except operate on relations and correspond to the mathematical set-theory operations  $\cup$ ,  $\cap$ , and  $-$ .

#### **The Union Operation**

To find the set of all employees who belong to either “IT” dept. or “Comp” dept, or both, (select \* from employee where dept\_name=“IT”) union(select \*from employee where dept\_name = “Comp”);

The union operation automatically eliminates duplicates, unlike the select clause.

On the similar concepts, intersect and minus/except operations works by retaining their individual properties.

### **Conclusion:**

Design at least 10 SQL queries for suitable database application using SQL DML statements to retrieve, insert, delete and update data and queries which involves DML Features

like Set Operation, Set Comparisons, Aggregation, group by, having, order by, queries to demonstrate Transaction control language (TCL): commit, savepoint, Rollback and Data Control Language (DCL): Grant/Revoke.

**Questions:**

**1) Explain the use of DDL language:**

DDL is an abbreviation of Data Definition Language. It is used to create and modify the structure of database objects in database.

**2) Explain the use of DML language:**

DML is an abbreviation of Data Manipulation Language. It is used to retrieve, store, modify, delete, insert and data in database.

**3) Explain the use of DCL language:**

DCL is an abbreviation of Data Control Language. It is used to create rules, permissions, and referential integrity as well it is used to control access to database by securing it.

**4) Explain the use of TCL language:**

TCL is an abbreviation of Transactional Control Language. It is used to manage different transactions occurring within a database.

**5) What is the difference between DROP TABLE and TRUNCATE commands of DDL language?**

DROP TABLE command deletes the whole table whereas TRUNCATE deletes only rows present in the table.

**Course Teacher**  
**Mr. Vinit Kakde**