Đào Huy Hoàng_21110297

SET search_path TO hrschema;

-- Lab 03 – câu d (Bổ sung)

```
WITH managers AS(
SELECT e.employee_id AS id, e.salary, e.first_name, e.job_id,
        RANK() OVER(ORDER BY e.salary DESC) AS salary_rank
FROM employees e
WHERE e.employee_id
IN (SELECT DISTINCT(manager_id) FROM employees)
)
SELECT m.id AS manager_id, m.first_name, m.salary, j.job_title
FROM managers m
JOIN jobs j ON m.job_id = j.job_id
WHERE salary_rank < 5;
```

	manager_id integer	first_name character varying (255)	salary integer	job_title character varying (255)
1	100	Steven	24000	President
2	102	Lex	17000	Administration Vice President
3	101	Neena	17000	Administration Vice President
4	201	Michael	13000	Marketing Manager

--Lab 04

--Câu a

```
SELECT
        j.job_title,
        COUNT(e.employee_id) AS total_employees,
        ROUND(AVG(e.salary), 2) AS average_salary
```

```
FROM employees e
JOIN jobs j USING(job_id)
GROUP BY j.job_title;
```

	job_title character varying (255)	total_employees bigint	average_salary numeric
1	Accounting Manager	1	12000.00
2	Stock Clerk	1	2700.00
3	Human Resources Representa...	1	6500.00
4	Marketing Representative	1	6000.00
5	Purchasing Manager	1	11000.00
6	Sales Manager	2	13750.00
7	Administration Assistant	1	4400.00
8	Sales Representative	4	7550.00
9	Stock Manager	4	7650.00
10	Accountant	5	7920.00
11	Finance Manager	1	12000.00
12	Administration Vice President	2	17000.00
13	Programmer	5	5760.00
14	Marketing Manager	1	13000.00
15	Public Relations Representative	1	10000.00
16	President	1	24000.00
17	Public Accountant	1	8300.00
18	Shipping Clerk	2	3950.00
19	Purchasing Clerk	5	2780.00

```
--Câu b
SELECT
        first_name,
        last_name,
```

```
    salary,
    SUM(salary) OVER() AS total_salary
FROM employees;
```

	first_name character varying (255)	last_name character varying (255)	salary integer	total_salary bigint
1	Steven	King	24000	322400
2	Neena	Kochhar	17000	322400
3	Lex	De Haan	17000	322400
4	Alexander	Hunold	9000	322400
5	Bruce	Ernst	6000	322400
6	David	Austin	4800	322400
7	Valli	Pataballa	4800	322400
8	Diana	Lorentz	4200	322400
9	Nancy	Greenberg	12000	322400
10	Daniel	Faviet	9000	322400
11	John	Chen	8200	322400
12	Ismael	Sciarra	7700	322400
13	Jose Manuel	Urman	7800	322400
14	Luis	Popp	6900	322400
15	Den	Raphaely	11000	322400
16	Alexander	Khoo	3100	322400
17	Shelli	Baida	2900	322400
18	Sigal	Tobias	2800	322400
19	Guy	Himuro	2600	322400
20	Karen	Colmenares	2500	322400

21	Matthew	Weiss	8000	322400
22	Adam	Fripp	8200	322400
23	Payam	Kaufling	7900	322400
24	Shanta	Vollman	6500	322400
25	Irene	Mikkilineni	2700	322400
26	John	Russell	14000	322400
27	Karen	Partners	13500	322400
28	Jonathon	Taylor	8600	322400
29	Jack	Livingston	8400	322400
30	Kimberely	Grant	7000	322400
31	Charles	Johnson	6200	322400
32	Sarah	Bell	4000	322400
33	Britney	Everett	3900	322400
34	Jennifer	Whalen	4400	322400
35	Michael	Hartstein	13000	322400
36	Pat	Fay	6000	322400
37	Susan	Mavris	6500	322400
38	Hermann	Baer	10000	322400
39	Shelley	Higgins	12000	322400
40	William	Gietz	8300	322400

--Câu c

```
SELECT
        first_name,
        last_name,
        salary,
        ROUND(AVG(salary) OVER(), 2) AS average_salary,
        ROUND(salary - AVG(salary) OVER(), 2) AS salary_difference
FROM employees;
```

	first_name character varying (255)	last_name character varying (255)	salary integer	average_salary numeric	salary_difference numeric
1	Steven	King	24000	8060.00	15940.00
2	Neena	Kochhar	17000	8060.00	8940.00
3	Lex	De Haan	17000	8060.00	8940.00
4	Alexander	Hunold	9000	8060.00	940.00
5	Bruce	Ernst	6000	8060.00	-2060.00
6	David	Austin	4800	8060.00	-3260.00
7	Valli	Pataballa	4800	8060.00	-3260.00
8	Diana	Lorentz	4200	8060.00	-3860.00
9	Nancy	Greenberg	12000	8060.00	3940.00
10	Daniel	Faviet	9000	8060.00	940.00
11	John	Chen	8200	8060.00	140.00
12	Ismael	Sciarra	7700	8060.00	-360.00
13	Jose Manuel	Urman	7800	8060.00	-260.00
14	Luis	Popp	6900	8060.00	-1160.00
15	Den	Raphaely	11000	8060.00	2940.00
16	Alexander	Khoo	3100	8060.00	-4960.00
17	Shelli	Baida	2900	8060.00	-5160.00
18	Sigal	Tobias	2800	8060.00	-5260.00
19	Guy	Himuro	2600	8060.00	-5460.00
20	Karen	Colmenares	2500	8060.00	-5560.00
21	Matthew	Weiss	8000	8060.00	-60.00
22	Adam	Fripp	8200	8060.00	140.00
23	Payam	Kaufling	7900	8060.00	-160.00
24	Shanta	Vollman	6500	8060.00	-1560.00
25	Irene	Mikkilineni	2700	8060.00	-5360.00
26	John	Russell	14000	8060.00	5940.00
27	Karen	Partners	13500	8060.00	5440.00
28	Jonathon	Taylor	8600	8060.00	540.00
29	Jack	Livingston	8400	8060.00	340.00
30	Kimberely	Grant	7000	8060.00	-1060.00
31	Charles	Johnson	6200	8060.00	-1860.00
32	Sarah	Bell	4000	8060.00	-4060.00
33	Britney	Everett	3900	8060.00	-4160.00
34	Jennifer	Whalen	4400	8060.00	-3660.00
35	Michael	Hartstein	13000	8060.00	4940.00
36	Pat	Fay	6000	8060.00	-2060.00
37	Susan	Mavris	6500	8060.00	-1560.00
38	Hermann	Baer	10000	8060.00	1940.00
39	Shelley	Higgins	12000	8060.00	3940.00
40	William	Gietz	8300	8060.00	240.00

--Câu d

```
SELECT
    e.first_name,
    e.last_name,
    e.salary,
    ROUND(e.salary - AVG(e.salary) OVER(PARTITION BY
d.department_name), 2) AS salary_difference
FROM employees e
JOIN departments d USING(department_id)
WHERE d.department_name = 'IT';
```

	first_name character varying (255)	last_name character varying (255)	salary integer	salary_difference numeric
1	Alexander	Hunold	9000	3240.00
2	Bruce	Ernst	6000	240.00
3	David	Austin	4800	-960.00
4	Valli	Pataballa	4800	-960.00
5	Diana	Lorentz	4200	-1560.00

--Câu e

```
SELECT
    first_name,
    last_name,
    salary,
    COUNT(employee_id) OVER() AS total_employees
FROM employees
WHERE salary > 7000
ORDER BY salary DESC;
```

	first_name character varying (255)	last_name character varying (255)	salary integer	total_employees bigint
1	Steven	King	24000	21
2	Neena	Kochhar	17000	21
3	Lex	De Haan	17000	21
4	John	Russell	14000	21
5	Karen	Partners	13500	21
6	Michael	Hartstein	13000	21
7	Nancy	Greenberg	12000	21
8	Shelley	Higgins	12000	21
9	Den	Raphaely	11000	21
10	Hermann	Baer	10000	21
11	Alexander	Hunold	9000	21
12	Daniel	Faviet	9000	21
13	Jonathon	Taylor	8600	21
14	Jack	Livingston	8400	21
15	William	Gietz	8300	21
16	Adam	Fripp	8200	21
17	John	Chen	8200	21
18	Matthew	Weiss	8000	21
19	Payam	Kaufling	7900	21
20	Jose Manuel	Urman	7800	21
21	Ismael	Sciarra	7700	21

--Câu f

```
SELECT
        first_name,
        last_name,
        EXTRACT(MONTH FROM hire_date) AS hire_month,
        COUNT(employee_id) OVER(PARTITION BY EXTRACT(MONTH
FROM hire_date)) AS hire_count
FROM employees;
```

	first_name character varying (255)	last_name character varying (255)	hire_month numeric	hire_count bigint
1	Charles	Johnson	1	4
2	Karen	Partners	1	4
3	Lex	De Haan	1	4
4	Alexander	Hunold	1	4
5	Diana	Lorentz	2	4
6	Michael	Hartstein	2	4
7	Valli	Pataballa	2	4
8	Sarah	Bell	2	4
9	Britney	Everett	3	3
10	Jose Manuel	Urman	3	3
11	Jonathon	Taylor	3	3
12	Jack	Livingston	4	2
13	Adam	Fripp	4	2
14	Kimberely	Grant	5	4
15	Alexander	Khoo	5	4
16	Payam	Kaufling	5	4
17	Bruce	Ernst	5	4
18	Steven	King	6	6
19	David	Austin	6	6
20	Susan	Mavris	6	6

	first_name character varying (255)	last_name character varying (255)	hire_month numeric	hire_count bigint
20	Susan	Mavris	6	6
21	Hermann	Baer	6	6
22	Shelley	Higgins	6	6
23	William	Gietz	6	6
24	Matthew	Weiss	7	2
25	Sigal	Tobias	7	2
26	Daniel	Faviet	8	4
27	Nancy	Greenberg	8	4
28	Pat	Fay	8	4
29	Karen	Colmenares	8	4
30	Ismael	Sciarra	9	5
31	John	Chen	9	5
32	Jennifer	Whalen	9	5
33	Neena	Kochhar	9	5
34	Irene	Mikkilineni	9	5
35	Shanta	Vollman	10	2
36	John	Russell	10	2
37	Guy	Himuro	11	1
38	Den	Raphaely	12	3
39	Luis	Popp	12	3
40	Shelli	Baida	12	3

--Câu g

```
SELECT
        e.employee_id,
        e.first_name,
        e.last_name,
        EXTRACT(MONTH FROM e.hire_date) AS hire_month,
        e.manager_id,
        CONCAT(m.first_name, ' ', m.last_name) AS manager_name,
```

MAX(e.salary) OVER(PARTITION BY EXTRACT(MONTH FROM e.hire_date), e.manager_id)

FROM employees e

LEFT JOIN employees m ON e.manager_id = m.employee_id

ORDER BY e.manager_id, hire_month;

	employee_id [PK] integer	first_name character varying (255)	last_name character varying (255)	hire_month numeric	manager_id integer	manager_name text	max integer
1	146	Karen	Partners	1	100	Steven King	17000
2	102	Lex	De Haan	1	100	Steven King	17000
3	179	Charles	Johnson	1	100	Steven King	17000
4	201	Michael	Hartstein	2	100	Steven King	13000
5	176	Jonathon	Taylor	3	100	Steven King	8600
6	177	Jack	Livingston	4	100	Steven King	8400
7	121	Adam	Fripp	4	100	Steven King	8400
8	122	Payam	Kaufling	5	100	Steven King	7900
9	178	Kimberely	Grant	5	100	Steven King	7900
10	120	Matthew	Weiss	7	100	Steven King	8000
11	101	Neena	Kochhar	9	100	Steven King	17000
12	145	John	Russell	10	100	Steven King	14000
13	123	Shanta	Vollman	10	100	Steven King	14000
14	114	Den	Raphaely	12	100	Steven King	11000
15	203	Susan	Mavris	6	101	Neena Kochhar	12000
16	204	Hermann	Baer	6	101	Neena Kochhar	12000
17	205	Shelley	Higgins	6	101	Neena Kochhar	12000
18	108	Nancy	Greenberg	8	101	Neena Kochhar	12000
19	200	Jennifer	Whalen	9	101	Neena Kochhar	4400
20	103	Alexander	Hunold	1	102	Lex De Haan	9000
21	106	Valli	Pataballa	2	103	Alexander Hunold	4800
22	107	Diana	Lorentz	2	103	Alexander Hunold	4800
23	104	Bruce	Ernst	5	103	Alexander Hunold	6000
24	105	David	Austin	6	103	Alexander Hunold	4800
25	112	Jose Manuel	Urman	3	108	Nancy Greenberg	7800
26	109	Daniel	Faviet	8	108	Nancy Greenberg	9000
27	111	Ismael	Sciarra	9	108	Nancy Greenberg	8200
28	110	John	Chen	9	108	Nancy Greenberg	8200
29	113	Luis	Popp	12	108	Nancy Greenberg	6900
30	115	Alexander	Khoo	5	114	Den Raphaely	3100
31	117	Sigal	Tobias	7	114	Den Raphaely	2800
32	119	Karen	Colmenares	8	114	Den Raphaely	2500
33	118	Guy	Himuro	11	114	Den Raphaely	2600
34	116	Shelli	Baida	12	114	Den Raphaely	2900
35	126	Irene	Mikkilineni	9	120	Matthew Weiss	2700
36	192	Sarah	Bell	2	123	Shanta Vollman	4000
37	193	Britney	Everett	3	123	Shanta Vollman	3900
38	202	Pat	Fay	8	201	Michael Hartstein	6000
39	206	William	Gietz	6	205	Shelley Higgins	8300
40	100	Steven	King	6	[null]		24000

--Câu h

```
SELECT
        e.employee_id,
        j.job_title,
        d.department_name,
        COUNT(e.employee_id) OVER(PARTITION BY j.job_title) AS count_title,
        COUNT(e.employee_id) OVER(PARTITION BY d.department_name) AS count_department
FROM employees e
JOIN jobs j USING(job_id)
JOIN departments d USING(department_id);
```

	employee_id integer	job_title character varying (255)	department_name character varying (255)	count_title bigint	count_department bigint
1	111	Accountant	Finance	5	6
2	113	Accountant	Finance	5	6
3	112	Accountant	Finance	5	6
4	110	Accountant	Finance	5	6
5	109	Accountant	Finance	5	6
6	205	Accounting Manager	Accounting	1	2
7	200	Administration Assistant	Administration	1	1
8	102	Administration Vice President	Executive	2	3
9	101	Administration Vice President	Executive	2	3
10	108	Finance Manager	Finance	1	6
11	203	Human Resources Representa...	Human Resources	1	1
12	201	Marketing Manager	Marketing	1	2
13	202	Marketing Representative	Marketing	1	2
14	100	President	Executive	1	3
15	107	Programmer	IT	5	5
16	106	Programmer	IT	5	5
17	105	Programmer	IT	5	5
18	104	Programmer	IT	5	5
19	103	Programmer	IT	5	5
20	206	Public Accountant	Accounting	1	2

21	204	Public Relations Representative	Public Relations	1	1
22	116	Purchasing Clerk	Purchasing	5	6
23	117	Purchasing Clerk	Purchasing	5	6
24	119	Purchasing Clerk	Purchasing	5	6
25	118	Purchasing Clerk	Purchasing	5	6
26	115	Purchasing Clerk	Purchasing	5	6
27	114	Purchasing Manager	Purchasing	1	6
28	146	Sales Manager	Sales	2	6
29	145	Sales Manager	Sales	2	6
30	178	Sales Representative	Sales	4	6
31	179	Sales Representative	Sales	4	6
32	176	Sales Representative	Sales	4	6
33	177	Sales Representative	Sales	4	6
34	192	Shipping Clerk	Shipping	2	7
35	193	Shipping Clerk	Shipping	2	7
36	126	Stock Clerk	Shipping	1	7
37	123	Stock Manager	Shipping	4	7
38	122	Stock Manager	Shipping	4	7
39	121	Stock Manager	Shipping	4	7
40	120	Stock Manager	Shipping	4	7

--Câu i

```
SELECT
    e.employee_id,
    e.salary,
    d.department_name,
    ROUND(e.salary::numeric / (SUM(salary) OVER(PARTITION BY d.department_name)) * 100, 2) || ' %' AS ratio
FROM employees e
JOIN departments d USING(department_id)
ORDER BY d.department_name;
```

	employee_id integer	salary integer	department_name character varying (255)	ratio text
1	206	8300	Accounting	40.89 %
2	205	12000	Accounting	59.11 %
3	200	4400	Administration	100.00 %
4	102	17000	Executive	29.31 %
5	100	24000	Executive	41.38 %
6	101	17000	Executive	29.31 %
7	109	9000	Finance	17.44 %
8	111	7700	Finance	14.92 %
9	112	7800	Finance	15.12 %
10	110	8200	Finance	15.89 %
11	113	6900	Finance	13.37 %
12	108	12000	Finance	23.26 %
13	203	6500	Human Resources	100.00 %
14	107	4200	IT	14.58 %
15	103	9000	IT	31.25 %
16	104	6000	IT	20.83 %
17	105	4800	IT	16.67 %
18	106	4800	IT	16.67 %
19	202	6000	Marketing	31.58 %
20	201	13000	Marketing	68.42 %

21	204	10000	Public Relations	100.00 %
22	117	2800	Purchasing	11.24 %
23	116	2900	Purchasing	11.65 %
24	119	2500	Purchasing	10.04 %
25	114	11000	Purchasing	44.18 %
26	115	3100	Purchasing	12.45 %
27	118	2600	Purchasing	10.44 %
28	177	8400	Sales	14.56 %
29	176	8600	Sales	14.90 %
30	146	13500	Sales	23.40 %
31	145	14000	Sales	24.26 %
32	179	6200	Sales	10.75 %
33	178	7000	Sales	12.13 %
34	120	8000	Shipping	19.42 %
35	122	7900	Shipping	19.17 %
36	123	6500	Shipping	15.78 %
37	126	2700	Shipping	6.55 %
38	192	4000	Shipping	9.71 %
39	193	3900	Shipping	9.47 %
40	121	8200	Shipping	19.90 %

--Câu j

```
WITH ranked_employees AS (
SELECT
        employee_id,
        first_name || ' ' || last_name AS name,
        salary,
        ROW_NUMBER() OVER(ORDER BY salary) AS salary_rank
FROM employees
```

)

SELECT

employee_id,

name,

salary AS top_salary

FROM ranked_employees

WHERE salary_rank >= ROUND((1 - 0.2) * (SELECT COUNT(*) FROM ranked_employees));

	employee_id [PK] integer	name text	top_salary integer
1	114	Den Raphaely	11000
2	108	Nancy Greenberg	12000
3	205	Shelley Higgins	12000
4	201	Michael Hartstein	13000
5	146	Karen Partners	13500
6	145	John Russell	14000
7	101	Neena Kochhar	17000
8	102	Lex De Haan	17000
9	100	Steven King	24000

--Câu k

SELECT

city,

department_name,

COUNT(employee_id) AS employee_count,

ROUND(COUNT(employee_id) / SUM(COUNT(employee_id)) OVER() * 100, 2) || ' %' AS rate

FROM (

SELECT

```
        l.city,
        d.department_name,
        e.employee_id
    FROM employees e
    JOIN departments d USING(department_id)
    JOIN locations l USING(location_id)
    WHERE l.city = 'Seattle'
) AS subquery
GROUP BY city, department_name;
```

	city character varying (255)	department_name character varying (255)	employee_count bigint	rate text
1	Seattle	Accounting	2	11.11 %
2	Seattle	Administration	1	5.56 %
3	Seattle	Executive	3	16.67 %
4	Seattle	Finance	6	33.33 %
5	Seattle	Purchasing	6	33.33 %

```
--Câu l
SELECT
    e.first_name,
    e.last_name,
    d.department_name,
    salary,
    DENSE_RANK() OVER(ORDER BY salary DESC) AS salary_rank,
    NTILE(3) OVER(ORDER BY salary DESC) AS salary_group
FROM employees e
JOIN departments d USING(department_id);
```

	first_name character varying (255)	last_name character varying (255)	department_name character varying (255)	salary integer	salary_rank bigint	salary_group integer
1	Steven	King	Executive	24000	1	1
2	Neena	Kochhar	Executive	17000	2	1
3	Lex	De Haan	Executive	17000	2	1
4	John	Russell	Sales	14000	3	1
5	Karen	Partners	Sales	13500	4	1
6	Michael	Hartstein	Marketing	13000	5	1
7	Shelley	Higgins	Accounting	12000	6	1
8	Nancy	Greenberg	Finance	12000	6	1
9	Den	Raphaely	Purchasing	11000	7	1
10	Hermann	Baer	Public Relations	10000	8	1
11	Alexander	Hunold	IT	9000	9	1
12	Daniel	Faviet	Finance	9000	9	1
13	Jonathon	Taylor	Sales	8600	10	1
14	Jack	Livingston	Sales	8400	11	1
15	William	Gietz	Accounting	8300	12	2
16	John	Chen	Finance	8200	13	2
17	Adam	Fripp	Shipping	8200	13	2
18	Matthew	Weiss	Shipping	8000	14	2
19	Payam	Kaufling	Shipping	7900	15	2
20	Jose Manuel	Urman	Finance	7800	16	2
21	Ismael	Sciarra	Finance	7700	17	2
22	Kimberely	Grant	Sales	7000	18	2
23	Luis	Popp	Finance	6900	19	2
24	Susan	Mavris	Human Resources	6500	20	2
25	Shanta	Vollman	Shipping	6500	20	2
26	Charles	Johnson	Sales	6200	21	2
27	Bruce	Ernst	IT	6000	22	2
28	Pat	Fay	Marketing	6000	22	3
29	David	Austin	IT	4800	23	3
30	Valli	Pataballa	IT	4800	23	3
31	Jennifer	Whalen	Administration	4400	24	3
32	Diana	Lorentz	IT	4200	25	3
33	Sarah	Bell	Shipping	4000	26	3
34	Britney	Everett	Shipping	3900	27	3
35	Alexander	Khoo	Purchasing	3100	28	3
36	Shelli	Baida	Purchasing	2900	29	3
37	Sigal	Tobias	Purchasing	2800	30	3
38	Irene	Mikkilineni	Shipping	2700	31	3
39	Guy	Himuro	Purchasing	2600	32	3
40	Karen	Colmenares	Purchasing	2500	33	3

--Câu m

```
WITH country_stats AS (
SELECT
	c.country_name,
	COUNT(employee_id) AS num_emp,
	ROUND(SUM(e.salary), 2) AS sum_sal,
	ROUND(AVG(e.salary), 2) AS avg_sal
FROM employees e
JOIN departments d USING(department_id)
JOIN locations l USING(location_id)
JOIN countries c USING(country_id)
GROUP BY c.country_name
)
SELECT
	*,
	cs.avg_sal - LAG(cs.avg_sal, -1) OVER(ORDER BY cs.avg_sal DESC)
AS diff
FROM country_stats cs
ORDER BY avg_sal DESC;
```

	country_name character varying (255)	num_emp bigint	sum_sal numeric	avg_sal numeric	diff numeric
1	Germany	1	10000.00	10000.00	500.00
2	Canada	2	19000.00	9500.00	328.57
3	United Kingdom	7	64200.00	9171.43	1531.43
4	United States of America	30	229200.00	7640.00	[null]