Đào Huy Hoàng – 21110297

```
SET search_path TO hrschema;
--Câu a
SELECT
    d.department_name,
    c.country_name,
    COUNT(e.employee_id) AS num_employee,
    MIN(e.salary),
    MAX(e.salary),
    AVG(e.salary),
    SUM(e.salary)
FROM employees e
JOIN departments d USING(department_id)
JOIN locations l USING(location_id)
JOIN countries c USING(country_id)
GROUP BY d.department_name, c.country_name, d.department_id
ORDER BY d.department_id ASC;
```

OUTPUT

department_name	country_name	num_employee	min	max	avg	sum
Administration	United States of America	1	4400	4400	4400	4400
Marketing	Canada	2	6000	13000	9500	19000
Purchasing	United States of America	6	2500	11000	4150	24900
Human Resources	United Kingdom	1	6500	6500	6500	6500
Shipping	United States of America	7	2700	8200	5885.714286	41200
IT	United States of America	5	4200	9000	5760	28800
Public Relations	Germany	1	10000	10000	10000	10000
Sales	United Kingdom	6	6200	14000	9616.666667	57700
Executive	United States of America	3	17000	24000	19333.33333	58000
Finance	United States of America	6	6900	12000	8600	51600
Accounting	United States of America	2	8300	12000	10150	20300

--Câu b

```
SELECT
        d.department_name,
        c.country_name,
        COUNT(e.employee_id),
        SUM(e.salary)
FROM employees e
JOIN departments d USING(department_id)
JOIN locations l USING(location_id)
JOIN countries c USING(country_id)
JOIN regions r USING(region_id)
WHERE r.region_name = 'Americas'
GROUP BY d.department_name, c.country_name
HAVING SUM(e.salary) > 30000;
```

OUTPUT

department_name	country_name	count	sum
Executive	United States of America	3	58000
Finance	United States of America	6	51600
Shipping	United States of America	7	41200

--Câu c

```
SELECT
        e.employee_id,
        CONCAT(e.first_name, ' ',e.last_name) AS name
FROM employees AS e
JOIN departments AS d USING(department_id)
JOIN locations AS l USING(location_id)
```

```
WHERE EXTRACT(MONTH FROM hire_date) != 6
AND l.city NOT IN ('London');
```

OUTPUT

employee_id	name
200	Jennifer Whalen
202	Pat Fay
201	Michael Hartstein
119	Karen Colmenares
118	Guy Himuro
117	Sigal Tobias
116	Shelli Baida
115	Alexander Khoo
114	Den Raphaely
193	Britney Everett
192	Sarah Bell
126	Irene Mikkilineni
123	Shanta Vollman
122	Payam Kaufling
121	Adam Fripp
120	Matthew Weiss
107	Diana Lorentz
106	Valli Pataballa
104	Bruce Ernst
103	Alexander Hunold
179	Charles Johnson
178	Kimberely Grant
177	Jack Livingston
176	Jonathon Taylor
146	Karen Partners
145	John Russell
102	Lex De Haan
101	Neena Kochhar
113	Luis Popp
112	Jose Manuel Urman
111	Ismael Sciarra
110	John Chen
109	Daniel Faviet
108	Nancy Greenberg

--Câu d

```
WITH managers AS(
SELECT e.employee_id AS id, e.salary, e.first_name, e.job_id
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
```

OUTPUT

manager_id	first_name	salary	job_title
205	Shelley	12000	Accounting Manager
100	Steven	24000	President
102	Lex	17000	Administration Vice President
101	Neena	17000	Administration Vice President
108	Nancy	12000	Finance Manager
103	Alexander	9000	Programmer
201	Michael	13000	Marketing Manager
114	Den	11000	Purchasing Manager
123	Shanta	6500	Stock Manager
120	Matthew	8000	Stock Manager

--Câu e

```
WITH managers AS(
        SELECT e.employee_id AS id, e.salary, e.first_name, e.last_name,
e.department_id
        FROM employees e
        WHERE e.employee_id
        IN (SELECT DISTINCT(manager_id) FROM employees)
),
managers_in_USA AS(
```

```
        SELECT m.id
        FROM managers AS m
        JOIN departments AS d USING(department_id)
        JOIN locations AS l USING(location_id)
        JOIN countries AS c USING(country_id)
        WHERE c.country_name IN ('United States of America')
),
avg_each_group AS(
   SELECT m.id, AVG(e.salary) AS avg_salary
   FROM employees e
   JOIN managers_in_USA m ON e.manager_id = m.id
   GROUP BY m.id
)
SELECT
        m.id AS id_manager,
        CONCAT(e.first_name, ' ',e.last_name) AS employee_name,
        e.salary
FROM employees AS e
JOIN managers_in_USA AS m ON e.manager_id = m.id
JOIN avg_each_group AS a ON e.manager_id = a.id
WHERE e.salary > a.avg_salary;
```

OUTPUT

id_manager	employee_name	salary
100	Neena Kochhar	17000
100	Lex De Haan	17000
103	Bruce Ernst	6000
101	Nancy Greenberg	12000
108	Daniel Faviet	9000
108	John Chen	8200
100	Den Raphaely	11000
114	Alexander Khoo	3100
114	Shelli Baida	2900
114	Sigal Tobias	2800
100	John Russell	14000
100	Karen Partners	13500
123	Sarah Bell	4000
100	Michael Hartstein	13000
101	Hermann Baer	10000
101	Shelley Higgins	12000

--Câu f

```
WITH RECURSIVE manager_table AS (
    SELECT
        0 AS level,
        CAST(first_name || ' ' || last_name AS VARCHAR(255)) AS path,
        CASE
            WHEN manager_id IS NULL THEN ''
            ELSE first_name || ' ' || last_name
        END AS manager_name,
        first_name || ' ' || last_name AS employee_name,
        manager_id,
        employee_id
    FROM employees
```

```
        WHERE manager_id IS NULL

    UNION ALL

        SELECT
            mt.level + 1 AS level,
            CAST(mt.path || ' -> ' || e.first_name || ' ' || e.last_name AS
VARCHAR(255)),
            mt.employee_name AS manager_name,
            e.first_name || ' ' || e.last_name,
            e.manager_id,
            e.employee_id
        FROM employees e
        JOIN manager_table mt ON e.manager_id = mt.employee_id
)
SELECT * FROM manager_table;
```

OUTPUT

level	path	manager_name	employee_name	manager_id	employee_id
0	Steven King		Steven King	NULL	100
1	Steven King -> Neena Kochhar	Steven King	Neena Kochhar	100	101
1	Steven King -> Lex De Haan	Steven King	Lex De Haan	100	102
1	Steven King -> Den Raphaely	Steven King	Den Raphaely	100	114
1	Steven King -> Matthew Weiss	Steven King	Matthew Weiss	100	120
1	Steven King -> Adam Fripp	Steven King	Adam Fripp	100	121
1	Steven King -> Payam Kaufling	Steven King	Payam Kaufling	100	122
1	Steven King -> Shanta Vollman	Steven King	Shanta Vollman	100	123
1	Steven King -> John Russell	Steven King	John Russell	100	145
1	Steven King -> Karen Partners	Steven King	Karen Partners	100	146
1	Steven King -> Jonathon Taylor	Steven King	Jonathon Taylor	100	176
1	Steven King -> Jack Livingston	Steven King	Jack Livingston	100	177
1	Steven King -> Kimberely Grant	Steven King	Kimberely Grant	100	178
1	Steven King -> Charles Johnson	Steven King	Charles Johnson	100	179
1	Steven King -> Michael Hartstein	Steven King	Michael Hartstein	100	201
2	Steven King -> Lex De Haan -> Alexander Hunold	Lex De Haan	Alexander Hunold	102	103
2	Steven King -> Neena Kochhar -> Nancy Greenberg	Neena Kochhar	Nancy Greenberg	101	108
2	Steven King -> Den Raphaely -> Alexander Khoo	Den Raphaely	Alexander Khoo	114	115
2	Steven King -> Den Raphaely -> Shelli Baida	Den Raphaely	Shelli Baida	114	116
2	Steven King -> Den Raphaely -> Sigal Tobias	Den Raphaely	Sigal Tobias	114	117
2	Steven King -> Den Raphaely -> Guy Himuro	Den Raphaely	Guy Himuro	114	118
2	Steven King -> Den Raphaely -> Karen Colmenares	Den Raphaely	Karen Colmenares	114	119
2	Steven King -> Matthew Weiss -> Irene Mikkilineni	Matthew Weiss	Irene Mikkilineni	120	126
2	Steven King -> Shanta Vollman -> Sarah Bell	Shanta Vollman	Sarah Bell	123	192
2	Steven King -> Shanta Vollman -> Britney Everett	Shanta Vollman	Britney Everett	123	193
2	Steven King -> Neena Kochhar -> Jennifer Whalen	Neena Kochhar	Jennifer Whalen	101	200
2	Steven King -> Michael Hartstein -> Pat Fay	Michael Hartstein	Pat Fay	201	202
2	Steven King -> Neena Kochhar -> Susan Mavris	Neena Kochhar	Susan Mavris	101	203
2	Steven King -> Neena Kochhar -> Hermann Baer	Neena Kochhar	Hermann Baer	101	204
2	Steven King -> Neena Kochhar -> Shelley Higgins	Neena Kochhar	Shelley Higgins	101	205
3	Steven King -> Lex De Haan -> Alexander Hunold -> Bruce Ernst	Alexander Hunold	Bruce Ernst	103	104
3	Steven King -> Lex De Haan -> Alexander Hunold -> David Austin	Alexander Hunold	David Austin	103	105
3	Steven King -> Lex De Haan -> Alexander Hunold -> Valli Pataballa	Alexander Hunold	Valli Pataballa	103	106
3	Steven King -> Lex De Haan -> Alexander Hunold -> Diana Lorentz	Alexander Hunold	Diana Lorentz	103	107
3	Steven King -> Neena Kochhar -> Nancy Greenberg -> Daniel Faviet	Nancy Greenberg	Daniel Faviet	108	109
3	Steven King -> Neena Kochhar -> Nancy Greenberg -> John Chen	Nancy Greenberg	John Chen	108	110
3	Steven King -> Neena Kochhar -> Nancy Greenberg -> Ismael Sciarra	Nancy Greenberg	Ismael Sciarra	108	111
3	Steven King -> Neena Kochhar -> Nancy Greenberg -> Jose Manuel Urman	Nancy Greenberg	Jose Manuel Urman	108	112
3	Steven King -> Neena Kochhar -> Nancy Greenberg -> Luis Popp	Nancy Greenberg	Luis Popp	108	113
3	Steven King -> Neena Kochhar -> Shelley Higgins -> William Gietz	Shelley Higgins	William Gietz	205	206

--Câu Bonus

```
WITH RECURSIVE region_table AS (
    SELECT
        0 AS level,
        CAST(region_name AS VARCHAR(255)) AS location,
        CAST(region_id AS VARCHAR(5)) AS id
    FROM regions

    UNION ALL
```

```
    SELECT
        rt.level + 1 AS level,
        CAST(rt.location || ' -> ' || c.country_name AS VARCHAR(255)),
        CAST(c.country_id AS VARCHAR(5)) AS id
    FROM region_table rt
    JOIN countries c ON CAST(rt.id AS VARCHAR(5)) = CAST(c.region_id AS
VARCHAR(5))


),
location_table AS (
    SELECT
        rt.level + 2 AS level,
        CAST(rt.location || ' -> ' || c.country_name || ' -> ' || l.city AS
VARCHAR(255)),
        CAST(l.location_id AS VARCHAR(5)) AS id
    FROM region_table rt
    JOIN countries c ON CAST(rt.id AS VARCHAR(5)) = CAST(c.region_id AS
VARCHAR(5))
    JOIN locations l ON CAST(c.country_id AS VARCHAR(5)) = l.country_id
)
SELECT * FROM region_table
UNION ALL
SELECT * FROM location_table;
```

OUTPUT

level	location	id
0	Europe	1
0	Americas	2
0	Asia	3
0	Middle East and Africa	4
1	Europe -> United Kingdom	UK
1	Europe -> Netherlands	NL
1	Europe -> Italy	IT
1	Europe -> France	FR
1	Europe -> Denmark	DK
1	Europe -> Germany	DE
1	Europe -> Switzerland	CH
1	Europe -> Belgium	BE
1	Americas -> United States of America	US
1	Americas -> Mexico	MX
1	Americas -> Canada	CA
1	Americas -> Brazil	BR
1	Americas -> Argentina	AR
1	Asia -> Singapore	SG
1	Asia -> Japan	JP
1	Asia -> India	IN
1	Asia -> HongKong	HK
1	Asia -> China	CN
1	Asia -> Australia	AU
1	Middle East and Africa -> Zimbabwe	ZW
1	Middle East and Africa -> Zambia	ZM
1	Middle East and Africa -> Nigeria	NG
1	Middle East and Africa -> Kuwait	KW
1	Middle East and Africa -> Israel	IL
1	Middle East and Africa -> Egypt	EG
2	Europe -> United Kingdom -> Oxford	2500
2	Europe -> United Kingdom -> London	2400
2	Europe -> Germany -> Munich	2700
2	Americas -> United States of America -> Seattle	1700
2	Americas -> United States of America -> South San Francisco	1500
2	Americas -> United States of America -> Southlake	1400
2	Americas -> Canada -> Toronto	1800