

Assignment 7

TITLE

To perform:

- a) SQL queries using Search conditions
- b) SQL queries using Summary queries (Aggregate Functions)
- c) SQL queries using Sub-queries
- d) Subqueries with **IN** clause
- e) Subqueries with **EXISTS** clause

PROBLEM STATEMENT

Perform SQL operations to demonstrate the use of search conditions, aggregation and summary queries, and nested queries (subqueries) using **IN** and **EXISTS** clauses. This lab will help in understanding how subqueries can be used to filter, update, and manipulate data efficiently.

OBJECTIVES

- To learn how to apply **search conditions** in SQL queries.
- To perform **summary queries** using aggregate functions and grouping.
- To understand the concept of **subqueries** and their use inside SELECT, UPDATE, DELETE, and INSERT statements.
- To demonstrate the use of **IN** clause in subqueries.
- To demonstrate the use of **EXISTS** clause in subqueries.

PLATFORM REQUIRED

- **Operating System:** Windows or Linux
- **Software/Tools:** MySQL, Oracle, or any SQL environment (MySQL Workbench, XAMPP, Command Line)

THEORY

1. Search Conditions

Search conditions are used with SQL queries to filter rows based on logical expressions.

```
SELECT * FROM Student WHERE dept = 'CSE' AND marks > 60;
```

2. Summary Queries (Aggregate Functions)

Summary queries use aggregate functions such as SUM, AVG, MAX, MIN, and COUNT to summarize data.

```
SELECT dept, AVG(marks) FROM Student GROUP BY dept;
```

3. Subqueries

A **subquery** is a query inside another query.

- The **inner query** executes first.
- The **outer query** uses the result of the inner query.

General syntax:

```
SELECT column_name
```

```
FROM table_name
```

```
WHERE column_name operator (SELECT column_name FROM table_name WHERE condition);
```

4. Subquery with IN Clause

```
SELECT NAME, LOCATION, PHONE_NUMBER
```

```
FROM DATABASE
```

```
WHERE ROLL_NO IN (SELECT ROLL_NO FROM STUDENT WHERE SECTION='A');
```

5. Subquery with EXISTS Clause

```
SELECT NAME, ROLL_NO
```

```
FROM DATABASE D
```

```
WHERE EXISTS (SELECT * FROM STUDENT S WHERE D.ROLL_NO = S.ROLL_NO);
```

SCENARIO EXAMPLE

Consider two tables:

DATABASE Table

NAME ROLL_NO LOCATION PHONE_NUMBER

Ram	101	Chennai	9988775566
Raj	102	Coimbatore	8877665544
Sasi	103	Madurai	7766553344
Ravi	104	Salem	8989898989
Sumathi	105	Kanchipuram	8989856868

STUDENT Table

NAME ROLL_NO SECTION

Ravi	104	A
Sumathi	105	B
Raj	102	A

Examples:

1. Search Condition:

```
SELECT * FROM DATABASE WHERE LOCATION = 'Chennai';
```

2. Summary Query:

```
SELECT COUNT(*), MAX(ROLL_NO), MIN(ROLL_NO) FROM STUDENT;
```

3. Subquery with IN:

```
SELECT NAME, LOCATION FROM DATABASE
```

```
WHERE ROLL_NO IN (SELECT ROLL_NO FROM STUDENT WHERE SECTION='A');
```

4. Subquery with EXISTS:

```
SELECT NAME, ROLL_NO FROM DATABASE D
```

```
WHERE EXISTS (SELECT * FROM STUDENT S WHERE D.ROLL_NO = S.ROLL_NO);
```

STEP-BY-STEP ALGORITHM

1. Open your SQL environment and connect to the database server.
2. Create the required tables (DATABASE and STUDENT) and insert sample records.
3. Write queries with **search conditions** to filter records.
4. Write **summary queries** using aggregate functions.
5. Implement subqueries inside SELECT statements.
6. Use **IN** clause with subqueries to filter records from another table.
7. Use **EXISTS** clause to check existence of related records.
8. Verify outputs using `SELECT * FROM table_name;`

QUESTIONS FOR PRACTICE

1. Write a query to display all students from DATABASE whose roll numbers are in section B of STUDENT table.

2. Write a query to count the total number of students from the STUDENT table.
3. What is the difference between **IN** and **EXISTS** clauses? Give examples.
4. Write a query using a subquery to update a student's location in DATABASE table if they exist in STUDENT table.
5. Write a query to find the names and locations of students who belong to section A using subqueries.

CONCLUSION

In this lab, we learned how to use **search conditions** to filter data, **summary queries** to summarize records, and **subqueries** for nested query execution. We also practiced using **IN** and **EXISTS** clauses for efficient filtering. These techniques improve SQL query formulation and are widely used in real-world applications for reporting and data analysis.