-- Câu 1

select \* from( select sa.salary, em.\*, de.dept\_no, ROW\_NUMBER() OVER (PARTITION BY de.dept\_no ORDER BY sa.salary ASC) AS row\_num

FROM employees.employees as em

JOIN employees.salaries as sa on em.emp\_no = sa.emp\_no

join employees.dept\_emp as de on de.emp\_no = em.emp\_no

join employees.departments as dep on dep.dept\_no = de.dept\_no

WHEre year(em.hire\_date) = 1995

AND sa.from\_date <= '1996-07-25' AND sa.to\_date >= '1996-07-25') a

where row\_num = 1;

-- Câu 2

select avg(sa.salary), dep.dept\_name

FROM employees.employees as em

JOIN employees.salaries as sa on em.emp\_no = sa.emp\_no

join employees.dept\_emp as de on de.emp\_no = em.emp\_no

join employees.departments as dep on dep.dept\_no = de.dept\_no

where sa.from\_date <= '1996-07-25' AND sa.to\_date >= '1996-07-25'

group by dep.dept\_no

order by avg(sa.salary) DESC, dep.dept\_name;

-- câu 3

select em.\* , sum(sa.salary)

from employees.employees as em

join employees.titles as tit on em.emp\_no = tit.emp\_no

join employees.salaries as sa on sa.emp\_no = em.emp\_no

where em.emp\_no = '10005'

and tit.title = 'Staff';

-- câu 4:

select count(de.emp\_no)

from dept\_manager as dm

join employees as em on dm.emp\_no = em.emp\_no

join dept\_emp as de on de.dept\_no = dm.dept\_no

where first\_name = 'Margareta'

and last\_name = 'Markovitch'

and de.from\_date > dm.from\_date

and de.to\_date < dm.to\_date;

-- câu 5

select avg(sa.salary), t.title

from salaries as sa

join titles as t on t.emp\_no = sa.emp\_no

where sa.from\_date <= '1996-07-25' AND sa.to\_date >= '1996-07-25'

group by t.title;

-- Câu6:

SELECT SUM(

CASE

WHEN salary <= 40000 THEN salary \* 0.00

WHEN salary > 40000 AND salary <= 60000 THEN (40000 \* 0.00) + ((salary - 40000) \* 0.05)

WHEN salary > 60000 AND salary <= 90000 THEN (40000 \* 0.00) + (20000 \* 0.05) + ((salary - 60000) \* 0.1)

ELSE (40000 \* 0.00) + (20000 \* 0.05) + (30000 \* 0.1) + ((salary - 90000) \* 0.15)

END

) AS total\_tax

FROM salaries

WHERE from\_date <= '1996-07-25' AND to\_date >= '1996-07-25';

-- Câu 7: Tính lương phải trả cho từng phòng trong năm 1996

select dep.dept\_no, sum(sa.salary)

from salaries as sa

join employees as em on em.emp\_no = sa.emp\_no

join dept\_emp as demp on demp.emp\_no = em.emp\_no

join departments as dep on demp.dept\_no = dep.dept\_no

where year(sa.from\_date) = 1996 AND year(sa.to\_date) = 1996

group by dep.dept\_no;

-- Câu 8: Liệt kê các nhân viên nam chỉ làm một trong 2 phòng d003 và d002 (không liệt kê nhân

-- viên đã từng làm cả 2 phòng)

SELECT

em.\*

FROM

employees.employees AS em

JOIN employees.dept\_emp AS de ON em.emp\_no = de.emp\_no

JOIN employees.departments AS dep ON dep.dept\_no = de.dept\_no

WHERE

em.gender = 'M'

AND dep.dept\_no IN ('d002', 'd003')

GROUP BY

em.emp\_no, em.first\_name, em.last\_name, em.gender

HAVING

COUNT(DISTINCT dep.dept\_no) = 1;

-- Câu 9: Tìm xem tổng lương phải trả của mỗi phòng ban trong kỳ lương 25/07/1996 là bao

-- nhiêu và lọc những phòng ban trả tổng lương cao hơn 300k$

select dep.dept\_no, sum(sa.salary) as sum

from salaries as sa

join employees as em on em.emp\_no = sa.emp\_no

join dept\_emp as demp on demp.emp\_no = em.emp\_no

join departments as dep on demp.dept\_no = dep.dept\_no

where sa.from\_date <= '1996-07-25' AND sa.to\_date >= '1996-07-25'

group by dep.dept\_no

having sum >= 300000;

-- CÂu 11: Trong mỗi phòng ban, hãy tìm 5 người đang làm việc giữ chức vụ staff có lương cao nhất.

select sa.salary, de.dept\_no

from salaries as sa

join dept\_emp as de on sa.emp\_no = de.emp\_no

join titles as t on t.emp\_no = sa.emp\_no

join departments as dep on dep.dept\_no = de.dept\_no

where t.title = 'Staff'

group by (sa.salary, de.dept\_no)

order by salary DESC

limit 5;

SELECT d.dept\_no, e.first\_name, e.last\_name, s.salary

FROM (

SELECT de.dept\_no, e.emp\_no, ROW\_NUMBER() OVER (PARTITION BY de.dept\_no ORDER BY s.salary DESC) as row\_num

FROM dept\_emp de

JOIN employees e ON de.emp\_no = e.emp\_no

JOIN titles t ON e.emp\_no = t.emp\_no AND t.title = 'Staff'

JOIN salaries s ON e.emp\_no = s.emp\_no

) AS ranked

JOIN departments d ON ranked.dept\_no = d.dept\_no

JOIN employees e ON ranked.emp\_no = e.emp\_no

JOIN salaries s ON ranked.emp\_no = s.emp\_no

WHERE ranked.row\_num = 5

ORDER BY s.salary DESC;

-- Câu 12:

-- Cách 1

SELECT e.emp\_no, e.first\_name, e.last\_name,

d1.dept\_no AS dept\_no1, t1.title AS title1,

d2.dept\_no AS dept\_no2, t2.title AS title2

FROM employees e

JOIN dept\_emp de1 ON e.emp\_no = de1.emp\_no

JOIN titles t1 ON e.emp\_no = t1.emp\_no

JOIN departments d1 ON de1.dept\_no = d1.dept\_no

JOIN dept\_emp de2 ON e.emp\_no = de2.emp\_no AND de2.dept\_no != de1.dept\_no

JOIN titles t2 ON e.emp\_no = t2.emp\_no

JOIN departments d2 ON de2.dept\_no = d2.dept\_no

WHERE t1.title != t2.title

ORDER BY e.emp\_no;

-- Bài 2:

-- 2.1: Thăng chức cho nhân viên 10002 từ “Staff” lên “Senior Staff”. Lưu ý, phải dừng chức vụ hiện tại mới được chuyển chức vụ mới

-- Test: select em.\*, t.\* from employees em, titles t where t.emp\_no = em.emp\_no and em.emp\_no = '10002';

START TRANSACTION;

-- Bước 1: Hủy chức vụ hiện tại

UPDATE titles

SET to\_date = NOW() -- dừng chức vụ hiện tại

WHERE emp\_no = 10002 AND title = 'Staff' AND to\_date = '9999-01-01';

-- Bước 2: Thăng chức lên "Senior Staff"

INSERT INTO titles (emp\_no, title, from\_date, to\_date)

VALUES (10002, 'Senior Staff', NOW(), '9999-01-01');

COMMIT;

-- 2.2: Hãy xóa phòng ban Production cùng toàn bộ nhân viên của phòng này, cùng với tất cả các dữ liệu có liên quan.

Start transaction;

DELETE from departments where dept\_name = 'Production';

DELETE FROM dept\_emp WHERE dept\_no = 'd004';

DELETE FROM dept\_manager WHERE dept\_no = 'd004';

-- XÓa các dữ liệu liên quan

DELETE FROM titles WHERE emp\_no IN (SELECT emp\_no FROM dept\_emp WHERE dept\_no = 'd004');

DELETE FROM salaries WHERE emp\_no IN (SELECT emp\_no FROM dept\_emp WHERE dept\_no = 'd004');

COMMIT;

-- 2.3:

START TRANSACTION;

INSERT INTO departments (dept\_no, dept\_name)

VALUES ('d010', 'Bigdata & ML');

INSERT INTO dept\_manager (emp\_no, dept\_no, from\_date, to\_date)

VALUES (10173, 'd010', NOW(), '9999-01-01');

COMMIT;

-- Bài 3:

DELIMITER //

CREATE PROCEDURE GetEmployeeInfoAndTotalSalary(IN employee\_name VARCHAR(255))

BEGIN

-- 1: id, full name, gioi tinh

SELECT e.emp\_no, CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name, e.gender,

t.title, d.dept\_name

FROM employees e

JOIN titles t ON e.emp\_no = t.emp\_no AND t.to\_date = '9999-01-01'

JOIN dept\_emp de ON e.emp\_no = de.emp\_no AND de.to\_date = '9999-01-01'

JOIN departments d ON de.dept\_no = d.dept\_no

WHERE CONCAT(e.first\_name, ' ', e.last\_name) = employee\_name;

-- 2: tong luong

SELECT e.emp\_no, CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name,

SUM(s.salary) AS total\_salary

FROM employees e

JOIN salaries s ON e.emp\_no = s.emp\_no AND s.to\_date = '9999-01-01'

WHERE CONCAT(e.first\_name, ' ', e.last\_name) = employee\_name

GROUP BY e.emp\_no, CONCAT(e.first\_name, ' ', e.last\_name);

END //

DELIMITER ;

-- Call stored procerdure

CALL GetEmployeeInfoAndTotalSalary('Parto Bamford');

-- BÀI 4: Viết một Store Procedure để thuyên chuyển phòng ban cho một nhân viên nào đó, với chức

-- vụ mới (không chuyển lên làm quản lý). Đồng thời trả lại một kết quả bao gồm:

-- ra id, full name, giới tính, title (hay chức vụ), tên phòng ban

DELIMITER //

CREATE PROCEDURE TransferEmployeeToDepartment(

IN emp\_id INT,

IN new\_dept\_no CHAR(4),

IN new\_title VARCHAR(50)

)

BEGIN

-- Kết quả: Lấy ra id, full name, giới tính, title, tên phòng ban trước khi thuyên chuyển

SELECT e.emp\_no, CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name,

e.gender, t.title, d.dept\_name

FROM employees e

JOIN titles t ON e.emp\_no = t.emp\_no AND t.to\_date = '9999-01-01'

JOIN dept\_emp de ON e.emp\_no = de.emp\_no AND de.to\_date = '9999-01-01'

JOIN departments d ON de.dept\_no = d.dept\_no

WHERE e.emp\_no = emp\_id;

-- Thực hiện thuyên chuyển

UPDATE dept\_emp

SET dept\_no = new\_dept\_no

WHERE emp\_no = emp\_id AND to\_date = '9999-01-01';

-- Cập nhật chức vụ mới

UPDATE titles

SET to\_date = NOW()

WHERE emp\_no = emp\_id AND to\_date = '9999-01-01';

INSERT INTO titles (emp\_no, title, from\_date, to\_date)

VALUES (emp\_id, new\_title, NOW(), '9999-01-01');

-- Kết quả: Lấy ra id, full name, giới tính, title, tên phòng ban sau khi thuyên chuyển

SELECT e.emp\_no, CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name,

e.gender, t.title, d.dept\_name

FROM employees e

JOIN titles t ON e.emp\_no = t.emp\_no AND t.to\_date = '9999-01-01'

JOIN dept\_emp de ON e.emp\_no = de.emp\_no AND de.to\_date = '9999-01-01'

JOIN departments d ON de.dept\_no = d.dept\_no

WHERE e.emp\_no = emp\_id;

END //

DELIMITER ;

-- Chuyển Patricio từ title Engineer phòng Development sang phòng Sales và làm Senior Staff

CALL TransferEmployeeToDepartment(10012, 'd007', 'Senior Staff');

CALL TransferEmployeeToDepartment(10007, 'd004', 'StaffDT'); -- (chuyen tu phong d008 sang d004)

-- Test

SELECT e.emp\_no, CONCAT(e.first\_name, ' ', e.last\_name) AS full\_name,

e.gender, t.title, d.dept\_name, de.from\_date, de.to\_date, de.dept\_no

FROM employees e

JOIN titles t ON e.emp\_no = t.emp\_no

JOIN dept\_emp de ON e.emp\_no = de.emp\_no

JOIN departments d ON de.dept\_no = d.dept\_no

where d.dept\_name = 'Sales';

SELECT \* FROM employees.employees;

-- #################################################################-END ############################################################

select dept\_no, dept\_name, full\_name, salary

from (

select d.dept\_no, d.dept\_name, a.full\_name, b.salary, row\_number() over (partition by d.dept\_no order by b.salary asc) as row\_num

from departments d

join dept\_emp de on d.dept\_no = de.dept\_no

join (select e.emp\_no, concat(e.first\_name, " ", e.last\_name) as full\_name

from employees e

where year(e.hire\_date) >= 1995

) a on de.emp\_no = a.emp\_no

join (SELECT s.salary, s.emp\_no

from salaries s

where s.from\_date <= '1996-07-25' and s.to\_date > '1996-07-25'

) b on a.emp\_no = b.emp\_no

) c

where row\_num <= 1;