University of Peradeniya CO527 Advanced Database Systems

Lab Task:

Refer to the Company ER diagram shown in Figure 1 and create a database named Company. All the questions listed below are based on this database.

1. Load data to each of the tables from the given .sql files. It should have the following mentioned number of records for each table if the import was successful.

Initially there were five databases. Created company database using following commands and it is shown in below figure.

- Create database company;
- Show databases;

To use this database for further implementation, use below code.

Use company;

Then tables were created according to the given ER diagram.

```
MariaDB [company]>
MariaDB [company]> CREATE TABLE dept_manager (
              emp no int,
    -> dept_no char(4),
   -> from date date,
   -> to date date,
               primary key (dept_no,emp_no),
   ->
    -> foreign key (dept_no) references departments (dept_no),
    -> foreign key (emp_no) references employees (emp_no)
   -> );
Query OK, 0 rows affected (0.23 sec)
MariaDB [company]>
MariaDB [company]> CREATE TABLE titles (
              emp_no int,
   -> title varchar(50),
    -> from date date,
    -> to date date,
              primary key (emp no, title, from date, to date),
    -> foreign key (emp no) references employees (emp no)
    -> );
Query OK, 0 rows affected (0.23 sec)
MariaDB [company]>
MariaDB [company]> CREATE TABLE salaries (
              emp no int,
    -> salary int,
   -> from_date date,
   -> to date date,
               primary key (emp_no,from_date,to_date),
    -> foreign key (emp no) references employees (emp no)
    -> );
Query OK, 0 rows affected (0.27 sec)
MariaDB [company]>
MariaDB [company]> CREATE TABLE dept_emp (
              emp_no int,
    -> dept_no char(4),
   -> from date date,
    -> to date date,
               primary key (dept no,emp no),
    -> foreign key (dept_no) references departments (dept_no),
    -> foreign key (emp no) references employees (emp no)
    -> );
Query OK, 0 rows affected (0.28 sec)
```

Then stored all the .sql files in C:\xampp\mysql\bin directory. Using below codes, load the data in to tables.

- mysql -u root company < load_employees.sql
- mysql -u root company < load_departments.sql
- mysql -u root company < load_dept_emp.sql
- mysql -u root company < load dept manager.sql
- mysql -u root company < load dept salaries1.sql
- mysql -u root company < load_dept_salaries2.sql
- mysql -u root company < load_titles.sql

```
MariaDB [company]>
MariaDB [company]> select count(*) from employees;
  count(*)
    300024
 row in set (0.48 sec)
MariaDB [company]> select count(*) from dept_manager;
  count(*)
        24
 row in set (0.05 sec)
MariaDB [company]> select count(*) from dept_emp;
  count(*)
    331603
 row in set (0.38 sec)
MariaDB [company]> select count(*) from titles;
  count(*)
    443308
 row in set (0.38 sec)
MariaDB [company]> select count(*) from salaries;
  count(*)
   1876717
  row in set (0.91 sec)
```

```
MariaDB [company]> select count(*) from departments;

+-----+

| count(*) |

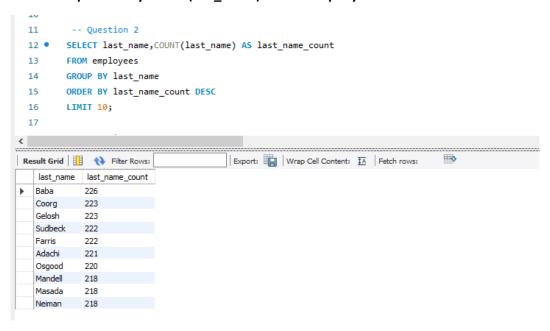
+-----+

| 9 |

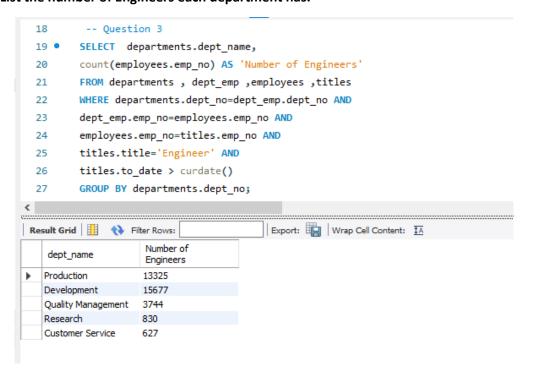
+-----+

1 row in set (0.02 sec)
```

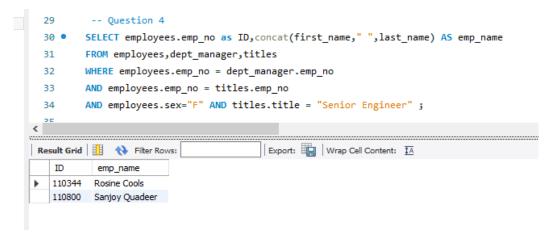
2. Find the top 10 family names(last_name) in the company.



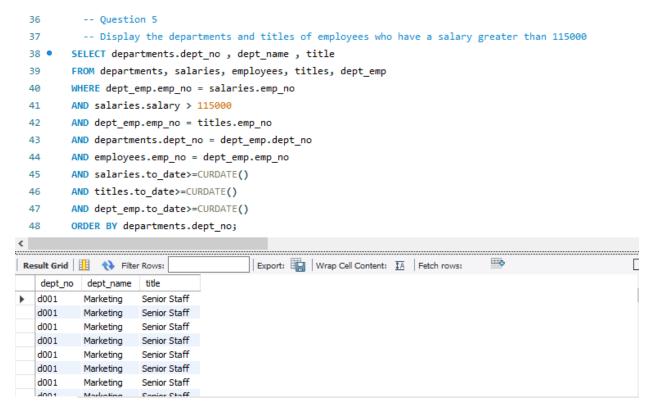
3. List the number of Engineers each department has.

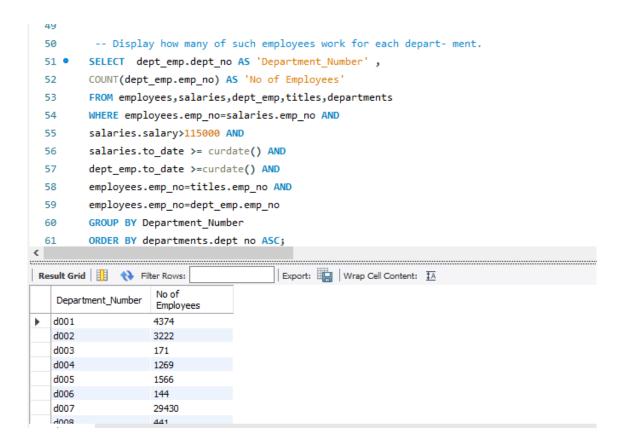


4. List all the female employees who are department managers and have worked as a senior engineer.

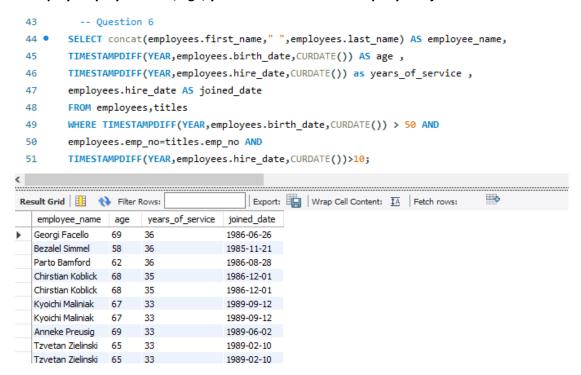


5. Display the departments and titles of employees who have a salary greater than 115000. Display how many of such employees work for each department.

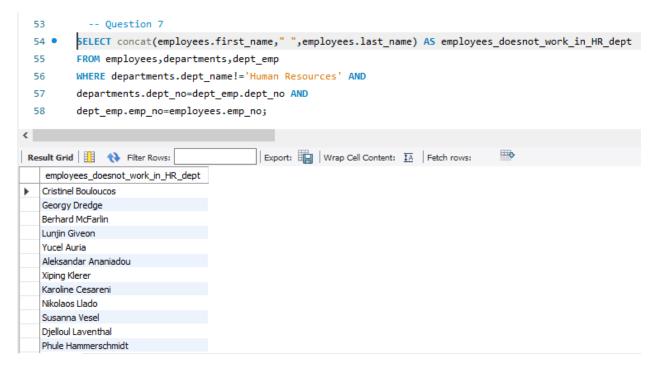




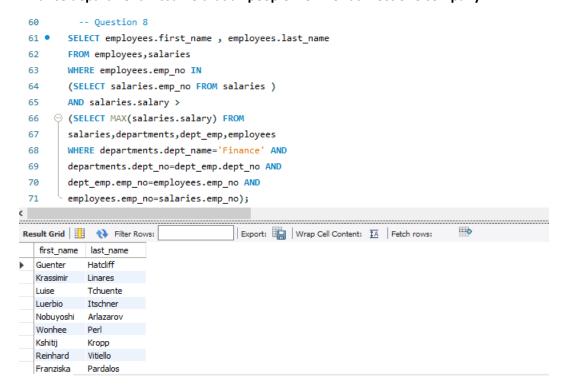
6. Assume that the company wants to reward the most senior employees who are more than 50 years of age and have contributed to the company for more than 10 years. Who is on the list? Display employee name, age, years of service in the company and joined date.



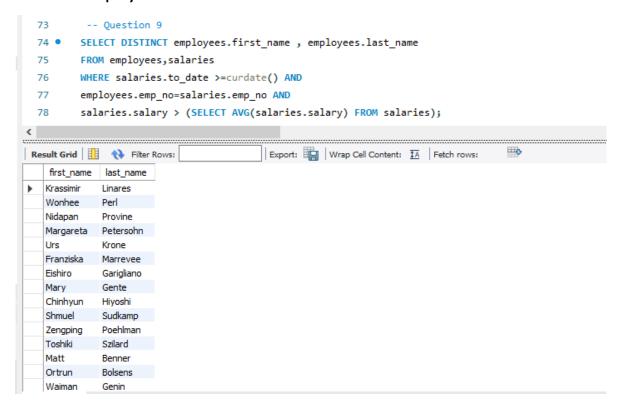
7. Find all the names (first name + last name) of employees in the database who do not work in the Human Resources department. Assume that all the people work for exactly one department.



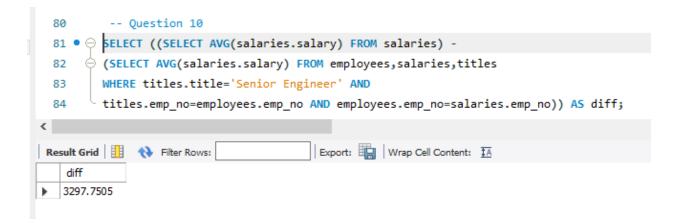
8. Find the names of all employees in the database who earn more than every employee in the Finance department. Assume that all people work for at most one company.



9. Find the names of all employees who earn more than the average salary of all employees of their company.



10. Compute the difference between the average salary of a Senior Engineer and the average salary of all employees (including Senior Engineers).



11. Create a view current dept emp (emp no, fromdate, todate) to show only the current department for each employee. You may have to use two views for this.

```
-- Question 11

107 • CREATE VIEW current_dept_emp AS

108 SELECT e.emp_no , de.from_date , de.to_date

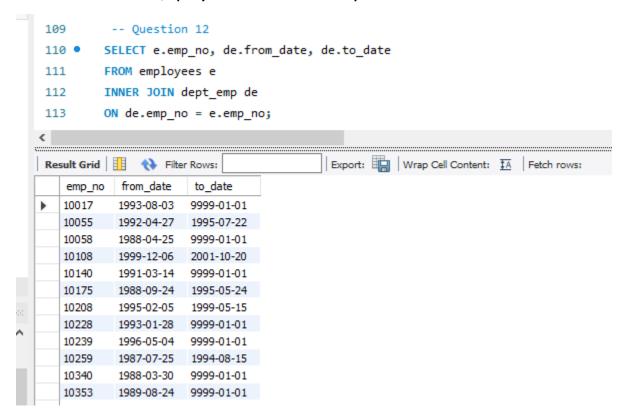
109 FROM employees e

110 INNER JOIN dept_emp de

111 ON e.emp_no = de.emp_no;

112
```

12. Write a normal SQL query to do the above task in problem 11.



13. Create a trigger to print salary changes of employees. For example, if you enter an SQL statement such as UPDATE salaries SET salary = salary + 1000 WHERE emp no = 1500, the trigger should fire once for each row that is updated and it should print the new and old salaries, and the difference.

```
119
          -- Question 13
        create table emp salary change
120 •
121
122
                 old_salary int,
                 new_salary int,
123
                difference int,
124
                 action VARCHAR(50) DEFAULT NULL
125
126
            );
127
128
        delimiter $
        create trigger after_salaries_update
129 •
        after update on salaries
130
        for each row
131
    ⊖ begin
132
133
        insert into emp_salary_change
        SET action = 'update',
134
135
        old Salary = old.salary,
136
        new Salary = new.salary,
        difference = new.salary-old.salary;
137
138
        end $
        delimiter ;
139
```

14. Create a trigger that will cause an error when an update occurs that would result in a salary increase greater than 10% of the current salary

```
-- Ouestion 14
141
142
        delimiter $
143
144 •
        create trigger error_salary_update
        before update on salaries
145
146
        for each row
147
     ⊖ begin
        declare msg varchar(50);
148
149

    if(new.salary-old.salary)>(old.salary*0.1)then

            set msg ="Error : Salary increment > 10%";
150
            signal sqlstate '45000' set message_text = msg;
151
        end if;
152
153
        end $
154
        delimiter ;
```