Kyrusenko Roman

Telegram @roman\_k98

/\*SQL Exercises. Week 2

9. Fulfill the script below to achieve the following data:

Group all employees according to their age at January 1st, 1995 into four groups:

30 or younger, 31-40, 41-50 and older. Show average salary for each group and gender

(8 categories in total) Also add subtotals and grand total.

SELECT CASE -- let’s create age categories first

WHEN (datediff('1995-01-01', ... )THEN '30 and below'

...

END AS category,

...,

...

FROM employees e

INNER JOIN ...

WHERE ... -- filter out those employees, who were not employed at that date yet.

AND (SELECT MAX(to\_date) FROM dept\_emp de WHERE de.emp\_no = e.emp\_no

GROUP BY de.emp\_no) > '1995-01-01' -- this subquery filters out employees,

-- who already left the company by that date (think

– about how it works)

GROUP BY ... ... ...; -- don’t forget to add totals;\*/

/\*

1. For the current maximum annual wage in the company SHOW the full name of the

employee, department, current position, for how long the current position is held, and

total years of service in the company.

SELECT CONCAT(ee.first\_name, ' ', ee.last\_name) AS FULLNAME, dept\_name AS 'Department name', et.title, DATEDIFF(CURDATE(),et.from\_date) AS "TItle position time in days", ROUND(DATEDIFF(CURDATE(),ee.hire\_date)/365) AS '~YEARS work in company'

FROM employees.salaries AS es

INNER JOIN employees.employees AS ee ON (ee.emp\_no =es.emp\_no)

INNER JOIN employees.dept\_emp AS emp ON (ee.emp\_no = emp.emp\_no)

INNER JOIN employees.departments AS ed ON (emp.dept\_no=ed.dept\_no)

INNER JOIN employees.titles AS et ON (et.emp\_no=ee.emp\_no)

WHERE es.salary = (Select MAX(salary)

FROM employees.salaries

WHERE CURDATE() BETWEEN from\_date AND to\_date)

AND CURDATE() BETWEEN es.from\_date AND es.to\_date

AND CURDATE() BETWEEN et.from\_date AND et.to\_date

AND CURDATE() BETWEEN emp.from\_date AND emp.to\_date

;

2. For each department, show its name and current manager’s name, last name, and

current salary

SELECT ed.dept\_name, ee.first\_name, ee.last\_name, es.salary

FROM employees.employees AS ee

INNER JOIN employees.dept\_manager AS edm USING (emp\_no)

INNER JOIN employees.departments AS ed USING (dept\_no)

INNER JOIN employees.salaries AS es ON (es.emp\_no=edm.emp\_no)

WHERE curdate() BETWEEN edm.from\_date AND edm.to\_date

AND curdate() BETWEEN es.from\_date AND es.to\_date

;

3. Show for each employee, their current salary and their current manager’s current salary.

SELECT es.emp\_no AS 'Працівник', es.salary AS 'Зарплата працівника',es\_manager.emp\_no, es\_manager.salary AS 'Зарплата його менеджера'

FROM employees.salaries AS es

INNER JOIN employees.dept\_emp AS ede ON (es.emp\_no = ede.emp\_no)

INNER JOIN employees.dept\_manager AS edm ON (ede.dept\_no=edm.dept\_no)

INNER JOIN employees.salaries AS es\_manager ON (edm.emp\_no=es\_manager.emp\_no)

WHERE CURDATE() BETWEEN edm.from\_date AND edm.to\_date

AND CURDATE() BETWEEN es.from\_date AND es.to\_date

AND CURDATE() BETWEEN es\_manager.from\_date AND es\_manager.to\_date

AND CURDATE() BETWEEN ede.from\_date AND ede.to\_date

;

4. Show all employees that currently earn more than their managers.;

SELECT es.emp\_no AS 'Працівник, що заробляє більше менеджера', es.salary AS 'Зарплата працівника', es\_manager.salary AS 'Зарплата його менеджера'

FROM employees.salaries AS es

INNER JOIN employees.dept\_emp AS ede ON (es.emp\_no = ede.emp\_no)

INNER JOIN employees.dept\_manager AS edm ON (ede.dept\_no=edm.dept\_no)

INNER JOIN employees.salaries AS es\_manager ON (edm.emp\_no=es\_manager.emp\_no)

WHERE CURDATE() BETWEEN edm.from\_date AND edm.to\_date

AND CURDATE() BETWEEN es.from\_date AND es.to\_date

AND CURDATE() BETWEEN es\_manager.from\_date AND es\_manager.to\_date

AND es.salary > es\_manager.salary

AND CURDATE() BETWEEN ede.from\_date AND ede.to\_date

;

5. Show how many employees currently hold each title, sorted in descending order by the

number of employees.;

SELECT count(et.emp\_no), et.title

FROM employees.titles AS et

WHERE CURDATE() BETWEEN et.from\_date AND et.to\_date

GROUP BY et.title

ORDER BY COUNT(et.emp\_no) DESC;

6. Show full name of the all employees who were employed in more than one department.;

SELECT CONCAT(ee.first\_name, ' ', ee.last\_name) AS 'Fullname of employer', COUNT(ede.emp\_no) AS ' Quantity departments'

FROM employees.employees AS ee

INNER JOIN employees.dept\_emp AS ede ON (ee.emp\_no=ede.emp\_no)

GROUP BY ede.emp\_no

HAVING COUNT(ede.emp\_no)>1;

7.Show the average salary and maximum salary in thousands of dollars for every year;

SELECT YEAR(from\_date), ROUND(AVG(salary)/1000), ROUND(MAX(salary)/1000)

FROM salaries

group by YEAR(from\_date)

ORDER BY YEAR(from\_date);

8. Show how many employees were hired on weekends (Saturday + Sunday), split by

gender;

SELECT COUNT(hire\_date), ee.gender

FROM employees.employees AS ee

WHERE WEEKDAY(ee.hire\_date) = 6

OR WEEKDAY(ee.hire\_date) = 5

GROUP BY ee.gender;

9. /\*Fulfill the script below to achieve the following data:

Group all employees according to their age at January 1st, 1995 into four groups:

30 or younger, 31-40, 41-50 and older. Show average salary for each group and gender

(8 categories in total) Also add subtotals and grand total.\*/;

SELECT

(CASE

WHEN gender ='M'

THEN 'Mans'

ELSE 'Females'

END

) AS case\_gender,

(CASE

WHEN timestampdiff(year, ee.birth\_date, '1995-01-01') > 50

THEN '50 and more, '

WHEN timestampdiff(year, ee.birth\_date, '1995-01-01') > 40

THEN '41-50 '

WHEN timestampdiff(year, ee.birth\_date, '1995-01-01') > 30

THEN '31-40 '

ELSE '<30 '

END

) AS case\_age,ROUND(AVG(salary))

FROM employees.employees AS ee

INNER JOIN employees.salaries AS es ON(ee.emp\_no=es.emp\_no)

WHERE ee.hire\_date<'1995-01-01'

AND (SELECT MAX(to\_date)

FROM dept\_emp de

WHERE de.emp\_no = ee.emp\_no

GROUP BY de.emp\_no) > '1995-01-01'

GROUP BY case\_age, case\_gender

WITH ROLLUP

ORDER BY case\_age

;