

Third Year B. Tech (EL & CE)

Semester: V

Subject: Data Base Management System

Name: Shreerang Mhatre

Class: TY

Roll No: 52

Batch: A3

Experiment No: 06

Name of the Experiment: SQL Queries on Functions

Performed on: 14/10/2023

Submitted on: 30/10/2023

Problem Statement:

SQL Queries on Functions: Single row, Aggregate functions, Data sorting, Subquery.

Code:

```
MySQL 8.0 Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.34 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE Company;
-> CREATE DATABASE Company;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'CREATE DATABASE Company' at line 2
mysql> CREATE DATABASE Company;
ERROR 1007 (HY000): Can't create database 'company'; database exists
mysql> DROP DATABASE Company;
Query OK, 2 rows affected (0.03 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
6 rows in set (0.01 sec)

mysql> CREATE DATABASE Company;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| company |
| information_schema |
| mysql |
| performance_schema |
| sakila |
| sys |
| world |
+-----+
7 rows in set (0.00 sec)
```

```
MySQL 8.0 Command Line Client

mysql> CREATE DATABASE Company;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| company  |
| information_schema |
| mysql    |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
7 rows in set (0.00 sec)

mysql> USE Company;
Database changed

mysql> CREATE TABLE employees (
  ->   emp_id INT PRIMARY KEY,
  ->   first_name VARCHAR(50),
  ->   last_name VARCHAR(50),
  ->   salary DECIMAL(10, 2),
  ->   department_id INT
  -> );
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_company |
+-----+
| employees          |
+-----+
1 row in set (0.01 sec)

mysql> INSERT INTO employees (emp_id, first_name, last_name, salary, department_id)
  -> VALUES
  -> (1, 'John', 'Doe', 50000, 1),
  -> (2, 'Jane', 'Smith', 60000, 1),
  -> (3, 'Bob', 'Johnson', 75000, 2),
  -> (4, 'Alice', 'Williams', 80000, 2),
  -> (5, 'Charlie', 'Brown', 55000, 1);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> SHOW TABLES;
+-----+
| Tables_in_company |
+-----+
| employees          |
+-----+
1 row in set (0.00 sec)
```

```
MySQL 8.0 Command Line Client

  -> (4, 'Alice', 'Williams', 80000, 2),
  -> (5, 'Charlie', 'Brown', 55000, 1);
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> SHOW TABLES;
+-----+
| Tables_in_company |
+-----+
| employees          |
+-----+
1 row in set (0.00 sec)

mysql> DESC employees;
+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+
| emp_id | int | NO | PRI | NULL | |
| first_name | varchar(50) | YES | | NULL | |
| last_name | varchar(50) | YES | | NULL | |
| salary | decimal(10,2) | YES | | NULL | |
| department_id | int | YES | | NULL | |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT * FROM employees;
+-----+
| emp_id | first_name | last_name | salary | department_id |
+-----+
| 1 | John | Doe | 50000.00 | 1 |
| 2 | Jane | Smith | 60000.00 | 1 |
| 3 | Bob | Johnson | 75000.00 | 2 |
| 4 | Alice | Williams | 80000.00 | 2 |
| 5 | Charlie | Brown | 55000.00 | 1 |
+-----+
5 rows in set (0.00 sec)

mysql> SELECT UPPER(first_name || ' ' || last_name) AS full_name, salary
  -> FROM employees;
+-----+
| full_name | salary |
+-----+
| 0 | 50000.00 |
| 0 | 60000.00 |
| 0 | 75000.00 |
| 0 | 80000.00 |
| 0 | 55000.00 |
+-----+
5 rows in set, 12 warnings (0.00 sec)
```

```
MySQL 8.0 Command Line Client
+----+-----+-----+-----+-----+
| 2 | Jane | Smith | 60000.00 | 1 |
| 3 | Bob | Johnson | 75000.00 | 2 |
| 4 | Alice | Williams | 80000.00 | 2 |
| 5 | Charlie | Brown | 55000.00 | 1 |
+----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT UPPER(first_name || ' ' || last_name) AS full_name, salary
-> FROM employees;
+-----+-----+
| full_name | salary |
+-----+-----+
| 0 | 50000.00 |
| 0 | 60000.00 |
| 0 | 75000.00 |
| 0 | 80000.00 |
| 0 | 55000.00 |
+-----+-----+
5 rows in set, 12 warnings (0.00 sec)

mysql> SELECT AVG(salary) AS average_salary
-> FROM employees;
+-----+
| average_salary |
+-----+
| 64000.000000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUM(salary) AS total_salary_expenditure
-> FROM employees
-> WHERE department_id = 1;
+-----+
| total_salary_expenditure |
+-----+
| 165000.00 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT emp_id, first_name, last_name, salary
-> FROM employees
-> WHERE department_id = 2
-> ORDER BY salary DESC;
+-----+-----+-----+-----+
| emp_id | first_name | last_name | salary |
+-----+-----+-----+-----+
| 4 | Alice | Williams | 80000.00 |
| 3 | Bob | Johnson | 75000.00 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
MySQL 8.0 Command Line Client
+-----+-----+
| 0 | 50000.00 |
| 0 | 60000.00 |
| 0 | 75000.00 |
| 0 | 80000.00 |
| 0 | 55000.00 |
+-----+-----+
5 rows in set, 12 warnings (0.00 sec)

mysql> SELECT AVG(salary) AS average_salary
-> FROM employees;
+-----+
| average_salary |
+-----+
| 64000.000000 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT SUM(salary) AS total_salary_expenditure
-> FROM employees
-> WHERE department_id = 1;
+-----+
| total_salary_expenditure |
+-----+
| 165000.00 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT emp_id, first_name, last_name, salary
-> FROM employees
-> WHERE department_id = 2
-> ORDER BY salary DESC;
+-----+-----+-----+-----+
| emp_id | first_name | last_name | salary |
+-----+-----+-----+-----+
| 4 | Alice | Williams | 80000.00 |
| 3 | Bob | Johnson | 75000.00 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT emp_id, first_name, last_name, salary
-> FROM employees
-> WHERE department_id = 1
-> AND salary > (SELECT AVG(salary) FROM employees);
Empty set (0.00 sec)

mysql>
```

Exp 6

SQL Queries on Functions:
Single row
Aggregate functions
Data sorting
Subquery

⇒ // TABLE

```
CREATE TABLE employee(  
    emp_id INT PRIMARY KEY,  
    first_name VARCHAR(60),  
    last_name VARCHAR(60),  
    salary DECIMAL(10,2),  
    department_id INT  
);
```

// values

```
INSERT INTO employee(  
    emp_id, first_name, last_name,  
    salary, department_id)
```

VALUES

```
(1, 'John', 'Doe', 50000, 1);  
(2, 'Jane', 'Smith', 60000, 1);  
(3, 'Bob', 'Johnson', 75000, 2);  
(4, 'Alice', 'Williams', 80000, 2);  
(5, 'Charlie', 'Brown', 65000, 1);
```

Functions

// Single-Row Functions

```
SELECT UPPER (first_name || ' ' || last_name)
AS full_name, salary
FROM employees;
```

// Aggregate Functions

```
SELECT AVG (salary) AS average_salary
FROM employees;
```

// Data Sorting:

```
SELECT emp_id, first_name, last_name,
salary
FROM employees
WHERE department_id = 2
ORDER BY salary DESC;
```

// Subquery:

```
SELECT emp_id, first_name, last_name, salary
FROM employees
WHERE department_id = 1
AND salary > (SELECT AVG(salary)
FROM employees);
```