**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).

* 1. **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 I was expected to include this in the query and result, so just in case I unioned it into the query.

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;

* 1. **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);

* 1. **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;

* 1. **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. \*/

* 1. **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;

* 1. **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

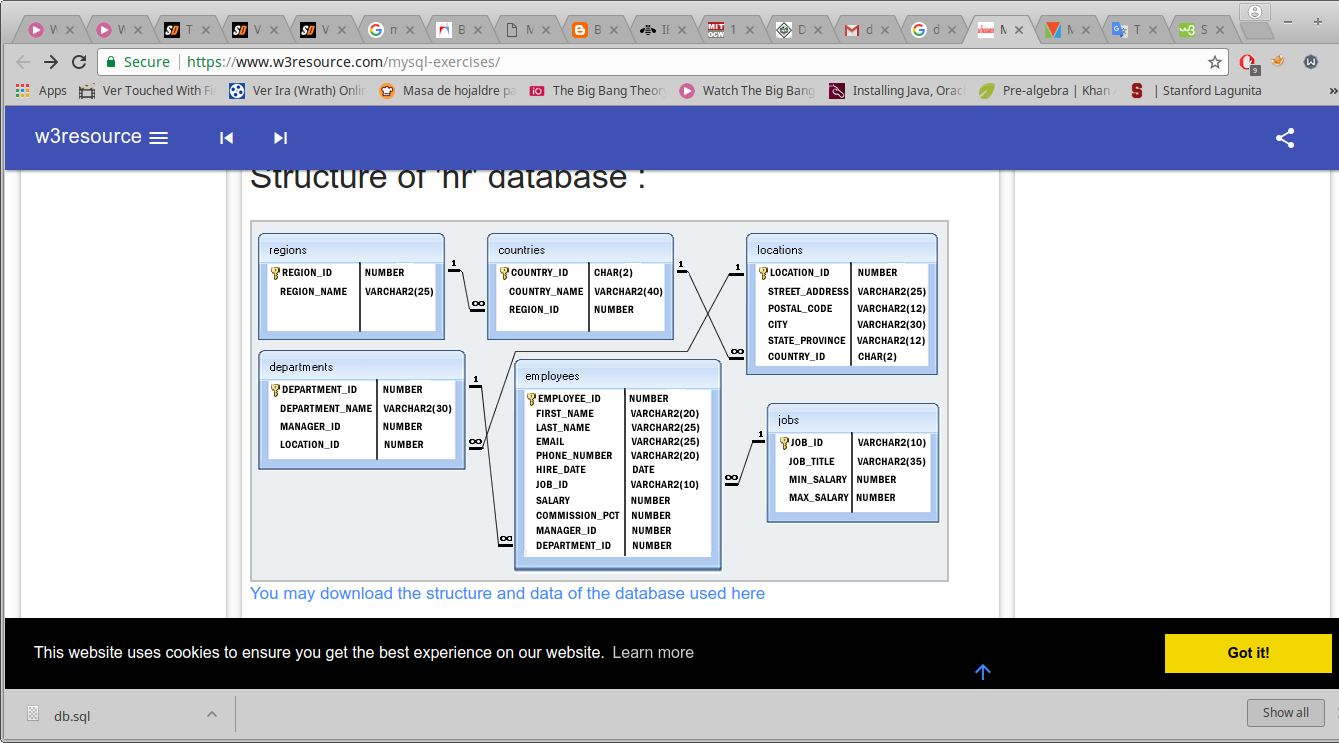
LEFT JOIN departments d ON d.location\_id = l.location\_id

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.

working version:

select \*

from(

SELECT a.first\_name,a.last\_name,a.job\_id

FROM employees A

WHERE A.salary > (Select avg(salary) from employees) and a.salary < (select max(salary) from employees)

) b

where b.job\_id='SY\_ANA';

A more readable version in my opinion:

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](http://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.

-- Sorry for the connecting lines between tables mostly connecting from sides not straight from top to bottom. I couldn’t quickly figure out how to connect tables that way on sqllite.com.

-- I’m not really sure what was meant by “branch” in the exercise and didn’t ask for clarification since I had two non-work days to complete the homework. I assumed with googles help that it’s 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.