SQL - 2 Construct Week
Jai Phookan
Fw17_0739

Q1.

Table: Person
+-------------------+------------+
| Column Name | Type |
+-------------------+------------+
| id | int |
| email | varchar |
+-------------------+------------+
id is the primary key column for this table.
Each row of this table contains an email. The emails will not contain uppercase letters.
Write an SQL query to report all the duplicate emails.
Return the result table in any order.
The query result format is in the following example.
Input:
Person table:
+----+------------+
| id | email |
+----+------------+
1	a@b.com
2	c@d.com
3	a@b.com
+----+------------+
Output:
+-------------+
| Email |
+-------------+
| a@b.com |
+-------------+
Explanation: a@b.com is repeated two times

```
mysql> create table Person(id int primary key, email varchar(30));
Query OK, 0 rows affected (0.13 sec)

mysql> insert into Person values(1, "a@b.com");
Query OK, 1 row affected (0.06 sec)

mysql> insert into Person values(2, "c@d.com");
Query OK, 1 row affected (0.08 sec)

mysql> insert into Person values(3, "a@b.com");
Query OK, 1 row affected (0.04 sec)
```

mysql> select email from Person group by email HAVING count(email) > 1;

```
+---------+
| email   |
+---------+
| a@b.com |
+---------+
1 row in set (0.01 sec)
```

Question2:

1) Write an SQL query to show the second highest salary from a Worker table

```
SELECT * FROM Worker WHERE salary= (SELECT DISTINCT(salary) from Worker ORDER BY salary desc limit 1,1);
+-----------+------------+-----------+--------+---------------------+------------+
| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE        | DEPARTMENT |
+-----------+------------+-----------+--------+---------------------+------------+
|         3 | Vishal     | Singhal   | 300000 | 2014-02-20 09:00:00 | HR         |
+-----------+------------+-----------+--------+---------------------+------------+
1 row in set (0.02 sec)
```

2) Write an SQL query to determine the 5 highest salary from a Worker table

```
select * from Worker order by salary desc limit 5;
+-----------+------------+-----------+--------+---------------------+------------+
| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE        | DEPARTMENT |
+-----------+------------+-----------+--------+---------------------+------------+
|         4 | Amitabh    | Singh     | 500000 | 2014-02-20 09:00:00 | Admin      |
|         5 | Vivek      | Bhati     | 500000 | 2014-06-11 09:00:00 | Admin      |
|         3 | Vishal     | Singhal   | 300000 | 2014-02-20 09:00:00 | HR         |
|         6 | Vipul      | Diwan     | 200000 | 2014-06-11 09:00:00 | Account    |
|         1 | Monika     | Arora     | 100000 | 2014-02-20 09:00:00 | HR         |
+-----------+------------+-----------+--------+---------------------+------------+
```

3. Write an SQL query to show only even rows from a Worker table.

```
select * from Worker where mod(worker_id, 2) = 0 order by worker_id asc;

+-----------+------------+-----------+--------+---------------------+------------+
| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE        | DEPARTMENT |
+-----------+------------+-----------+--------+---------------------+------------+
|         2 | Niharika   | Verma     |  80000 | 2014-06-11 09:00:00 | Admin      |
|         4 | Amitabh    | Singh     | 500000 | 2014-02-20 09:00:00 | Admin      |
|         6 | Vipul      | Diwan     | 200000 | 2014-06-11 09:00:00 | Account    |
|         8 | Geetika    | Chauhan   |  90000 | 2014-04-11 09:00:00 | Admin      |
+-----------+------------+-----------+--------+---------------------+------------+
```

4. Write an SQL query to fetch the no. of workers for each department in the descending order from the Worker table.

```
select count(worker_id), department from Worker group by department order by count(worker_id) desc;
```

```
+------------------+------------+
| count(worker_id) | department |
+------------------+------------+
|                4 | Admin      |
|                2 | HR         |
|                2 | Account    |
+------------------+------------+
```

5. Write an SQL query to fetch the list of employees with the same salary from the Worker table.

```
select * from Worker where salary in (select salary from Worker w where worker.worker_id <> w.worker_id);
```

```
+-----------+------------+-----------+--------+---------------------+------------+
| WORKER_ID | FIRST_NAME | LAST_NAME | SALARY | JOINING_DATE        | DEPARTMENT |
+-----------+------------+-----------+--------+---------------------+------------+
|         5 | Vivek      | Bhati     | 500000 | 2014-06-11 09:00:00 | Admin      |
|         4 | Amitabh    | Singh     | 500000 | 2014-02-20 09:00:00 | Admin      |
+-----------+------------+-----------+--------+---------------------+------------+
2 rows in set (0.00 sec)
```

QUESTION 3.

1.
Write an SQL query to fetch the employees whose name begins with any two characters, followed by a text “hn” and ending with any sequence of characters.

```
select * from EmployeeDetail where fullName like "%%hn%";
+-------+-----------+-----------+---------------+---------+
| empID | fullName  | ManagerId | DateOfJoining | city    |
+-------+-----------+-----------+---------------+---------+
|   121 | John Snow |       321 | 2014-01-31    | Toronto |
+-------+-----------+-----------+---------------+---------+
1 row in set (0.00 sec
```

2. Write an SQL query to fetch common records between two tables.

```
select empID from EmployeeDetail union select empID from employeeSalary;
+-------+
| empID |
+-------+
|   121 |
```

```
|   321 |
|   421 |
+-------+
```

3. Write an SQL query to fetch records that are present in one table but not in another table.

```
select * from Employeedetail where not exists (select * from employeesalary where Employeedetail.empID =
employeesalary.empID);

Empty set (0.00 sec)
```

4. Write an SQL query to find the maximum, minimum, and average salary of the employees.

```
mysql>  select max(salary) from employeeDetail natural join employeeSalary;
+-------------+
| max(salary) |
+-------------+
|       12000 |
+-------------+
1 row in set (0.00 sec)

mysql> select min(salary) from employeeDetail natural join employeeSalary;
+-------------+
| min(salary) |
+-------------+
|        8000 |
+-------------+
1 row in set (0.00 sec)

mysql> select avg(salary) from employeeDetail natural join employeeSalary;
+-------------+
| avg(salary) |
+-------------+
|  10000.0000 |
+-------------+
1 row in set (0.01 sec)
```

5. Fetch all the employees who are not working on any project.

```
select * from employeeDetail natural join employeeSalary where project = null;
Empty set (0.01 sec)
```