EXPERIMENT-9

TITLE: Group by & having clause

Objective: To understand the use of group by and having clause.

Write the SQL Queries for the following queries (use EMP and DEPT table of Exp 8).

1. List the Deptno where there are no emps.

   Commands:

   SELECT DEPTNO

   FROM DEPT

   WHERE DEPTNO NOT IN (SELECT DISTINCT DEPTNO FROM EMP);

   Output:

```
mysql> SELECT DEPTNO
    -> FROM DEPT
    -> WHERE DEPTNO NOT IN (SELECT DISTINCT DEPTNO FROM EMP);
+--------+
| DEPTNO |
+--------+
|     40 |
+--------+
1 row in set (0.00 sec)
```

2. List the No.of emp's and Avg salary within each department for each job.

   Commands:

   SELECT DEPTNO, JOB, COUNT(*) AS Num_Employees, AVG(SAL) AS Avg_Salary

   FROM EMP

   GROUP BY DEPTNO, JOB;

   Output:

```
mysql> SELECT DEPTNO, JOB, COUNT(*) AS Num_Employees, AVG(SAL) AS Avg_Salary
    -> FROM EMP
    -> GROUP BY DEPTNO, JOB;
+--------+-----------+---------------+------------+
| DEPTNO | JOB       | Num_Employees | Avg_Salary |
+--------+-----------+---------------+------------+
|     20 | CLERK     |             2 |   800.0000 |
|     30 | SALESMAN  |             4 |  1400.0000 |
|     20 | MANAGER   |             1 |  2975.0000 |
|     30 | MANAGER   |             1 |  2850.0000 |
|     10 | MANAGER   |             1 |  2450.0000 |
|     20 | ANALYST   |             2 |  3000.0000 |
|     10 | PRESIDENT |             1 |  5000.0000 |
|     30 | CLERK     |             1 |   950.0000 |
|     10 | CLERK     |             1 |  1300.0000 |
+--------+-----------+---------------+------------+
9 rows in set (0.00 sec)
```

3. Find the maximum average salary drawn for each job except for ‘President’.

Commands:

```
SELECT JOB, MAX(Avg_Salary) AS Max_Avg_Salary
FROM (SELECT JOB, AVG(SAL) AS Avg_Salary
    FROM EMP
    WHERE JOB != 'PRESIDENT'
    GROUP BY JOB) AS Subquery
GROUP BY JOB;
```

Output:

```
mysql> SELECT JOB, MAX(Avg_Salary) AS Max_Avg_Salary
    -> FROM (SELECT JOB, AVG(SAL) AS Avg_Salary
    ->       FROM EMP
    ->       WHERE JOB != 'PRESIDENT'
    ->       GROUP BY JOB) AS Subquery
    -> GROUP BY JOB;
+----------+-----------------+
| JOB      | Max_Avg_Salary |
+----------+-----------------+
| CLERK    |       962.5000 |
| SALESMAN |      1400.0000 |
| MANAGER  |      2758.3333 |
| ANALYST  |      3000.0000 |
+----------+-----------------+
4 rows in set (0.00 sec)
```

4. List the department details where at least two emps are working.

Commands:

```
SELECT DEPTNO
FROM EMP
GROUP BY DEPTNO
HAVING COUNT(*) >= 2;
```

Output:

```
mysql> SELECT DEPTNO
    -> FROM EMP
    -> GROUP BY DEPTNO
    -> HAVING COUNT(*) >= 2;
+--------+
| DEPTNO |
+--------+
|     20 |
|     30 |
|     10 |
+--------+
3 rows in set (0.00 sec)
```

5. List the no. of emps in each department where the no. is more than 3.

   Commands:

   SELECT DEPTNO, COUNT(*) AS Num_Employees

   FROM EMP

   GROUP BY DEPTNO

   HAVING COUNT(*) > 3;

   Output:

```
mysql> SELECT DEPTNO, COUNT(*) AS Num_Employees
    -> FROM EMP
    -> GROUP BY DEPTNO
    -> HAVING COUNT(*) > 3;
+--------+---------------+
| DEPTNO | Num_Employees |
+--------+---------------+
|     20 |             5 |
|     30 |             6 |
+--------+---------------+
2 rows in set (0.00 sec)
```

6. List the names of the emps who are getting the highest sal dept wise.

   Commands:

   SELECT E.ENAME, E.DEPTNO

   FROM EMP E

   WHERE (E.DEPTNO, E.SAL) IN (SELECT DEPTNO, MAX(SAL) FROM EMP GROUP BY DEPTNO);

   Output:

```
mysql> SELECT E.ENAME, E.DEPTNO
    -> FROM EMP E
    -> WHERE (E.DEPTNO, E.SAL) IN (SELECT DEPTNO, MAX(SAL) FROM EMP GROUP BY DEPTNO);
+-------+--------+
| ENAME | DEPTNO |
+-------+--------+
| BLAKE |     30 |
| SCOTT |     20 |
| KING  |     10 |
| FORD  |     20 |
+-------+--------+
4 rows in set (0.00 sec)
```

7. List the Deptno and their average salaries for dept with the average salary less than the averages for all departments.

   Commands:

   SELECT DEPTNO, AVG(SAL) AS Avg_Salary

   FROM EMP

   GROUP BY DEPTNO

   HAVING AVG(SAL) < (SELECT AVG(SAL) FROM EMP);

   Output:

```
mysql> SELECT DEPTNO, AVG(SAL) AS Avg_Salary
    -> FROM EMP
    -> GROUP BY DEPTNO
    -> HAVING AVG(SAL) < (SELECT AVG(SAL) FROM EMP);
+--------+-------------+
| DEPTNO | Avg_Salary |
+--------+-------------+
|     30 |  1566.6667 |
+--------+-------------+
1 row in set (0.00 sec)
```

EXPERIMENT-10

TITLE: Joins in SQL

AIM: To execute and verify the SQL commands using Join.

OBJECTIVE: SQL joins are used to query data from two or more tables, based on a relationship between certain columns in these tables.

Refer Experiment 7 & 8 and execute the same questions

1. List the details of the employees whose salaries are more than the employee BLAKE.

Commands:

```
SELECT e.*
FROM EMP e
JOIN EMP m ON e.MGR = m.EMPNO
WHERE e.SAL > m.SAL AND m.ENAME = 'BLAKE';
```

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN EMP m ON e.MGR = m.EMPNO
    -> WHERE e.SAL > m.SAL AND m.ENAME = 'BLAKE';
Empty set (0.00 sec)
```

2. List the employees whose jobs are the same as ALLEN.

Commands:

```
SELECT e.*
FROM EMP e
JOIN EMP a ON e.JOB = a.JOB
WHERE a.ENAME = 'ALLEN';
```

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN EMP a ON e.JOB = a.JOB
    -> WHERE a.ENAME = 'ALLEN';
+-------+--------+----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB      | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+----------+------+------------+------+------+--------+
|  7499 | ALLEN  | SALESMAN | 7698 | 1981-02-20 | 1600 |  300 |     30 |
|  7521 | WARD   | SALESMAN | 7698 | 1981-02-22 | 1250 |  500 |     30 |
|  7654 | MARTIN | SALESMAN | 7698 | 1981-09-28 | 1250 | 1400 |     30 |
|  7844 | TURNER | SALESMAN | 7698 | 1981-09-08 | 1500 |    0 |     30 |
+-------+--------+----------+------+------------+------+------+--------+
4 rows in set (0.00 sec)
```

3. List the employees whose salary is the same as FORD or SMITH in descending order of names.

Commands:

```
SELECT e.*
FROM EMP e
JOIN EMP f ON e.SAL = f.SAL
JOIN EMP s ON e.SAL = s.SAL
WHERE f.ENAME = 'FORD' OR s.ENAME = 'SMITH'
ORDER BY e.ENAME DESC;
```

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN EMP f ON e.SAL = f.SAL
    -> JOIN EMP s ON e.SAL = s.SAL
    -> WHERE f.ENAME = 'FORD' OR s.ENAME = 'SMITH'
    -> ORDER BY e.ENAME DESC;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7369 | SMITH | CLERK   | 7902 | 1980-12-17 |  500 |  800 |     20 |
|  7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7788 | SCOTT | ANALYST | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |
+-------+-------+---------+------+------------+------+------+--------+
5 rows in set (0.00 sec)
```

4. List the employees whose jobs are the same as MILLER or salary is more than ALLEN.

Commands:

```
SELECT e.*
FROM EMP e
JOIN EMP m ON e.JOB = m.JOB
WHERE m.ENAME = 'MILLER'
  OR e.SAL > (SELECT SAL FROM EMP WHERE ENAME = 'ALLEN');
```

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN EMP m ON e.JOB = m.JOB
    -> WHERE m.ENAME = 'MILLER'
    ->    OR e.SAL > (SELECT SAL FROM EMP WHERE ENAME = 'ALLEN');
+-------+--------+-----------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-----------+------+------------+------+------+--------+
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450 | NULL |     10 |
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000 | NULL |     10 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1982-12-09 | 3000 | NULL |     20 |
|  7934 | MILLER | CLERK     | 7782 | 1982-01-23 | 1300 | NULL |     10 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-03 |  950 | NULL |     30 |
|  7876 | ADAMS  | CLERK     | 7788 | 1983-01-12 | 1100 | NULL |     20 |
|  7369 | SMITH  | CLERK     | 7902 | 1980-12-17 |  500 |  800 |     20 |
+-------+--------+-----------+------+------------+------+------+--------+
18 rows in set (0.00 sec)
```

5. Find the highest paid employee of the sales department.

Commands:

SELECT e.*
FROM EMP e
JOIN DEPT d ON e.DEPTNO = d.DEPTNO
WHERE d.DNAME = 'SALES'
ORDER BY e.SAL DESC
LIMIT 1;

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN DEPT d ON e.DEPTNO = d.DEPTNO
    -> WHERE d.DNAME = 'SALES'
    -> ORDER BY e.SAL DESC
    -> LIMIT 1;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7698 | BLAKE | MANAGER | 7839 | 1981-05-01 | 2850 | NULL |     30 |
+-------+-------+---------+------+------------+------+------+--------+
1 row in set (0.01 sec)
```

6. List the employees who are senior to the most recently hired employee working under King.

Commands:

```
SELECT e.*
FROM EMP e
JOIN EMP m ON e.MGR = m.EMPNO
WHERE m.HIREDATE = (SELECT MAX(HIREDATE) FROM EMP WHERE MGR = (SELECT EMPNO FROM EMP WHERE ENAME = 'KING'));
```

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN EMP m ON e.MGR = m.EMPNO
    -> WHERE m.HIREDATE = (SELECT MAX(HIREDATE) FROM EMP WHERE MGR = (SELECT EMPNO FROM EMP WHERE ENAME = 'KING'));
+-------+--------+-------+------+------------+------+------+--------+
| EMPNO | ENAME  | JOB   | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+--------+-------+------+------------+------+------+--------+
|  7934 | MILLER | CLERK | 7782 | 1982-01-23 | 1300 | NULL |     10 |
+-------+--------+-------+------+------------+------+------+--------+
1 row in set (0.00 sec)
```

7. List the names of the employees who are getting the highest salary department-wise.

Commands:

```
SELECT e.ENAME, e.DEPTNO
FROM EMP e
JOIN (SELECT DEPTNO, MAX(SAL) AS Max_Sal FROM EMP GROUP BY DEPTNO) AS MaxSalaries
ON e.DEPTNO = MaxSalaries.DEPTNO AND e.SAL = MaxSalaries.Max_Sal;
```

Output:

```
mysql> SELECT e.ENAME, e.DEPTNO
    -> FROM EMP e
    -> JOIN (SELECT DEPTNO, MAX(SAL) AS Max_Sal FROM EMP GROUP BY DEPTNO) AS MaxSalaries
    -> ON e.DEPTNO = MaxSalaries.DEPTNO AND e.SAL = MaxSalaries.Max_Sal;
+-------+--------+
| ENAME | DEPTNO |
+-------+--------+
| BLAKE |     30 |
| SCOTT |     20 |
| KING  |     10 |
| FORD  |     20 |
+-------+--------+
4 rows in set (0.00 sec)
```

8. List the employees whose salary is equal to the average of the maximum and minimum salary.

Commands:
SELECT e.*
FROM EMP e
WHERE e.SAL = (SELECT (MAX(SAL) + MIN(SAL)) / 2 FROM EMP);

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> WHERE e.SAL = (SELECT (MAX(SAL) + MIN(SAL)) / 2 FROM EMP);
Empty set (0.00 sec)
```

9. List the employees who joined the company on the same date.

Commands:
SELECT e1.*
FROM EMP e1
JOIN EMP e2 ON e1.HIREDATE = e2.HIREDATE AND e1.EMPNO != e2.EMPNO;

Output:

```
mysql> SELECT e1.*
    -> FROM EMP e1
    -> JOIN EMP e2 ON e1.HIREDATE = e2.HIREDATE AND e1.EMPNO != e2.EMPNO;
+-------+-------+---------+------+------------+------+------+--------+
| EMPNO | ENAME | JOB     | MGR  | HIREDATE   | SAL  | COMM | DEPTNO |
+-------+-------+---------+------+------------+------+------+--------+
|  7902 | FORD  | ANALYST | 7566 | 1981-12-03 | 3000 | NULL |     20 |
|  7900 | JAMES | CLERK   | 7698 | 1981-12-03 |  950 | NULL |     30 |
+-------+-------+---------+------+------------+------+------+--------+
2 rows in set (0.00 sec)
```

10. Find out the employees who joined the company before their Managers.

Commands:

```
SELECT e.*
FROM EMP e
JOIN EMP m ON e.MGR = m.EMPNO
WHERE e.HIREDATE < m.HIREDATE;
```

Output:

```
mysql> SELECT e.*
    -> FROM EMP e
    -> JOIN EMP m ON e.MGR = m.EMPNO
    -> WHERE e.HIREDATE < m.HIREDATE;
+-------+-------+----------+------+------------+------+------+--------+
| 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  | 7839 | 1981-04-02 | 2975 | NULL |     20 |
|  7698 | BLAKE | MANAGER  | 7839 | 1981-05-01 | 2850 | NULL |     30 |
|  7782 | CLARK | MANAGER  | 7839 | 1981-06-09 | 2450 | NULL |     10 |
+-------+-------+----------+------+------------+------+------+--------+
6 rows in set (0.00 sec)
```