Course Name: DBMS Lab

Course Code: CSEG2146

Faculty Name: Dr. Priyanka Singh

Submitted By:

Anoushka Pandey

Batch: B1

Sap Id: 500120245

Roll no.: R2142230030

Experiment 6:

Title: Use of Inbuilt functions and relational algebra operation

Objective: To understand the use of inbuilt function and relational algebra with sql query.

1. **Create the following two tables (EMP and DEPT)**

EMP TABLE:

```
mysql> CREATE TABLE EMP(
    -> EMPNO INT PRIMARY KEY,
    -> ENAME VARCHAR(20) NOT NULL,
    -> JOB VARCHAR(30) NOT NULL,
    -> MGR INT,
    -> HIREDATE DATE,
    -> SAL INT NOT NULL,
    -> COMM INT,
    -> DEPTNO INT);
Query OK, 0 rows affected (0.34 sec)
```

```
mysql> SELECT * FROM EMP;
+-------+--------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-----------+------+------------+------+------+--------+
|  7369 | SMITH  | CLERK     | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7499 | ALLEN  | SALESMAN  | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD   | SALESMAN  | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7566 | JONES  | MANAGER   | 7698 | 1981-04-02 | 2975 | NULL |     20 |
|  7654 | MARTIN | SALESMAN  | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7844 | TURNER | SALESMAN  | 7698 | 1981-09-08 | 1500 |    0 |     30 |
|  7876 | ADAMS  | CLERK     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7934 | MILLAR | CLERK     | 7782 | 1982-12-23 | 1300 | NULL |     10 |
+-------+--------+-----------+------+------------+------+------+--------+
14 rows in set (0.02 sec)
```

DEPT TABLE:

```
mysql> CREATE TABLE DEPT(
    -> DEPTNO INT PRIMARY KEY,
    -> DNAME VARCHAR(30) NOT NULL,
    -> LOC VARCHAR(35));
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> SELECT * FROM DEPT;
+--------+------------+----------+
| DEPTNO | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
|     40 | OPERATIONS | BOSTON   |
+--------+------------+----------+
4 rows in set (0.00 sec)
```

2. Write the Queries for the following using In-built functions.

1. Retrieve average salary of all employees.

```
mysql> SELECT AVG(SAL) FROM EMP;
+-----------+
| AVG(SAL)  |
+-----------+
| 2051.7857 |
+-----------+
1 row in set (0.02 sec)
```

2. Retrieve the number of employees.

```
mysql> SELECT COUNT(*) FROM EMP;
+----------+
| COUNT(*) |
+----------+
|       14 |
+----------+
1 row in set (0.02 sec)
```

3. Retrieve distinct number of employee.

```
mysql> SELECT COUNT(DISTINCT ENAME) FROM EMP;
+----------------------+
| COUNT(DISTINCT ENAME) |
+----------------------+
|                   14 |
+----------------------+
1 row in set (0.02 sec)
```

4. Retrieve total salary of employee group by job.

```
mysql> SELECT JOB,SUM(SAL) FROM EMP GROUP BY JOB;
+-----------+----------+
| JOB       | SUM(SAL) |
+-----------+----------+
| CLERK     |     3850 |
| SALESMAN  |     5600 |
| MANAGER   |     8275 |
| ANALYST   |     6000 |
| PRESIDENT |     5000 |
+-----------+----------+
5 rows in set (0.01 sec)
```

5. Display the employee information with maximum salary.

```
mysql> SELECT * FROM EMP WHERE SAL=(SELECT MAX(SAL) FROM EMP WHERE SAL < (SELECT MAX(SAL) FROM EMP))
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
2 rows in set (0.02 sec)
```

6. Find the highest paid employee in department 10.

```
mysql> SELECT ENAME FROM EMP WHERE DEPTNO=10 ORDER BY SAL DESC LIMIT 1;
+-------+
| ENAME |
+-------+
| KING  |
+-------+
1 row in set (0.01 sec)
```

7. List the emps whose sal is equal to the average of max and minimum.

```
mysql> SELECT * FROM EMP WHERE SAL=(SELECT(MAX(SAL)+MIN(SAL))/2 FROM EMP);
Empty set (0.01 sec)
```

8. List the emps who joined in the company on the same date.

```
mysql> SELECT * FROM EMP WHERE HIREDATE IN (SELECT HIREDATE FROM EMP E WHERE EMPNO <> E.EMPNO);
Empty set (0.00 sec)
```

9. Display the employee names in upper and lower case.

```
mysql> SELECT UPPER(ENAME),LOWER(ENAME) FROM EMP;
+--------------+--------------+
| UPPER(ENAME) | LOWER(ENAME) |
+--------------+--------------+
| SMITH        | smith        |
| ALLEN        | allen        |
| WARD         | ward         |
| JONES        | jones        |
| MARTIN       | martin       |
| BLAKE        | blake        |
| CLARK        | clark        |
| SCOTT        | scott        |
| KING         | king         |
| TURNER       | turner       |
| ADAMS        | adams        |
| JAMES        | james        |
| FORD         | ford         |
| MILLAR       | millar       |
+--------------+--------------+
14 rows in set (0.02 sec)
```

10. Find the date of 3 days later from hiredate.

```
mysql> SELECT HIREDATE, HIREDATE + INTERVAL 3 DAY FROM EMP;
+------------+--------------------------+
| HIREDATE   | HIREDATE + INTERVAL 3 DAY |
+------------+--------------------------+
| 1980-12-17 | 1980-12-20               |
| 1981-02-20 | 1981-02-23               |
| 1981-02-22 | 1981-02-25               |
| 1981-04-02 | 1981-04-05               |
| 1981-09-28 | 1981-10-01               |
| 1981-05-01 | 1981-05-04               |
| 1981-06-09 | 1981-06-12               |
| 1982-12-09 | 1982-12-12               |
| 1981-11-17 | 1981-11-20               |
| 1981-09-08 | 1981-09-11               |
| 1983-01-12 | 1983-01-15               |
| 1981-12-03 | 1981-12-06               |
| 1981-12-03 | 1981-12-06               |
| 1982-12-23 | 1982-12-26               |
+------------+--------------------------+
14 rows in set (0.01 sec)
```