AMAZON

INTERVIEW

Amazon Asked This... SQL Interview Question

EMPLOYEE DATA

ID	NAME	MANAGER
1	Alice (CEO)	NULL
2	Bob (VP)	1
4	Dan (Mgr)	2
5	Eve (Dev)	4

?

Q: FIND EVE'S CHAIN

Output Expected:

Eve → Dan

Dan → Bob

Bob → Alice

RECURSIVE PATTERN

FOLLOW FOR MORE SQL PATTERNS

Anandnarayanan S

MISTAKES

The Common Error

Why a simple JOIN fails

❌

1. Standard Self-Join

Querying `e1.manager_id = e2.id` only retrieves the immediate manager.

↓

📉

2. Limited Depth

It cannot traverse multiple levels (VP, CEO) dynamically.

↓

🔄

3. Infinite Loops

Without proper termination, circular references cause timeouts.

FOLLOW FOR MORE SQL PATTERNS

Anandnarayanan S

🔁

CONCEPT

Two Directions

The Join direction determines the path

⬆️ UPWARD

Find Managers Above

- Start with Employee
- Climb UP to CEO

```
JOIN on manager_id
```

⬇️ DOWNWARD

Find Reports Below

- Start with Manager
- Go DOWN to reports

```
JOIN on employee_id
```

🔑 Key Insight

Most people fail simply by flipping the JOIN condition!

FOLLOW FOR MORE SQL PATTERNS

Anandnarayanan S

SCENARIO 1

Upward Recursion

Finding Eve's Managers (Eve → Alice)

STANDARD SQL

```sql
WITH RECURSIVE hierarchy AS (
    -- Base Case: Start with Eve
    SELECT id, name, manager_id, 1 AS lvl
    FROM employees WHERE id = 5

    UNION ALL

    -- Recursive: Climb UP to managers
    SELECT e.id, e.name, e.manager_id, h.lvl + 1
    FROM employees e
    JOIN hierarchy h ON e.id = h.manager_id
    -- Match Employee TO Manager ↑
)
SELECT * FROM hierarchy;
```

FOLLOW FOR MORE SQL PATTERNS
Anandnarayanan S

SCENARIO 2

Downward Recursion

Finding Bob's Team (Bob → Eve)

STANDARD SQL

```sql
WITH RECURSIVE hierarchy AS (
    -- Base Case: Start with Bob
    SELECT id, name, manager_id, 1 AS lvl
    FROM employees WHERE id = 2

    UNION ALL

    -- Recursive: Go DOWN to reports
    SELECT e.id, e.name, e.manager_id, h.lvl + 1
    FROM employees e
    JOIN hierarchy h ON e.manager_id = h.id
    -- Match Manager TO Employee ↓
)
SELECT * FROM hierarchy;
```

FOLLOW FOR MORE SQL PATTERNS
Anandnarayanan S

SPEED

Why Use Recursive?

Performance Comparison

⚠️ Multiple Joins

- N queries for N levels
- High network overhead

~500ms+

🚀 Recursive CTE

- Single query execution
- Engine optimized

~50ms

⚡ 10X Faster

Recursive CTEs reduce data transfer and utilize database engine optimizations for hierarchical data.

FOLLOW FOR MORE SQL PATTERNS
Anandnarayanan S

MASTERY

7/7

Your Next Step...

- ✓ Master Hierarchies in SQL
- ✓ Upward: Join `id = manager_id`
- ✓ Downward: Join `manager_id = id`
- ✓ Don't mix up the direction!

REPOST TO HELP OTHERS LEARN THIS TOO!

FOLLOW FOR MORE SQL PATTERNS

Anandnarayanan S