

Here are more complex interview questions involving nested subqueries:

**53. Retrieve the departments where the total salary expenditure exceeds the average total salary expenditure across all departments.**

**Answer:**

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING SUM(salary) > (
    SELECT AVG(total_salary)
    FROM (
        SELECT department_id, SUM(salary) as total_salary
        FROM employees
        GROUP BY department_id
    ) AS subquery
);
```

**54. Find the employee with the third highest salary without using the LIMIT clause.**

**Answer:**

```
SELECT name, salary
FROM employees e1
WHERE 2 = (
    SELECT COUNT(DISTINCT e2.salary)
    FROM employees e2
    WHERE e2.salary > e1.salary
);
```

**55. Identify departments that have less than the company-wide median number of employees.**

**Answer:**

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING COUNT(id) < (
    SELECT AVG(employee_count)
    FROM (
        SELECT department_id, COUNT(id) as employee_count
        FROM employees
        GROUP BY department_id
    ) AS subquery
);
```

**56. Get the most common job title among employees who earn above the company average.**

**Answer:**

```
SELECT job_title
FROM employees
WHERE salary > (SELECT AVG(salary) FROM employees)
GROUP BY job_title
ORDER BY COUNT(*) DESC
LIMIT 1;
```

**57. Identify employees who earn more than the average salary in both their department and the company.**

**Answer:**

```
SELECT id, name, salary
FROM employees e1
WHERE salary > (
    SELECT AVG(salary)
    FROM employees
    WHERE department_id = e1.department_id
)
AND salary > (
    SELECT AVG(salary)
    FROM employees
);
```

**58. Retrieve the month (in numbers) with the highest total sales from a table of daily sales.**

**Answer:**

```
SELECT MONTH(date) as sales_month
FROM sales
GROUP BY MONTH(date)
ORDER BY SUM(amount) DESC
LIMIT 1;
```

**59. Get the department that has the maximum difference between the highest and lowest salaries.**

**Answer:**

```
SELECT department_id, (MAX(salary) - MIN(salary)) as salary_difference
FROM employees
GROUP BY department_id
HAVING salary_difference = (
    SELECT MAX(max_salary - min_salary)
    FROM (
        SELECT department_id, MAX(salary) as max_salary, MIN(salary) as min_salary
        FROM employees
        GROUP BY department_id
    ) AS subquery
);
```

**60. Find the employee who earns the median salary in each department.**

**Answer:**

```
SELECT e1.department_id, e1.name, e1.salary
FROM employees e1
WHERE (
    SELECT COUNT(*)
    FROM employees e2
    WHERE e2.department_id = e1.department_id AND e2.salary <= e1.salary
) = (
    SELECT COUNT(*)
    FROM employees e3
    WHERE e3.department_id = e1.department_id AND e3.salary >= e1.salary
);
```

**61. Retrieve employees who earn more than their respective department's median salary.**

**Answer:**

```
SELECT e1.name, e1.salary, e1.department_id
FROM employees e1
WHERE e1.salary > (
    SELECT AVG(salary)
    FROM (
        SELECT salary
        FROM employees e2
        WHERE e2.department_id = e1.department_id
        ORDER BY salary
        LIMIT 2 - (SELECT COUNT(*) FROM employees e3 WHERE e3.department_id = e1.department_id) MOD 2
        OFFSET (SELECT (COUNT(*) - 1) / 2 FROM employees e4 WHERE e4.department_id = e1.department_id)
    ) AS median_subquery
);
```

**62. Identify the departments where the minimum salary is greater than the maximum salary of at least one other department.**

**Answer:**

```
SELECT DISTINCT e1.department_id
FROM employees e1
WHERE e1.salary = (
    SELECT MIN(salary)
    FROM employees
    WHERE department_id = e1.department_id
)
AND e1.salary > ANY (
    SELECT MAX(salary)
    FROM employees
    GROUP BY department_id
);
```

**63. Find employees whose salary ranks in the top 3 within their department.**

**Answer:**

```
SELECT e1.name, e1.salary, e1.department_id
FROM employees e1
WHERE (
    SELECT COUNT(DISTINCT e2.salary)
    FROM employees e2
    WHERE e2.department_id = e1.department_id AND e2.salary > e1.salary
) < 3;
```

**64. Identify the department with the most diverse salary distribution, i.e., the largest difference between the highest and lowest salaries.**

**Answer:**

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING (MAX(salary) - MIN(salary)) = (
    SELECT MAX(salary_range)
    FROM (
        SELECT (MAX(salary) - MIN(salary)) as salary_range
        FROM employees
        GROUP BY department_id
    ) AS subquery
);
```

**65. Retrieve the employees who do not have the lowest salary in their department but earn less than the department average.**

**Answer:**

```
SELECT e1.name, e1.salary, e1.department_id
FROM employees e1
WHERE e1.salary NOT IN (
    SELECT MIN(e2.salary)
    FROM employees e2
    WHERE e2.department_id = e1.department_id
)
AND e1.salary < (
    SELECT AVG(e3.salary)
    FROM employees e3
    WHERE e3.department_id = e1.department_id
);
```

**66. Determine which departments have an average salary close to the company's median salary. Assume 'close' means a difference of less than 1000.**

**Answer:**

```

SELECT department_id
FROM employees
GROUP BY department_id
HAVING ABS(AVG(salary) - (
    SELECT AVG(median_salary)
    FROM (
        SELECT salary AS median_salary
        FROM employees
        ORDER BY salary
        LIMIT 2 - (SELECT COUNT(*) FROM employees) MOD 2
        OFFSET (SELECT (COUNT(*) - 1) / 2 FROM employees)
    ) AS median_subquery
)) < 1000;

```

**67. Find the departments where the total number of employees is above the company's average.**

**Answer:**

```

SELECT department_id
FROM employees
GROUP BY department_id
HAVING COUNT(id) > (
    SELECT AVG(employee_count)
    FROM (
        SELECT COUNT(id) AS employee_count
        FROM employees
        GROUP BY department_id
    ) AS avg_subquery
);

```

**68. Identify employees who earn more than the second highest earner in their respective department.**

**Answer:**

```

SELECT e1.name, e1.salary, e1.department_id
FROM employees e1
WHERE e1.salary > (
    SELECT MAX(e2.salary)
    FROM employees e2
    WHERE e2.department_id = e1.department_id AND e2.salary < (
        SELECT MAX(e3.salary)
        FROM employees e3
        WHERE e3.department_id = e1.department_id
    )
);

```

**69. Find the departments where the top earner makes at least twice as much as the second top earner.**

**Answer:**

```
SELECT department_id
FROM employees
GROUP BY department_id
HAVING MAX(salary) >= 2 * (
    SELECT MAX(salary)
    FROM employees e2
    WHERE e2.department_id = employees.department_id AND salary < MAX(employees.salary)
);
```

**70. Retrieve the employees who have been in the company for longer than the average tenure of their respective department managers.**

**Answer:**

```
SELECT e1.name, e1.join_date
FROM employees e1
WHERE DATEDIFF(CURDATE(), e1.join_date) > (
    SELECT AVG(DATEDIFF(CURDATE(), e2.join_date))
    FROM employees e2
    WHERE e2.id IN (
        SELECT manager_id
        FROM employees
        WHERE department_id = e1.department_id
    )
);
```

**71. Identify the department with the smallest gap between the lowest and average salary.**

**Answer:**

```

SELECT department_id
FROM employees
GROUP BY department_id
HAVING (AVG(salary) - MIN(salary)) = (
    SELECT MIN(gap)
    FROM (
        SELECT (AVG(salary) - MIN(salary)) AS gap
        FROM employees
        GROUP BY department_id
    ) AS gap_subquery
);

```

**\*\*72. Identify the employees who earn below the average salary of their peers who joined in the same year.\*\***

**\*\*Answer\*\*:**

```

```sql
SELECT e1.name, e1.salary, YEAR(e1.join_date) AS join_year
FROM employees e1
WHERE e1.salary < (
    SELECT AVG(e2.salary)
    FROM employees e2
    WHERE YEAR(e2.join_date) = YEAR(e1.join_date)
);

```

**73. Retrieve the employee who has the closest salary to their department's median but isn't the median earner.**

**Answer:**

```

SELECT e1.name, e1.salary
FROM employees e1
WHERE e1.department_id IN (
    SELECT department_id
    FROM employees
)
AND e1.salary <> (
    SELECT AVG(median_salary)
    FROM (
        SELECT salary AS median_salary
        FROM employees e2
        WHERE e2.department_id = e1.department_id
        ORDER BY salary
        LIMIT 2 - (SELECT COUNT(*) FROM employees e3 WHERE e3.department_id = e1.department_id) MOD 2
        OFFSET (SELECT (COUNT(*) - 1) / 2 FROM employees e4 WHERE e4.department_id = e1.department_id)
    ) AS median_subquery
)
ORDER BY ABS(e1.salary - (
    SELECT AVG(median_salary)
    FROM (
        SELECT salary AS median_salary
        FROM employees e5
        WHERE e5.department_id = e1.department_id
        ORDER BY salary
        LIMIT 2 - (SELECT COUNT(*) FROM employees e6 WHERE e6.department_id = e1.department_id) MOD 2
        OFFSET (SELECT (COUNT(*) - 1) / 2 FROM employees e7 WHERE e7.department_id = e1.department_id)
    ) AS median_subquery2
))
LIMIT 1;

```

**74. Determine the departments whose average tenure (time since joining) is greater than the company average.**

**Answer:**

```

SELECT department_id
FROM employees
GROUP BY department_id
HAVING AVG(DATEDIFF(CURDATE(), join_date)) > (
    SELECT AVG(DATEDIFF(CURDATE(), join_date))
    FROM employees
);

```

**75. Identify departments where more than half of the employees earn above the company's median salary.**

**Answer:**



```

SELECT e1.department_id
FROM employees e1
WHERE e1.salary > (
    SELECT AVG(median_salary)
    FROM (
        SELECT salary AS median_salary
        FROM employees
        ORDER BY salary
        LIMIT 2 - (SELECT COUNT(*) FROM employees) MOD 2
        OFFSET (SELECT (COUNT(*) - 1) / 2 FROM employees)
    ) AS median_subquery
)
GROUP BY e1.department_id
HAVING COUNT(e1.id) > 0.5 * (
    SELECT COUNT(*)
    FROM employees e2
    WHERE e2.department_id = e1.department_id
);

```

**76. Find employees who earn a salary in the top 3 of their department but are not in the top 10 company-wide.**

**Answer:**

```

SELECT e1.name, e1.salary, e1.department_id
FROM employees e1
WHERE (
    SELECT COUNT(DISTINCT e2.salary)
    FROM employees e2
    WHERE e2.department_id = e1.department_id AND e2.salary > e1.salary
) < 3
AND e1.salary NOT IN (
    SELECT DISTINCT salary
    FROM employees
    ORDER BY salary DESC
    LIMIT 10
);

```

**77. Identify employees whose salary is above the average salary of the two departments with the highest average salaries.**

**Answer:**

```

SELECT e1.name, e1.salary
FROM employees e1
WHERE e1.salary > (
    SELECT AVG(department_avg)
    FROM (
        SELECT department_id, AVG(salary) AS department_avg
        FROM employees
        GROUP BY department_id
        ORDER BY department_avg DESC
        LIMIT 2
    ) AS top_department_subquery
);

```

**78. Find employees who have a manager earning less than the lowest salary in their department.**

**Answer:**

```

SELECT e1.name, e1.salary
FROM employees e1
JOIN employees e2 ON e1.manager_id = e2.id
WHERE e2.salary < (
    SELECT MIN(e3.salary)
    FROM employees e3
    WHERE e3.department_id = e1.department_id
);

```

**79. Identify the department with the least difference between the top earner and the average salary of the department.**

**Answer:**

```

SELECT department_id
FROM employees
GROUP BY department_id
HAVING (MAX(salary) - AVG(salary)) = (
    SELECT MIN(top_minus_avg)
    FROM (
        SELECT (MAX(salary) - AVG(salary)) AS top_minus_avg
        FROM employees
        GROUP BY department_id
    ) AS difference_subquery
);

```

**80. Retrieve the employees who have the same rank (in terms of salary) in their department as they do in the company overall.**

**Answer:**

```

SELECT e1.name, e1.salary
FROM employees e1
WHERE (
    SELECT COUNT(DISTINCT e2.salary)
    FROM employees e2
    WHERE e2.department_id = e1.department_id AND e2.salary > e1.salary
) = (
    SELECT COUNT(DISTINCT e3.salary)
    FROM employees e3
    WHERE e3.salary > e1.salary
);

```

**81. Determine the departments where the third-highest earner makes more than double the department's average salary.**

**Answer:**

```

SELECT department_id
FROM employees e1
WHERE (
    SELECT DISTINCT salary
    FROM (
        SELECT salary
        FROM employees e2
        WHERE e2.department_id = e1.department_id
        ORDER BY e2.salary DESC
        LIMIT 3
    ) AS third_top_salary_subquery
    ORDER BY salary
    LIMIT 1 OFFSET 2
) > 2 * (
    SELECT AVG(e3.salary)
    FROM employees e3
    WHERE e3.department_id = e1.department_id
)
GROUP BY department_id;

```

**82. Find employees who have more direct reports (subordinates) than their manager.**

**Answer:**

```
SELECT e1.name
FROM employees e1
WHERE (
    SELECT COUNT(*)
    FROM employees e2
    WHERE e2.manager_id = e1.id
) > (
    SELECT COUNT(*)
    FROM employees e3
    WHERE e3.manager_id = e1.manager_id
);
```