Assignment 4

1. Write an SQL query to display the names and salaries of employees whose salary is greater than the average salary in the company. Explain how the subquery works.

```
mysql> SELECT name, salary
    -> FROM employees
    -> WHERE salary > (
    ->     SELECT AVG(salary)
    ->     FROM employees
    -> );
+--------+-----------+
| name   | salary    |
+--------+-----------+
| Ravi   |  90000.00 |
| Anjali | 120000.00 |
+--------+-----------+
2 rows in set (0.00 sec)

mysql>
```

2. Write a query to retrieve the top 5 highest-paid employees from an employees table. Explain how sorting affects the output.

```
mysql> SELECT *
    -> FROM employees
    -> ORDER BY salary DESC
    -> LIMIT 5;
+--------+--------+----------+------------+----------+------------+
| emp_id | name   | job_role | department | salary   | commission |
+--------+--------+----------+------------+----------+------------+
|      6 | Anjali | Manager  | IT         | 120000.00 |   10000.00 |
|      3 | Ravi   | Manager  | HR         |  90000.00 |       NULL |
|      2 | Neha   | Developer | IT        |  75000.00 |    5000.00 |
|      4 | Pooja  | Analyst  | Finance    |  65000.00 |    3000.00 |
|      1 | Amit   | Developer | IT        |  60000.00 |       NULL |
+--------+--------+----------+------------+----------+------------+
5 rows in set (0.00 sec)

mysql>
```

3. Write a query to calculate:
• Total number of employees
• Average salary
• Minimum and maximum salary
Explain the difference between aggregate and scalar functions.

```
mysql> SELECT
    ->     COUNT(*) AS total_employees,
    ->     AVG(salary) AS average_salary,
    ->     MIN(salary) AS minimum_salary,
    ->     MAX(salary) AS maximum_salary
    -> FROM employees;
+-----------------+----------------+----------------+----------------+
| total_employees | average_salary | minimum_salary | maximum_salary |
+-----------------+----------------+----------------+----------------+
|               6 |   75000.000000 |       40000.00 |      120000.00 |
+-----------------+----------------+----------------+----------------+
1 row in set (0.00 sec)

mysql>
```

Aggregate vs Scalar function

Aggregate functions operate on multiple rows and return one result.
Examples: COUNT, AVG, SUM, MIN, MAX

Scalar functions operate on a single row and return one result per row.
Examples: UPPER(name), ROUND(salary), LENGTH(name)

Aggregate functions reduce data.

Scalar functions transform data.

4. Given a sales table with columns (region, amount), write a query to find total sales per region. Filter only those regions where total sales exceed 50,000.

```
mysql> SELECT region, SUM(amount) AS total_sales
    -> FROM sales
    -> GROUP BY region
    -> HAVING SUM(amount) > 50000;
+--------+-------------+
| region | total_sales |
+--------+-------------+
| North  |    55000.00 |
| South  |    60000.00 |
| West   |    70000.00 |
+--------+-------------+
3 rows in set (0.00 sec)

mysql>
```

5. Write a query to find the number of unique job roles in an employees table. Explain why DISTINCT is necessary here.

```
mysql> SELECT COUNT(DISTINCT job_role) AS unique_job_roles
    -> FROM employees;
+------------------+
| unique_job_roles |
+------------------+
|                4 |
+------------------+
1 row in set (0.00 sec)

mysql>
```

6. Write a query to retrieve students who scored between 60 and 80 marks. Rewrite the same query using BETWEEN.

```
mysql> SELECT *
    -> FROM students
    -> WHERE marks >= 60 AND marks <= 80;
+------------+-------+-------+
| student_id | name  | marks |
+------------+-------+-------+
|          2 | Sita  |    72 |
|          3 | Mohan |    65 |
+------------+-------+-------+
2 rows in set (0.00 sec)

mysql> SELECT *
    -> FROM students
    -> WHERE marks BETWEEN 60 AND 80;
+------------+-------+-------+
| student_id | name  | marks |
+------------+-------+-------+
|          2 | Sita  |    72 |
|          3 | Mohan |    65 |
+------------+-------+-------+
2 rows in set (0.00 sec)

mysql>
```

7. Write a query to display employees whose commission is NULL. Explain the correct way to check NULL values in SQL.

```
mysql> SELECT *
    -> FROM employees
    -> WHERE commission IS NULL;
+--------+-------+-----------+------------+----------+------------+
| emp_id | name  | job_role  | department | salary   | commission |
+--------+-------+-----------+------------+----------+------------+
|      1 | Amit  | Developer | IT         | 60000.00 |       NULL |
|      3 | Ravi  | Manager   | HR         | 90000.00 |       NULL |
|      5 | Karan | Clerk     | HR         | 40000.00 |       NULL |
+--------+-------+-----------+------------+----------+------------+
3 rows in set (0.00 sec)

mysql>
```

8. Write a query to increase the salary of employees in the “IT” department by 10%. Explain how arithmetic operations are handled in SQL

```
mysql> UPDATE employees
    -> SET salary = salary * 1.10
    -> WHERE department = 'IT';
Query OK, 3 rows affected (0.02 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

9. Write a query to delete records of students who scored less than 40 marks. What precaution should be taken before executing DELETE?

```
mysql> DELETE FROM students
    -> WHERE marks < 40;
Query OK, 1 row affected (0.02 sec)

mysql> SELECT *
    -> FROM students
    -> WHERE marks < 40;
Empty set (0.00 sec)

mysql>
```

10.Write a query to find employees who earn more than the average salary of their department (without using joins). Explain the logic of the subquery.

```
mysql> SELECT *
    -> FROM employees e
    -> WHERE salary > (
    ->     SELECT AVG(salary)
    ->     FROM employees
    ->     WHERE department = e.department
    -> );
+--------+--------+----------+------------+-----------+------------+
| emp_id | name   | job_role | department | salary    | commission |
+--------+--------+----------+------------+-----------+------------+
|      3 | Ravi   | Manager  | HR         |  90000.00 |       NULL |
|      6 | Anjali | Manager  | IT         | 132000.00 |   10000.00 |
+--------+--------+----------+------------+-----------+------------+
2 rows in set (0.00 sec)

mysql>
```