Lauris Heinsalu, SOFTWARE ENGINEER – HOMEWORK

Exercise 1

Using the given script (see e-mail attachment) create the **company database** and write the following queries (SQL statements and results).

a. List all the departments and the number of employees per each department

Employees table has a department_id =0. A department_id with value of 0 doesn't exist in the departments table. This seems like an error in the data. I'm not sure whether it was expected to include this in the query, so I included it in case it was.

Solution:

```
SELECT d.department_name,
    COUNT(e.employee_id) number_of_employees
FROM departments d
    LEFT JOIN employees e ON e.department_id = d.department_id
GROUP BY d.department_name
UNION
SELECT 'UNEXISTANT_DEPT', COUNT(e.employee_id)
FROM employees e
WHERE department_id NOT IN (SELECT department_id FROM departments)
GROUP BY 1;
```

b. List the employees with the lowest salary per each department

Solution:

```
SELECT CASE WHEN d.department_name IS NULL THEN 'UNEXISTANT_DEPT' ELSE d.department_name
END department_name,
    e.first_name || ' ' || e.last_name employee_name,
    e.salary
FROM employees e
```

```
    LEFT JOIN departments d ON d.department_id = e.department_id
WHERE e.salary = (SELECT MIN(salary)
        FROM employees e1
            LEFT JOIN departments d1 ON e1.department_id = d1.department_id
        WHERE d1.department_id = d.department_id
          OR d1.department_id IS NULL);
```

c. Display the number of employees per city having not less than 34 employees

Since one Riga is in the US, I added country_name to the query to be able to distinguish between the two cities with the same name in different countries.

Solution:

```
WITH employees_per_city AS (SELECT l.city,
                    c.country_name,
                    COUNT(e.employee_id) employee_count
                FROM employees e
                    LEFT JOIN departments d ON d.department_id = e.department_id
                    JOIN locations l ON l.location_id = d.location_id
                    JOIN countries c ON c.country_id = l.country_id
                GROUP BY 1, 2)
SELECT *
FROM employees_per_city
WHERE employee_count >= 34;
```

d. List the 2 jobs with the maximum number of employees and the respective minimum and maximum salary per job.

Solution:

```
SELECT j.job_title,
```

```
    COUNT(e.employee_id) employee_count,
    j.min_salary,
    j.max_salary
FROM jobs j
    JOIN employees e ON j.job_id = e.job_id
GROUP BY 1
ORDER BY employee_count DESC LIMIT 2; /* fetch first 2 rows only with ties. sqliteonline.com
    doesn't seem to recognize fetch keyword. */
```

e. List the lowest and highest salary per city, the total number of employees per city, the number of employees earning the lowest salary, the number of employees earning the highest salary, the percentage of employees earning the highest salary per city and the percentage of employees earning the lowest salary per city. This information should be shown as 1 row per city containing all the information requested.

Solution:

```
WITH q AS (SELECT l.city,
                  c.country_name,
                  e.employee_id,
                  MIN(e.salary) min_sal,
                  MAX(e.salary) max_sal,
                  COUNT(e.employee_id) emp_count
           FROM locations l
                  JOIN countries c ON l.country_id = c.country_id
                  LEFT JOIN departments d ON d.location_id = l.location_id
                  LEFT JOIN employees e ON e.department_id = d.department_id
                  LEFT JOIN jobs j ON j.job_id = e.job_id
           GROUP BY 1, 2),

    i AS (SELECT e.employee_id, e.salary, d.location_id
           FROM departments d
                  JOIN employees e ON e.department_id = d.department_id)
```

```
SELECT q.city,
    q.country_name,
    q.min_sal,
    q.max_sal,
    q.emp_count,
    (SELECT COUNT(i.employee_id) FROM i WHERE i.salary = q.min_sal) no_of_lowest_paid,
    (SELECT COUNT(i.employee_id) FROM i WHERE i.salary = q.max_sal) no_of_highest_paid,
--(select count(e.employee_id) from employees e where e.salary = (select min(e1.salary) from employees e1) ) no_of_lowest_paid_everywhere,
--(select count(e.employee_id) from employees e where e.salary = (select max(e1.salary) from employees e1) ) no_of_highest_paid_everywhere,
    100 * (SELECT COUNT(i.employee_id) FROM i WHERE i.salary = q.min_sal) / q.emp_count AS pct_of_lowest_paid,
    100 * (SELECT COUNT(i.employee_id) FROM i WHERE i.salary = q.max_sal) / q.emp_count AS pct_of_highest_paid

FROM q
     LEFT JOIN i ON q.employee_id = i.employee_id;
```

f. Think yourself SQL statement which could be useful according data model for database users. Please substantiate your point of view and shortly explain what you did.

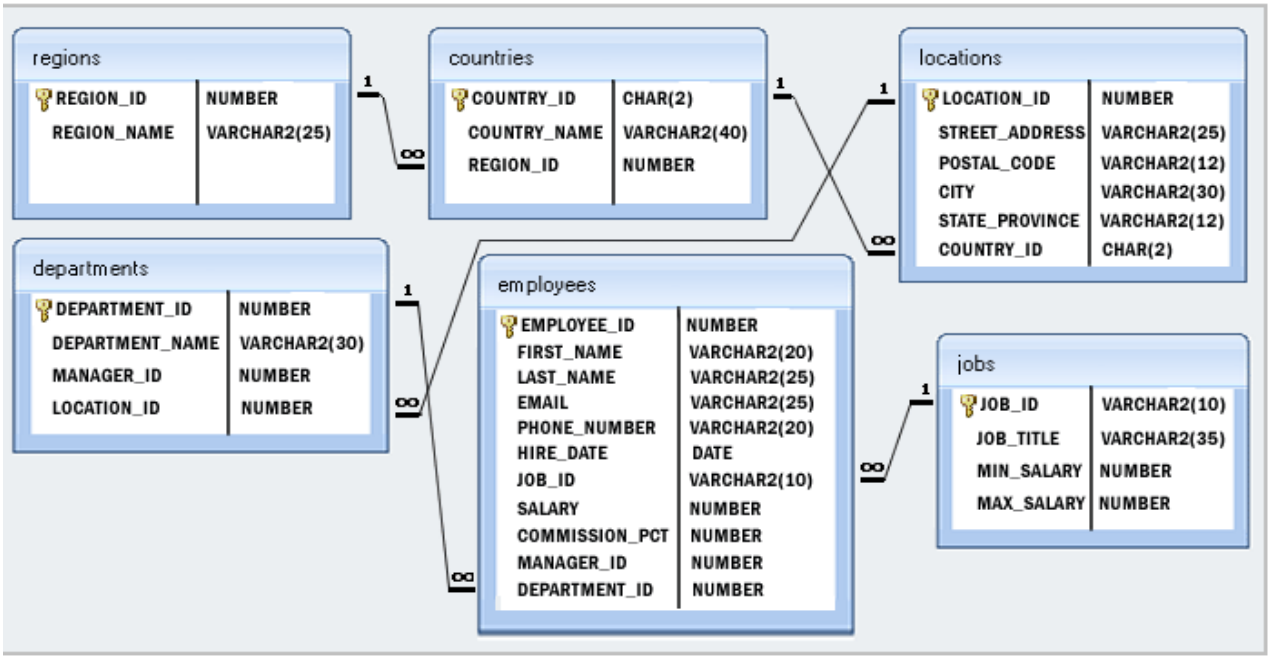
Solution:

I found the average salary of different jobs by country, to compare where the average salary for a given job position is higher or lower than in another country.

```
SELECT c.country_name,
    j.job_title,
    AVG(salary)
FROM locations l
     JOIN countries c ON l.country_id = c.country_id
     LEFT JOIN departments d ON d.location_id = l.location_id
```

```
        LEFT JOIN employees e ON e.department_id = d.department_id
        JOIN jobs j ON j.job_id = e.job_id
    GROUP BY 2, 1
    ORDER BY 2, 1;
```

To help you with these exercises you will find the data model here:



Exercise 2

What is the result of the following script and how would you fix it?

```
select *
from(
     SELECT a.fisrt_name,a.last_name,a.job_id
     FROM employees A
     WHERE A.slary >(Select avg(salary) from employees)) and a.slary < select
(max(slary) from employees)
) b
where b.job_id=SY_ANA;
```

Solution:

Result of the provided query on sqliteonline.com is "syntax error near 'and' "

To fix:

1. changed typos of slary to salary.

2. value of b.job_id needs to be in single quotes 'SYS_ANA'

3. opening bracket in subquery to get max salary needs to be before SELECT not before MAX.

4. there's an excess closing bracket after second subquery.

Fixed query:

```
SELECT first_name,
    last_name,
    job_id
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees)
 AND salary < (SELECT MAX(salary) FROM employees)
 AND job_id = 'SY_ANA';
```

Exercise 3

According to the **Jobs table** (provided in company_db.sql script in the previous step) what could you tell about the **employees table** for the System Analyst position (SY_ANA)?

Solution:

employees table would probably have job_id referencing jobs table job_id as a foreign key, which would be not null.

Part II

Pre-conditions:

- You can use this online tool for modeling (www.draw.io), but feel free to use paper or any other tool you want, just make sure when sending the results to attach them as image or PDF format to be able to open it.

Exercise 4

Create the data model for a restaurant chain database according to the following information:

Organization has restaurant chain (of which you want to store address, city, country, restaurant name and branch phone numbers (operator, kitchen, accounting). Each restaurant has seating tables with different seat count and types. Chain has flagship restaurant.

Restaurant clients are registered in restaurant database by operators while caller is making reservation. Information about operator contains name, surname, person identification number, personal phone number, branch where he is sitting (physically working) and restaurant for which he is making reservations. Client data could contain information about client name, surname, phone number and operator comment (discounts, allergens, etc.). Reservation can be made by everyone who call to restaurant operator (do not need store information about calls). While operator is creating reservation he is asking name, phone number, restaurant name (address), seating information, date, time and offers caller became as client. Operator can add comment about reservation.

Solution:

Schema is in a different file “Lauris Heinsalu, restaurant_schema.jpeg” attached to the email.

Connecting lines between tables are mostly connecting from sides not straight from top to bottom, because I couldn’t figure out in time how to connect tables that way on sqllite.com.

It wasn’t clear what was meant by “branch” in the exercise and there wasn’t time to ask for clarification since I had two non-work days to complete the homework. I assumed it was meant to be a unique name for any given restaurant, for example in the case where there’s more than one restaurant with the same name in the same building.