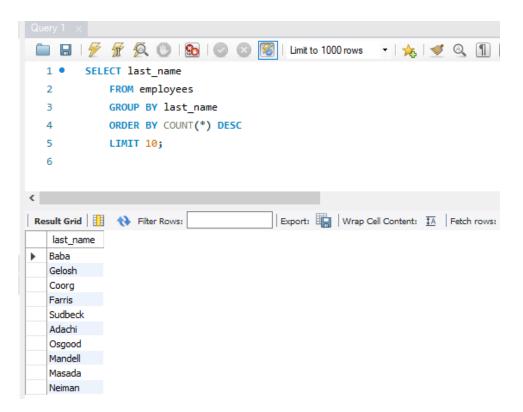


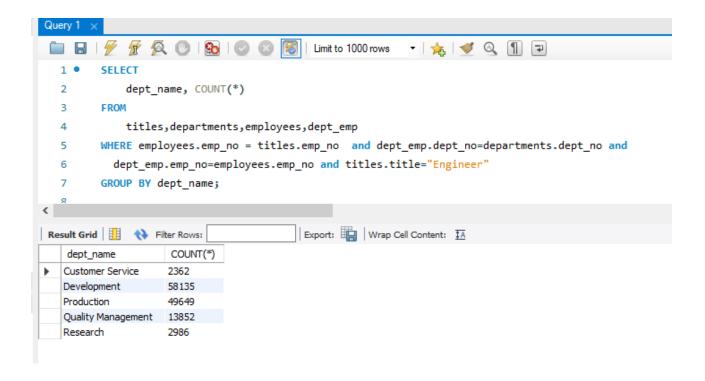
1) Load data to each of the tables from the given .sql files.



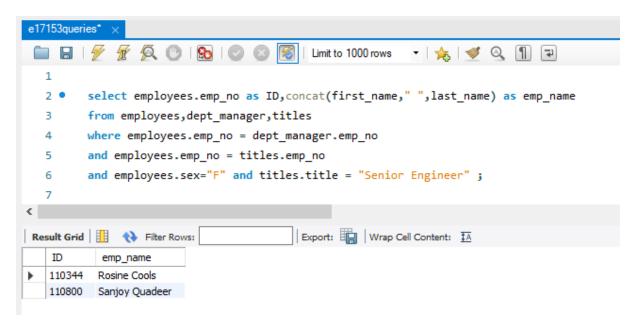
2) Find the top 10 family names(last_name) in the company.



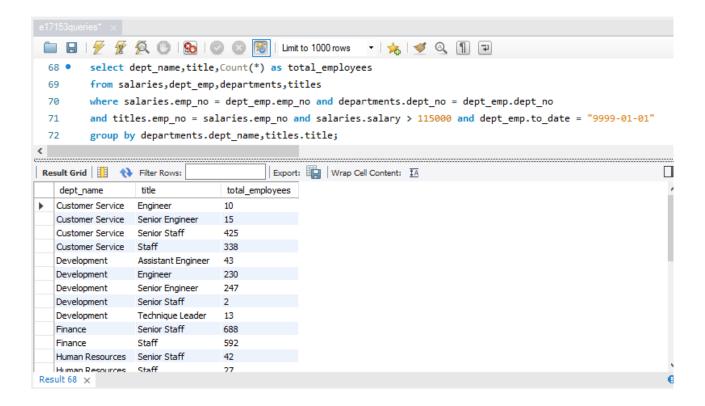
3) List the number of Engineers each department has



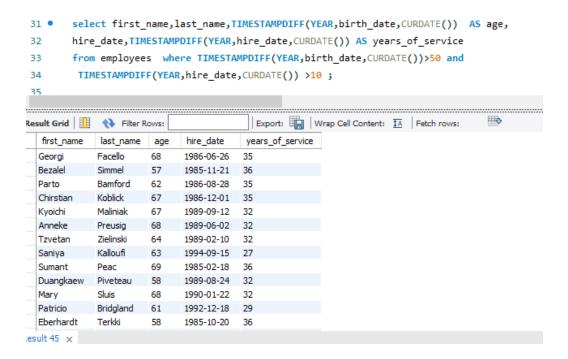
4) List all the female employees who are department managers and have worked as a senior engineer.



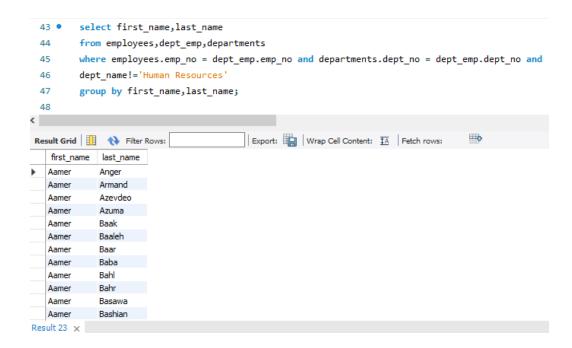
5) Display the departments and titles of employees who have a salary greater than 115000. Display how many of such employees work for each department.



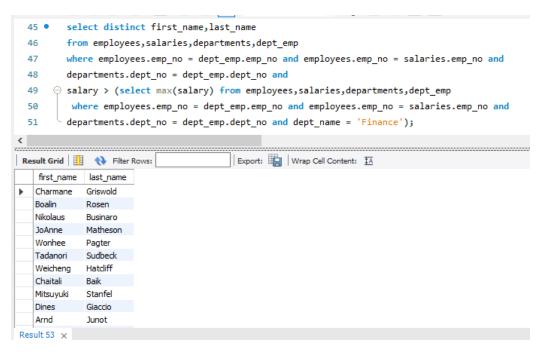
6) Assume that the company wants to reward the most senior employees who are more than 50 years of age and have contributed to the company for more than 10 years. Who is on the list? Display employee name, age, years of service in the company and joined date.



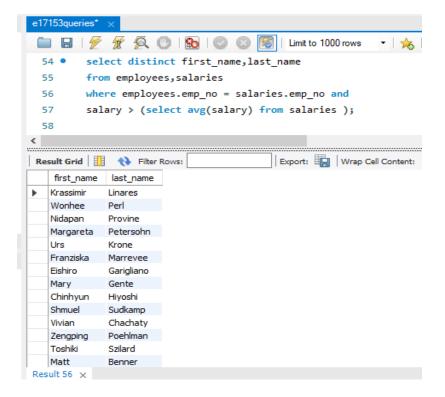
7) Find all the names (first name + last name) of employees in the database who do not work in the Human Resources department. Assume that all the people work for exactly one department.



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.



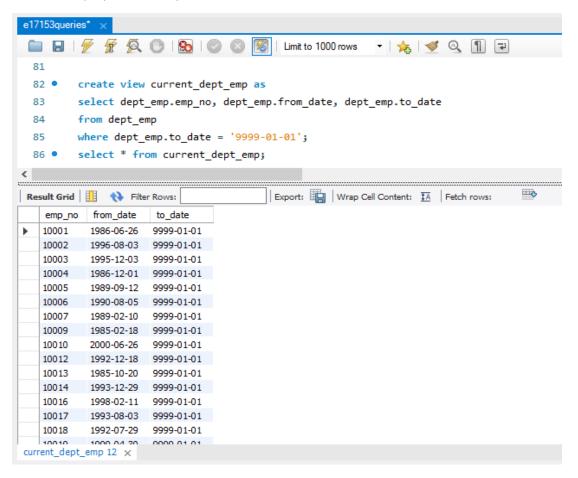
9) Find the names of all employees who earn more than the average salary of all employees of their company.



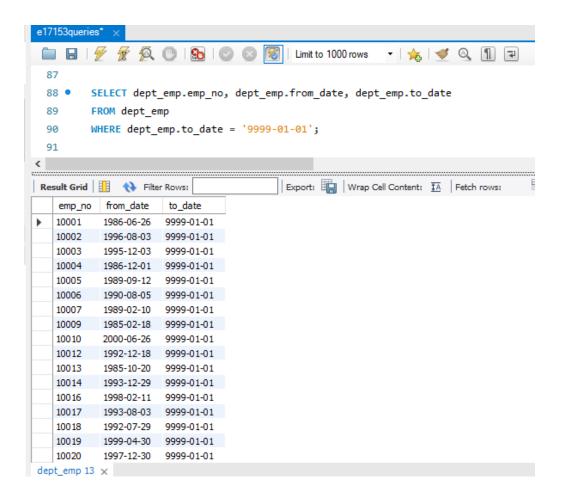
10) Compute the difference between the average salary of a Senior Engineer and the average salary of all employees (including Senior Engineers).

```
e17153queries*
   Limit to 1000 rows
                                        - | 🚖 | 🥩 🔍 🗻 🖘
 59
 60 •
      select
    61
      employees.emp_no = titles.emp_no and titles.title= 'Senior Engineer') -
 62
      (select avg(salary) from salaries)as difference;
 63
 64
< |
Result Grid
                             Export: Wrap Cell Content: IA
  difference
 -3297.7505
```

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



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.

```
Limit to 1000 rows
                                                    - | 🏡 | 🥩 🔍 👖 🖃
91
92 • ○ CREATE TABLE salary_change(
93
       sc emp no int,
94
       old_salary int,
       new_salary int,
       salary_diff int,
96
97
       PRIMARY KEY (sc_emp_no),
       FOREIGN KEY (sc_emp_no) REFERENCES employees(emp_no) ON DELETE CASCADE
98
99
       DELIMITER $$
100
101
       CREATE OR REPLACE TRIGGER print_salary
       AFTER UPDATE ON salaries
       FOR EACH ROW
103
104

⇒ BEGIN

     105
106
        insert into salary_change(sc_emp_no,old_salary,new_salary,salary_diff) Values
107
        (salaries.emp_no,:old..salary,new.salary,((new.salary)-(old.salary)));
       END IF
108
       END;
109
       DELIMITER;
110
111
       SELECT *
112
       FROM salary_changes
113
114
115
```

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.