FAIZAN CHOUDHARY

20BCS021

DBMS LAB

7th March 2022

Creation:

- ```
  mysql> use 20bcs021_faizan;
  ```
- ```
  mysql> create table employee2
      -> (
      -> emp_id   int(4),
      -> emp_name varchar(15),
      -> salary   int(8),
      -> d_no     varchar(10)     not null,
      -> primary key(d_no)
      -> );
  ```
- ```
  mysql> insert into employee2 values
      -> (101, 'Amit', 25000, 'D1001'),
      -> (102, 'Sunil', 20000, 'D1002'),
      -> (103, 'Rakesh', 18000, 'D1003'),
      -> (104, 'Ajay', 16000, 'D1001'),
      -> (105, 'Suhail', 20000, 'D1002'),
      -> (106, 'Arif', 18000, 'D1004'),
      -> (107, 'Suresh', 24000, 'D1002'),
      -> (108, 'Vijay', 22000, 'D1003');
  ```

- ```
  mysql> select * from employee2;
  ```

```
mysql> select * from employee2;
+--------+----------+--------+-------+
| emp_id | emp_name | salary | d_no  |
+--------+----------+--------+-------+
|    101 | Amit     |  25000 | D1001 |
|    102 | Sunil    |  20000 | D1002 |
|    103 | Rakesh   |  18000 | D1003 |
|    104 | Ajay     |  16000 | D1001 |
|    105 | Suhail   |  20000 | D1002 |
|    106 | Arif     |  18000 | D1004 |
|    107 | Suresh   |  24000 | D1002 |
|    108 | Vijay    |  22000 | D1003 |
+--------+----------+--------+-------+
8 rows in set (0.00 sec)
```

- ```
  mysql> create table department3
      -> (
      -> d_no     varchar(10)     not null,
      -> dept_name varchar(15)
      -> );
  ```
- ```
  mysql> insert into department3 values
      -> ('D1001', 'IT'),
      -> ('D1002', 'Sales'),
      -> ('D1003', 'Marketing'),
      -> ('D1004', 'HR');
  ```
- ```
  mysql> select * from department3;
  ```

```
mysql> select * from department3;
+-------+-----------+
| d_no  | dept_name |
+-------+-----------+
| D1001 | IT        |
| D1002 | Sales     |
| D1003 | Marketing |
| D1004 | HR        |
+-------+-----------+
4 rows in set (0.00 sec)
```

Queries:

1. **Display total sum required to pay the salary of all employees.**
   ```
   mysql> select sum(salary) as tot_sal from employee2;
   ```

   OUTPUT:

```
+---------+
| tot_sal |
+---------+
|  163000 |
+---------+
1 row in set (0.00 sec)
```

2. **Display the average salary, minimum salary and maximum salary of the Company.**
   ```
   mysql> select avg(salary) as avg_sal, min(salary) as min_sal,
   max(salary) as max_sal from employee2;
   ```

   OUTPUT:
   ```

```
+------------+---------+---------+
| avg_sal    | min_sal | max_sal |
+------------+---------+---------+
| 20375.0000 |   16000 |   25000 |
+------------+---------+---------+
1 row in set (0.00 sec)
```

3. **Display the sum of salary department-wise.**

```
mysql> select employee2.d_no, sum(salary), dept_name
    -> from employee2, department3
    -> where employee2.d_no = department3.d_no
    -> group by dept_name;
```

OUTPUT:

```
+-------+-------------+-----------+
| d_no  | sum(salary) | dept_name |
+-------+-------------+-----------+
| D1001 |       41000 | IT        |
| D1002 |       64000 | Sales     |
| D1003 |       40000 | Marketing |
| D1004 |       18000 | HR        |
+-------+-------------+-----------+
4 rows in set (0.00 sec)
```

4. **Display the maximum salary department-wise.**

```
mysql> select employee2.d_no, max(salary), dept_name
    -> from employee2, department3
    -> where employee2.d_no = department3.d_no
    -> group by dept_name;
```

OUTPUT:

```
+-------+-------------+-----------+
| d_no  | max(salary) | dept_name |
+-------+-------------+-----------+
| D1001 |       25000 | IT        |
| D1002 |       24000 | Sales     |
| D1003 |       22000 | Marketing |
| D1004 |       18000 | HR        |
+-------+-------------+-----------+
4 rows in set (0.00 sec)
```

5. **Display the details of the employee who earns the maximum salary.**

```
mysql> select *
    -> from employee2
    -> where salary = (select max(salary) from employee2);
```

OUTPUT:

```
+--------+----------+--------+-------+
| emp_id | emp_name | salary | d_no  |
+--------+----------+--------+-------+
|    101 | Amit     |  25000 | D1001 |
+--------+----------+--------+-------+
1 row in set (0.00 sec)
```

6. **Display details of every employee having maximum salary in his Department.**

```
mysql> select *
    -> from employee2 natural join department3
    -> where (salary, dept_name) in
          (select max(salary), dept_name
           from employee2 natural join department3
           group by dept_name);
```

OUTPUT:

```
+-------+--------+----------+--------+-----------+
| d_no  | emp_id | emp_name | salary | dept_name |
+-------+--------+----------+--------+-----------+
| D1001 |    101 | Amit     |  25000 | IT        |
| D1004 |    106 | Arif     |  18000 | HR        |
| D1002 |    107 | Suresh   |  24000 | Sales     |
| D1003 |    108 | Vijay    |  22000 | Marketing |
+-------+--------+----------+--------+-----------+
4 rows in set (0.00 sec)
```

7. **Display the details of the employee who earns more salary than the average salary of his department.**

```
mysql> select *
    -> from employee2 natural join
          (select dept_name, d_no, avg(salary) as avg_sal
           from department3 natural join employee2
           group by dept_name) temp
    -> where employee2.d_no = temp.d_no and salary > avg_sal;
```

OUTPUT:

```
+-------+--------+----------+--------+-----------+------------+
| d_no  | emp_id | emp_name | salary | dept_name | avg_sal    |
+-------+--------+----------+--------+-----------+------------+
| D1001 |    101 | Amit     |  25000 | IT        | 20500.0000 |
| D1002 |    107 | Suresh   |  24000 | Sales     | 21333.3333 |
| D1003 |    108 | Vijay    |  22000 | Marketing | 20000.0000 |
+-------+--------+----------+--------+-----------+------------+
3 rows in set (0.00 sec)
```

8. **Display total number of employees in each department along with the department name.**

```
mysql> select dept_name, count(*) as no_employees
    -> from employee2 natural join department3
    -> group by dept_name;
```

OUTPUT:

```
+-----------+--------------+
| dept_name | no_employees |
+-----------+--------------+
| IT        |            2 |
| Sales     |            3 |
| Marketing |            2 |
| HR        |            1 |
+-----------+--------------+
4 rows in set (0.00 sec)
```