**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.

```
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.

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

```
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. Print his full name, salary and the average salary of the employees under his manager.

```
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]

```
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
```

**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.

```
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.

```
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.

```
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.

```
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]

```
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
```