🚀 Top 6 Challenging SQL Questions Asked at Google! 💻

1️⃣ Find the Median of Salaries 💰
How to calculate the median salary in a table?

```sql
SELECT AVG(salary) AS median_salary
FROM (
    SELECT salary
    FROM employees
    ORDER BY salary
    LIMIT 2 - (SELECT COUNT(*) FROM employees) % 2
    OFFSET (SELECT (COUNT(*) - 1) / 2 FROM employees)
) AS median;
```

2️⃣ Employees with the Highest Salary in Each Department 📊
Query to find employees with the highest salary department-wise:

```sql
SELECT department, employee, salary
FROM (
    SELECT department, employee, salary, RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS rnk
    FROM employees
) AS ranked
WHERE rnk = 1;
```

3 Find All Active Users for the Last 7 Days 📅
How to find distinct users who logged in over the past 7 days:

```
SELECT COUNT(DISTINCT user_id) AS active_users
FROM user_logs
WHERE action = 'login'
  AND log_date >= CURRENT_DATE - INTERVAL '7 days';
```

4 Nth Highest Salary 🥇
Find the Nth highest salary in a table:

```
WITH salary_ranking AS (
    SELECT salary, DENSE_RANK() OVER (ORDER BY salary DESC)
AS rnk
    FROM employees
)
SELECT salary
FROM salary_ranking
WHERE rnk = N;
```

5 Find Consecutive Increases in Salary 📈
Identify employees whose salary has increased for 3 consecutive years:

```sql
WITH ranked_salaries AS (
   SELECT employee_id, year, salary,
     LAG(salary, 1) OVER (PARTITION BY employee_id ORDER BY
year) AS prev_salary_1,
     LAG(salary, 2) OVER (PARTITION BY employee_id ORDER BY
year) AS prev_salary_2
   FROM employee_salaries
)
SELECT employee_id
FROM ranked_salaries
WHERE salary > prev_salary_1 AND prev_salary_1 > prev_salary_2;
```

6 Average Time Between Two Actions ⏱️
Calculate the average time (in minutes) between two actions (login and logout):

```sql
WITH login_logout AS (
   SELECT user_id,
     MIN(CASE WHEN action = 'login' THEN timestamp END) AS
login_time,
     MIN(CASE WHEN action = 'logout' THEN timestamp END) AS
logout_time
   FROM user_actions
   WHERE action IN ('login', 'logout')
   GROUP BY user_id
)
SELECT AVG(EXTRACT(EPOCH FROM (logout_time - login_time))) /
60 AS avg_minutes
FROM login_logout;
```