Problem

Write a solution to find:

1/ Write a solution to find managers with at least five direct reports.

2/ Return the result table in any order. The result format is in the following example.

Example accepted result:

```
+------+
| name |
+======+
| John |
+------+
```

Example ‘Employee’ table

```
+-----+-------+------------+-----------+
| id  | name  | department | managerId |
+=====+=======+============+===========+
| 101 | John  | A          | null      |
+-----+-------+------------+-----------+
| 102 | Dan   | A          | 101       |
+-----+-------+------------+-----------+
| 103 | James | A          | 101       |
+-----+-------+------------+-----------+
| 104 | Amy   | A          | 101       |
+-----+-------+------------+-----------+
| 105 | Anne  | A          | 101       |
+-----+-------+------------+-----------+
| 106 | Ron   | B          | 101       |
+-----+-------+------------+-----------+
```

cte solution

```
WITH
link_employee_to_manager AS (
 SELECT
    e1.id,
    e1.name,
    e1.department,
    e1.managerId,
    e2.name AS manager_name
 FROM
    Employee e1 JOIN Employee e2
 ON
    e1.managerId = e2.id
),
```

```
managers_names_with_five_or_more_reports AS (
 SELECT
    manager_name AS name
 FROM
    employee_to_manager
 GROUP BY
    manager_name
 HAVING
    COUNT(name) >= 5
)

SELECT * FROM managers_names_with_five_or_more_reports;
```

Output

id	name	department	managerId	manager_name
101	John	A	null	null
102	Dan	A	101	John
103	James	A	101	John
104	Amy	A	101	John
105	Anne	A	101	John
106	Ron	B	101	John

name
John

Success

subquery solution

```
SELECT E1.name
FROM Employee E1
JOIN (
    SELECT
        managerId,
        COUNT(*) AS directReports
     FROM Employee
     GROUP BY managerId
     HAVING COUNT(*) >= 5
) E2 ON E1.id = E2.managerId;
```

Output

```
+------------+---------------+
| manager_id | directReports |
+============+===============+
| 101        | 5             |
+------------+---------------+
```

```
+------+
| name |
+======+
| John |
+------+
```

Success

Observations

- Each solution is working properly and return the answer we wanted (even though they may not have the same performance -- this was not the point of the post).

- To me, one solution is **clearly better: using cte**

  - Imagine working with a harder problem, **could you re-read your code 6 months after writing it and say to yourself “yes, I totally understand what’s going on” ?**

  - The **subquery code took me way longer to write than the cte one:**
    - We, as humans, are not wired to think of the execution order of what we write at scale - we should make things easier for ourselves by creating small cte blocks who act like variables in other programming languages.

  - Using a cte is **verbose**. However it does **not** affect the performance of your SQL, so why bother with poor syntax and difficult sql comprehension ?

What do you think !
Answer below ⤵