

Advanced SQL Online

Section B1-B2: Set A

1. Find employee id and hire date of all employees except those employees who are hired on the first hiring date or on the last hiring date in each department. You must use the set operation. (4)

EMPLOYEE_ID	HIRE_DATE
100	17-JUN-03
...	...
160	15-Dec-05

```
SELECT employee_id, hire_date
FROM employees
MINUS
(
    SELECT employee_id, hire_date
    FROM employees e1
    WHERE NOT EXISTS (
        SELECT *
        FROM employees e2
        WHERE e2.department_id = e1.department_id
        AND e2.hire_date > e1.hire_date
    ) OR NOT EXISTS (
        SELECT *
        FROM employees e2
        WHERE e2.department_id = e1.department_id
        AND e2.hire_date < e1.hire_date
    )
);
```

2. For each employee, show his full name, department name and full name of his manager. If an employee doesn't have a manager, print NULL as manager name. (4)

EMPLOYEE_NAME	DEPARTMENT_NAME	MANAGER_NAME
Michael Hartstein	Marketing	Steven King
...
Anthony Cabrio	Shipping	Adam Fripp

```
SELECT
    (e.first_name || ' ' || e.last_name) AS employee_name,
    d.department_name,
    (m.first_name || ' ' || m.last_name) AS manager_name
FROM employees e
JOIN departments d USING (department_id)
LEFT JOIN employees m ON (e.manager_id = m.employee_id);
```

3. Find the employees that are managed by the topmost manager. The topmost manager is the (4)

employee who is managed by none. Print the full name, department name, salary and hire date of those employees.

FULL_NAME	DEPARTMENT_NAME	SALARY	HIRE_DATE
Michael Hartstein	Marketing	13000	17-FEB-04
...
Lex De Haan	Executive	17000	13-JAN-01

```

SELECT
    (e.first_name || ' ' || e.last_name) AS full_name,
    d.department_name,
    e.salary,
    e.hire_date
FROM employees e
    JOIN departments d USING (department_id)
WHERE e.manager_id = (
    SELECT employee_id
    FROM employees
    WHERE manager_id IS NULL
);

```

4. Find the employees that get at most the average salary of the employees under his manager. (4)
Print his full name, salary and the average salary of the employees under his manager.

FULL_NAME	SALARY	AVG_SALARY
Alexander Hunold	9000	9000
...
Randall Perkins	2500	2950

```

SELECT
    (e.first_name || ' ' || e.last_name) AS full_name,
    e.salary,
    m.avg_salary
FROM employees e
    JOIN (
        SELECT manager_id, AVG(salary) AS avg_salary
        FROM employees
        GROUP BY manager_id
    ) m USING (manager_id)
WHERE e.salary <= m.avg_salary;

```

5. Rank the departments by their amount of expenditure in ascending manner (lowest expenditure gets rank 1). Order it by rank. [Note: expenditure = sum of salary of the employees in the department] (4)

RANK	DEPARTMENT_ID	EXPENDITURE
------	---------------	-------------

1	10	4400
...
12	80	304500

```

SELECT
    1+COUNT(t2.expenditure) AS rank,
    t1.department_id,
    t1.expenditure
FROM (
    SELECT
        department_id,
        SUM(salary) AS expenditure
    FROM employees
    GROUP BY department_id
) t1
LEFT JOIN (
    SELECT
        department_id,
        SUM(salary) AS expenditure
    FROM employees
    GROUP BY department_id
) t2 ON (t1.expenditure > t2.expenditure)
GROUP BY t1.department_id, t1.expenditure
ORDER BY rank;

```

Section B1-B2: Set B

1. Find employee id and salary of all employees except those employees who get minimum or maximum salary in each department. You must use the set operation. (4)

EMPLOYEE_ID	SALARY
104	6000
...	...
161	7000

```
SELECT employee_id, salary
FROM employees
MINUS
(
    SELECT employee_id, salary
    FROM employees e1
    WHERE NOT EXISTS (
        SELECT *
        FROM employees e2
        WHERE e2.department_id = e1.department_id
            AND e2.salary > e1.salary
    ) OR NOT EXISTS (
        SELECT *
        FROM employees e2
        WHERE e2.department_id = e1.department_id
            AND e2.salary < e1.salary
    )
);
```

2. For each employee, show his full name, job title and full name of his manager. If an employee doesn't have a manager, print NULL as manager name. (4)

EMPLOYEE_NAME	JOB_TITLE	MANAGER_NAME
Lex De Haan	Administration Vice President	Steven King
...
James Marlow	Stock Clerk	Adam Fripp

```
SELECT
    (e.first_name || ' ' || e.last_name) AS employee_name,
    j.job_title,
    (m.first_name || ' ' || m.last_name) AS manager_name
FROM employees e
JOIN jobs j USING (job_id)
LEFT JOIN employees m ON (e.manager_id = m.employee_id);
```

3. Find the employees that are not managed by the topmost manager. The topmost manager is the employee who is managed by none. Print the full name, department name, salary and hire date of those employees. (4)

FULL_NAME	DEPARTMENT_NAME	SALARY	HIRE_DATE
Jennifer Whalen	Administration	4400	17-SEP-03
...
Alexander Hunold	IT	9000	3-Jan-06

```

SELECT
    (e.first_name || ' ' || e.last_name) AS full_name,
    d.department_name,
    e.salary,
    e.hire_date
FROM employees e
    JOIN departments d USING (department_id)
WHERE e.manager_id <> (
    SELECT employee_id
    FROM employees
    WHERE manager_id IS NULL
);

```

4. Find the employees that get at least the average salary of the employees under his manager. (4)

FULL_NAME	SALARY	AVG_SALARY
Neena Kochhar	17000	11100
...
Shelley Higgins	12008	8983.2

```

SELECT
    (e.first_name || ' ' || e.last_name) AS full_name,
    e.salary,
    m.avg_salary
FROM employees e
    JOIN (
        SELECT manager_id, AVG(salary) AS avg_salary
        FROM employees
        GROUP BY manager_id
    ) m USING (manager_id)
WHERE e.salary >= m.avg_salary;

```

5. Rank the departments by their amount of expenditure in descending manner (highest expenditure gets rank 1). Order it by rank. [Note: expenditure = sum of salary of the employees in the department] (4)

RANK	DEPARTMENT_ID	EXPENDITURE
1	80	304500
...
12	10	4400

```

SELECT
    1+COUNT(t2.expenditure) AS rank,
    t1.department_id,
    t1.expenditure
FROM (
    SELECT
        department_id,
        SUM(salary) AS expenditure
    FROM employees
    GROUP BY department_id
) t1
LEFT JOIN (
    SELECT
        department_id,
        SUM(salary) AS expenditure
    FROM employees
    GROUP BY department_id
) t2 ON (t1.expenditure < t2.expenditure)
GROUP BY t1.department_id, t1.expenditure
ORDER BY rank;

```

Section A1-A2: Set A

1. For each job, count the number of managers. [Note: You are not allowed to use subquery.] (4)

JOB_TITLE	MANAGER_COUNT
Accounting Manager	1
...	...
Marketing Manager	1

```

SELECT j.job_title, COUNT(*) AS manager_count
FROM hr.employees m
      JOIN hr.employees e ON (e.manager_id = m.employee_id)
      JOIN hr.jobs j ON (m.job_id = j.job_id)
GROUP BY j.job_title;

```

2. For each department, find the three most junior employees (It is not necessary for an employee to be serving currently in this case). If there is a draw, print all of them (hence, there could be more than three employees printed for some department). Print department name, full name and hire date of employee. Ensure that the same departments are put in consecutively. [Note: You are not allowed to use subquery.] (4)

DEPARTMENT_NAME	EMPLOYEE_NAME	HIRE_DATE
Accounting	Shelley Higgins	07-JUN-02
...
Shipping	Trenna Rajs	17-OCT-03

```

SELECT
    d.department_name,
    (e.first_name || ' ' || e.last_name) AS employee_name,
    e.hire_date
FROM hr.departments d
      JOIN hr.employees e
      ON (e.department_id = d.department_id)
      LEFT JOIN hr.employees f
      ON (f.department_id = e.department_id
          AND f.hire_date < e.hire_date)
GROUP BY d.department_name, e.first_name, e.last_name,
         e.hire_date
HAVING COUNT(f.employee_id) < 3
ORDER BY d.department_name;

```

3. Find the employees whose managers come from the same region as them. Print the full name and region name of the employee. (4)

JOB_ID EMPLOYEE_NAME	MANAGER_COUNT REGION_NAME
----------------------	---------------------------

Peter Tucker	Europe
...	...
Ki Gee	Americas

```

SELECT
    (e.first_name || ' ' || e.last_name) AS employee_name,
    r.region_name
FROM (
    SELECT e.first_name, e.last_name, e.manager_id,
    c.region_id
    FROM hr.employees e
        JOIN hr.departments d USING (department_id)
        JOIN hr.locations l USING (location_id)
        JOIN hr.countries c USING (country_id)
    ) e
JOIN (
    SELECT e.employee_id, c.region_id
    FROM hr.employees e
        JOIN hr.departments d USING (department_id)
        JOIN hr.locations l USING (location_id)
        JOIN hr.countries c USING (country_id)
    ) m
ON (e.manager_id = m.employee_id AND e.region_id =
m.region_id)
JOIN hr.regions r ON (e.region_id = r.region_id);

```

4. Find the employees who are senior to at least half of the employees in his job (All the employees are currently serving in this case). (4)

EMPLOYEE_NAME	JOB_ID	HIRE_DATE
Lex De Haan	AD_VP	13-JAN-01
...
Kevin Feeney	SH_CLERK	23-MAY-06

```

SELECT
    (e.first_name || ' ' || e.last_name) AS employee_name,
    e.job_id,
    e.hire_date
FROM hr.employees e
WHERE (
    SELECT COUNT(*)
    FROM hr.employees e2
    WHERE e2.job_id = e.job_id AND e2.hire_date > e.hire_date
    ) >= (
    SELECT COUNT(*)/2
    FROM hr.employees e2

```



```

WHERE e2.job_id = e.job_id
);

```

5. Rank the jobs by number of managers.

(4)

RANK	JOB_ID	MANAGER_COUNT
1	SA_REP	30
...
10	AC_ACCOUNT	1

```

SELECT 1+COUNT(t2.job_id) AS rank, t1.job_id,
t1.manager_count
FROM
(
    SELECT e.job_id, COUNT(e.employee_id) AS manager_count
    FROM hr.employees e
    GROUP BY e.job_id
) t1
LEFT JOIN
(
    SELECT e.job_id, COUNT(e.employee_id) AS manager_count
    FROM hr.employees e
    GROUP BY e.job_id
) t2
ON (t1.manager_count < t2.manager_count)
GROUP BY t1.job_id, t1.manager_count
ORDER BY rank;

```

Section A1-A2: Set B

1. For each department, count the number of managers. [Note: You are not allowed to use subquery.] (4)

DEPARTMENT_NAME	MANAGER_COUNT
Sales	30
...	...
Accounting	1

```

SELECT d.department_name, COUNT(*) AS manager_count
FROM hr.employees m
      JOIN hr.employees e ON (e.manager_id = m.employee_id)
      JOIN hr.departments d ON (m.department_id =
d.department_id)
GROUP BY d.department_name;

```

2. For each department, find the three most senior employees (It is not necessary for an employee to be serving currently in this case). If there is a draw, print all of them (hence, there could be more than three employees printed for some department). Print department name, full name and hire date of employee. Ensure that the same departments are put in consecutively. [Note: You are not allowed to use subquery.] (4)

DEPARTMENT_NAME	EMPLOYEE_NAME	HIRE_DATE
Accounting	Shelley Higgins	07-JUN-02
...
Shipping	Steven Markle	08-MAR-08

```

SELECT
    d.department_name,
    (e.first_name || ' ' || e.last_name) AS employee_name,
    e.hire_date
FROM hr.departments d
      JOIN hr.employees e
      ON (e.department_id = d.department_id)
      LEFT JOIN hr.employees f
      ON (f.department_id = e.department_id
          AND f.hire_date > e.hire_date)
GROUP BY d.department_name, e.first_name, e.last_name,
e.hire_date
HAVING COUNT(f.employee_id) < 3
ORDER BY d.department_name;

```

3. Find the employees that are managed by an employee who is not of his own country. Print the full name and country name of both the employee and his manager. (4)

EMPLOYEE_NAME	COUNTRY_NAME	MANAGER_NAME	COUNTRY_NAME
John Russell	United Kingdom	Steven King	United States of America
...
Hermann Baer	Germany	Neena Kochhar	United States of America

```

SELECT
    (e.first_name || ' ' || e.last_name) AS employee_name,
    e.country_name,
    (m.first_name || ' ' || m.last_name) AS manager_name,
    m.country_name
FROM (
    SELECT e.first_name, e.last_name, e.manager_id,
    c.country_name
    FROM hr.employees e
        JOIN hr.departments d USING (department_id)
        JOIN hr.locations l USING (location_id)
        JOIN hr.countries c USING (country_id)
    ) e
JOIN (
    SELECT e.first_name, e.last_name, e.employee_id,
    c.country_name
    FROM hr.employees e
        JOIN hr.departments d USING (department_id)
        JOIN hr.locations l USING (location_id)
        JOIN hr.countries c USING (country_id)
    ) m
ON (e.manager_id = m.employee_id AND e.country_name <>
m.country_name);

```

4. Find the employees who are junior to at most half of the employees in his job (All the employees are currently serving in this case). (4)

EMPLOYEE_NAME	JOB_ID	HIRE_DATE
Lex De Haan	AD_VP	13-JAN-01
...
Kevin Feeney	SH_CLERK	23-MAY-06

```

SELECT
    (e.first_name || ' ' || e.last_name) AS employee_name,
    e.job_id,
    e.hire_date
FROM hr.employees e
WHERE (

```

```

SELECT COUNT(*)
FROM hr.employees e2
WHERE e2.job_id = e.job_id AND e2.hire_date < e.hire_date
) <= (
SELECT COUNT(*)/2
FROM hr.employees e2
WHERE e2.job_id = e.job_id
);

```

5. Rank the departments by number of managers.

(4)

RANK	DEPARTMENT_ID	MANAGER_COUNT
1	50	45
...
9	10	1

```

SELECT 1+COUNT(t2.department_id) AS rank, t1.department_id,
t1.manager_count
FROM
(
SELECT e.department_id, COUNT(e.employee_id) AS
manager_count
FROM hr.employees e
GROUP BY e.department_id
) t1
LEFT JOIN
(
SELECT e.department_id, COUNT(e.employee_id) AS
manager_count
FROM hr.employees e
GROUP BY e.department_id
) t2
ON (t1.manager_count < t2.manager_count)
GROUP BY t1.department_id, t1.manager_count
ORDER BY rank;

```