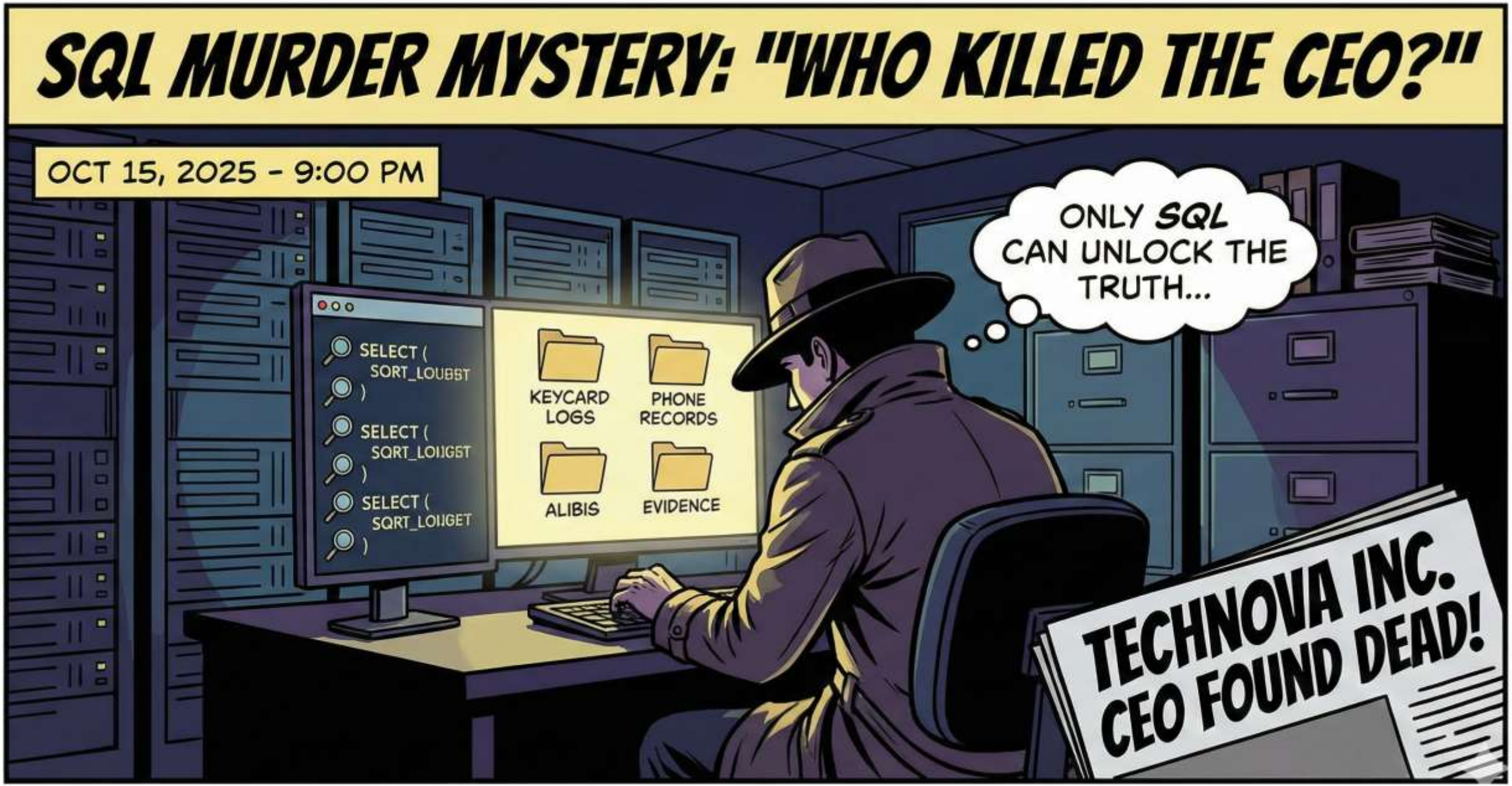

SQL MURDER MYSTERY: "WHO KILLED THE CEO?"
OCT 15, 2025 - 9:00 PM
SELECT (
SORT_LOUDST
)
SELECT (
SQRT_LOUGST
)
SELECT (
SQRT_LOUGET
)
KEYCARD LOGS
PHONE RECORDS
ALIBIS
EVIDENCE
ONLY SQL CAN UNLOCK THE TRUTH...
TECHNOVA INC. CEO FOUND DEAD!


Ankit, your insights are brilliant! You've earned a significant salary hike. Remember, DATA is the new fuel!
Just fuel? I build the engine! Why him?


THE PLAN IS SET.
HE WON'T SEE IT
COMING.
CEO OFFICE


OH MY GOD... IT'S THE CEO!
OCTOBER 15TH, 2025
21:00
IT'S DONE. NOW TO DISAPPEAR.


MR. TRIPATHI, THE KEYCARD LOGS AND AND SERVER DATA ARE CRITICAL. WE NEED YOU TO SOLVE THIS.
Ankit


UNDERSTOOD. DATA DOESN'T LIE. I'LL FIND THE KILLER.


```sql
SELECT kl.employee_id,
e.name, e.department,
e.role, kl.room,
kl.entry_time,
kl.exit_time
 FROM keycard_logs kl
 INNER JOIN employees e
 ON kl.employee_id = e.employee_id
WHERE kl.room = "CEO Office";
```

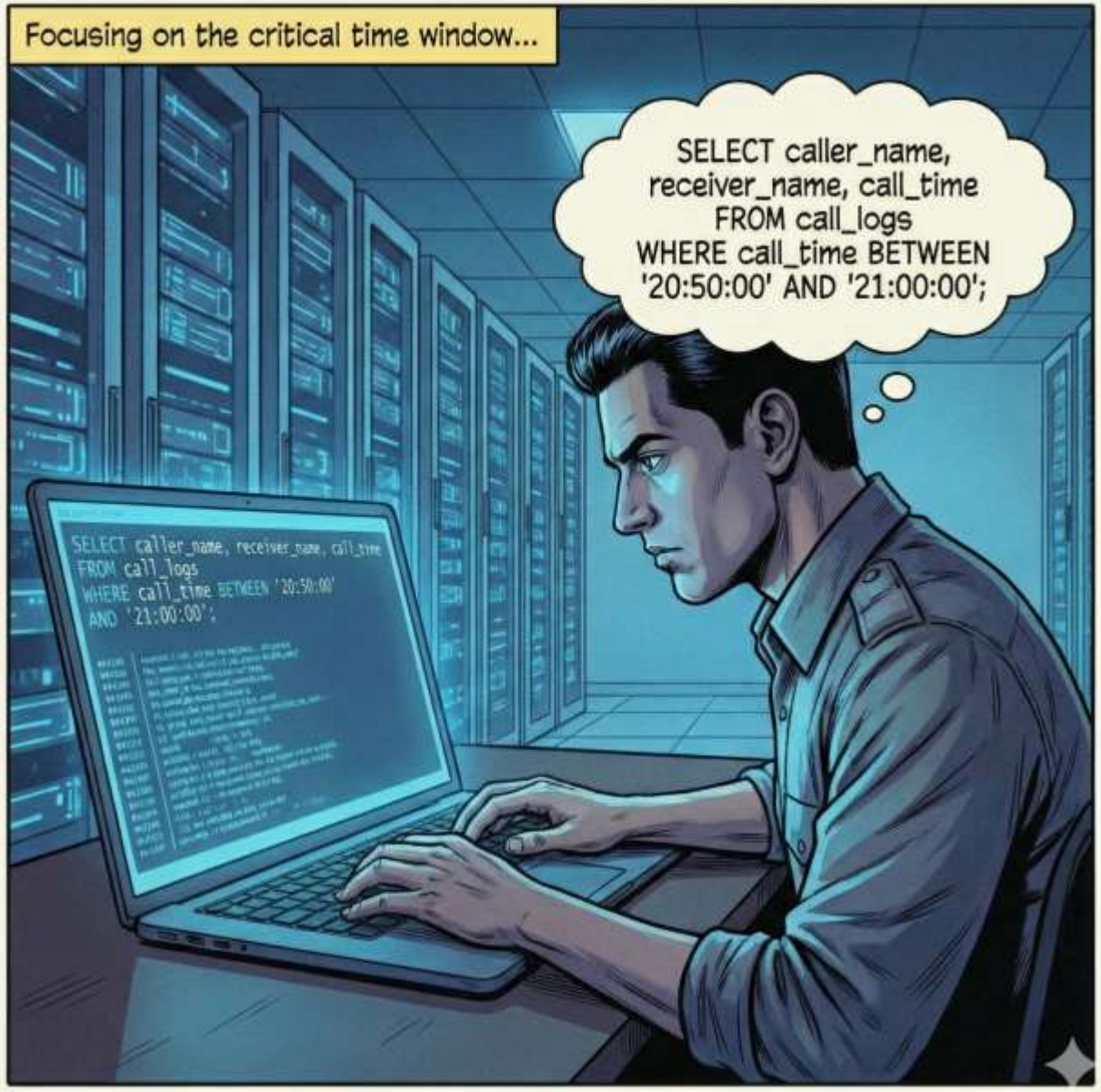
David Kumar has been identified as the prime suspect after access logs revealed highly suspicious entry and exit timestamps at the CEO's office.

employee_id	name	department	role	room	entry_time	exit_time
4	David Kumar	Engineering	DevOps Engineer	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00


```sql
SELECT
a.employee_id,
a.claimed_location,
a.claim_time,
k.room,
k.entry_time,
k.exit_time
FROM alibis a
INNER JOIN keycard_logs k ON a.employee_id = k.employee_ic
WHERE a.claimed_location <> k.room;
```

employee_id	claimed_location	claim_time	room	entry_time	exit_time
4	Server Room	2025-10-15 20:50:00	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00

The alibi for David Kumar (Employee_ID 4) crumbled under scrutiny. Although he stated he was working in the Server Room, location records place him directly at the crime scene—the CEO's office—during the critical window of the murder.




```sql
SELECT
    c.call_id,
    c.caller_id,
    caller.name AS caller_name,
    c.receiver_id,
    receiver.name AS receiver_name,
    c.call_time,
    c.duration_sec,
    DATE_ADD(c.call_time, INTERVAL c.duration_sec SECOND) AS call_end_time
FROM calls AS c
JOIN employees AS caller
    ON c.caller_id = caller.employee_id
JOIN employees AS receiver
    ON c.receiver_id = receiver.employee_id
WHERE c.call_time <= '2025-10-15 21:00:00'
  AND DATE_ADD(c.call_time, INTERVAL c.duration_sec SECOND) >= '2025-10-15 20:50:00';
```

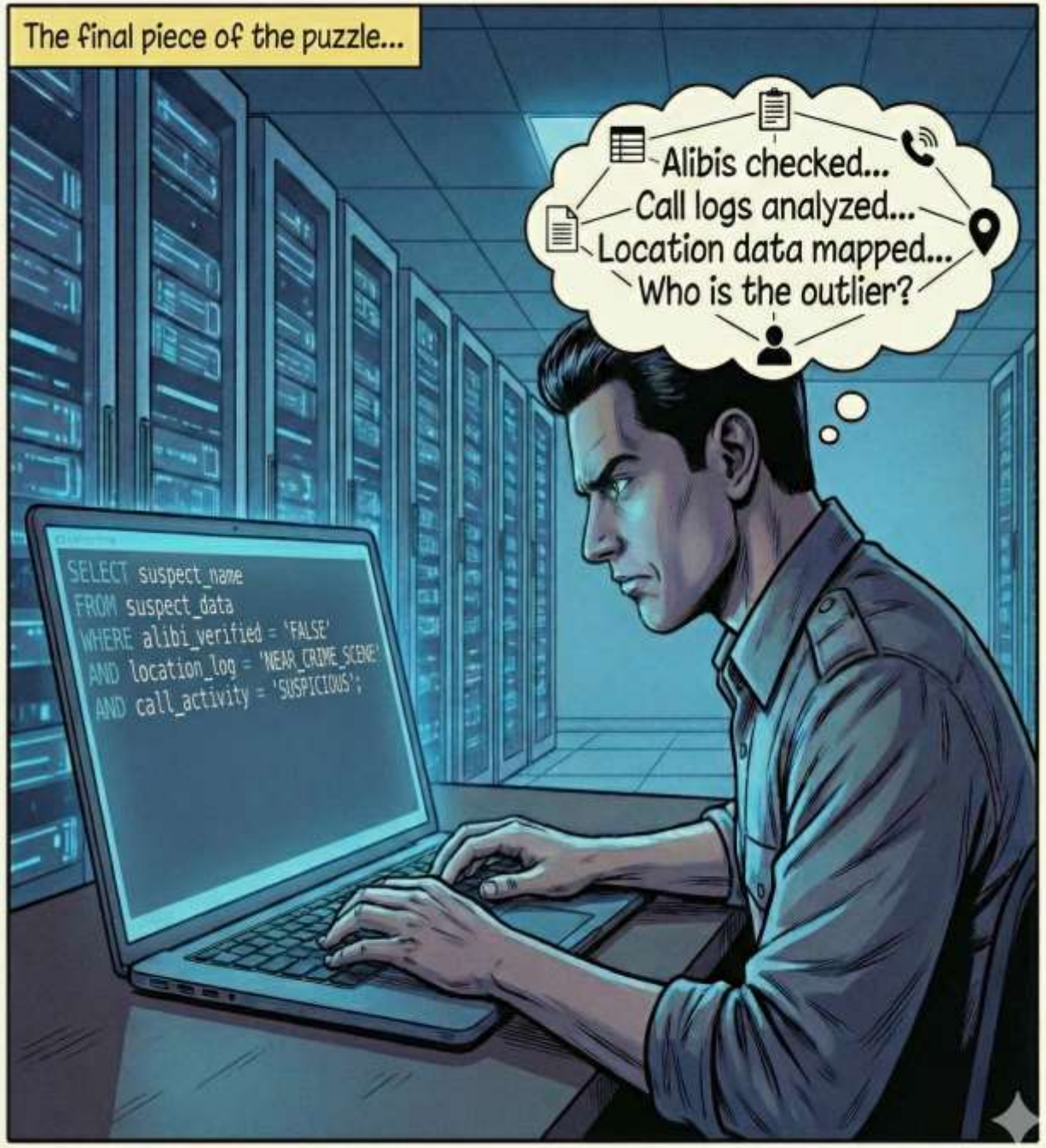
The suspicion on David Kumar gradually increases as we review the call history.

call_id	caller_id	caller_name	receiver_id	receiver_name	call_time	duration_sec	call_end_time
1	4	David Kumar	1	Alice Johnson	2025-10-15 20:55:00	45	2025-10-15 20:55:45


```sql
WITH room_logs_cte AS (
    SELECT
        k.employee_id,k.room,k.entry_time,k.exit_time,e.evidence_id,
        e.found_time,
        ROW_NUMBER() OVER (
            PARTITION BY e.room, e.evidence_id
            ORDER BY k.entry_time DESC
    ) AS rn
    FROM evidence AS e
    JOIN keycard_logs AS k
        ON e.room = k.room
       AND k.entry_time <= e.found_time)
SELECT
    r.evidence_id,r.room,r.found_time, r.employee_id, emp.name, r.entry_time,
    r.exit_time
FROM room_logs_cte AS r
JOIN employees AS emp
    ON emp.employee_id = r.employee_id
WHERE r.rn = 1
ORDER BY r.evidence_id;
```

The evidence overwhelmingly indicates David Kumar committed the murder.

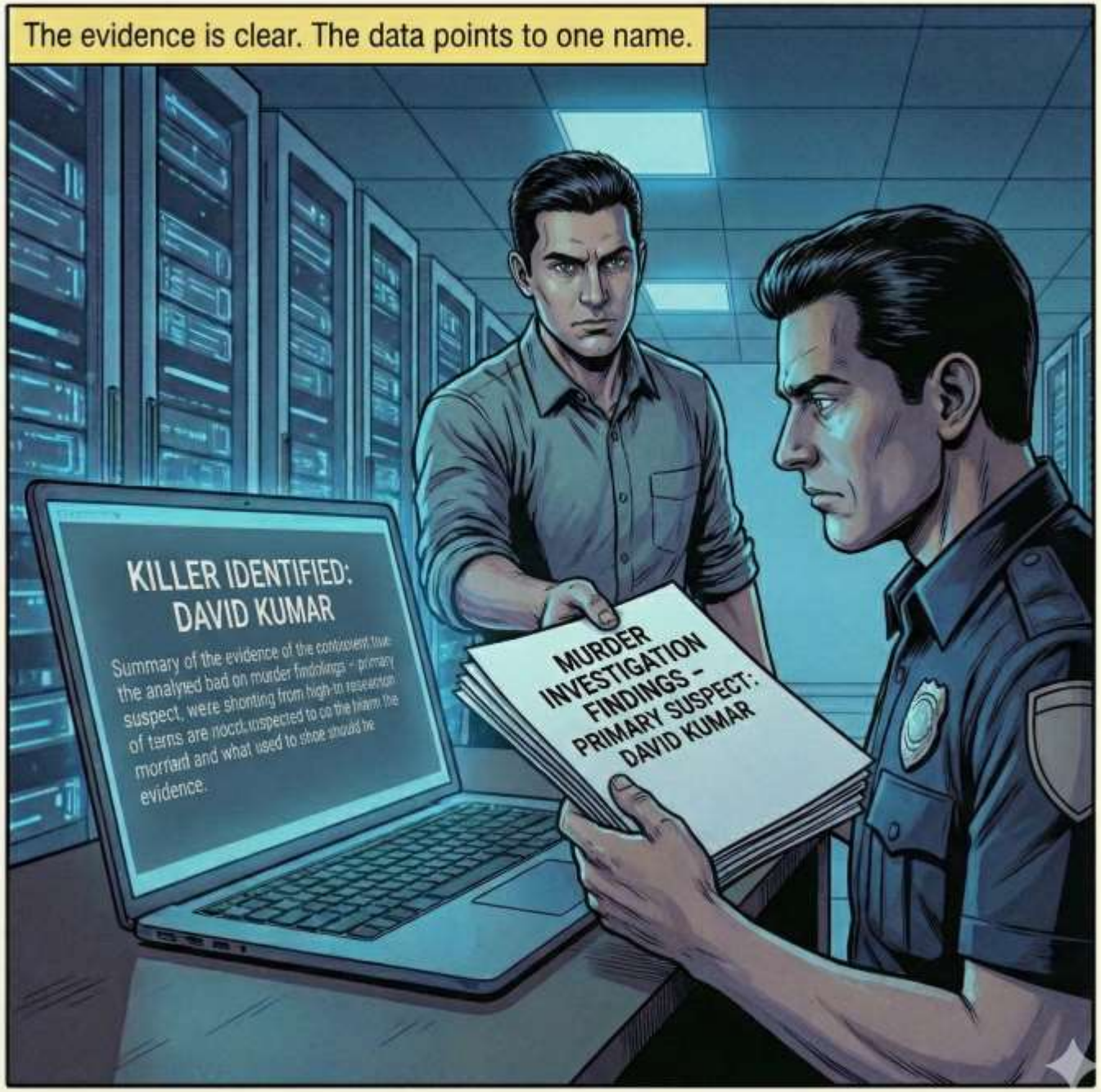
evidence_id	room	found_time	employee_id	name	entry_time	exit_time
1	CEO Office	2025-10-15 21:05:00	4	David Kumar	2025-10-15 20:50:00	2025-10-15 21:00:00
2	CEO Office	2025-10-15 21:10:00	4	David Kumar	2025-10-15 20:50:00	2025-10-15 21:00:00
3	Server Room	2025-10-15 21:15:00	4	David Kumar	2025-10-15 08:50:00	2025-10-15 09:10:00




```sql
SELECT DISTINCT
    CONCAT('Name of Killer: ', e.name) AS Killer
FROM employees e
JOIN
    alibis a ON e.employee_id = a.employee_id
JOIN
    keycard_logs l ON e.employee_id = l.employee_id
WHERE
    l.room = 'CEO Office'
    AND a.claimed_location <> l.room
    AND l.entry_time BETWEEN '2025-10-15 20:30:00' AND '2025-10-15 21:30:00';
```

Killer
Name of Killer: David Kumar

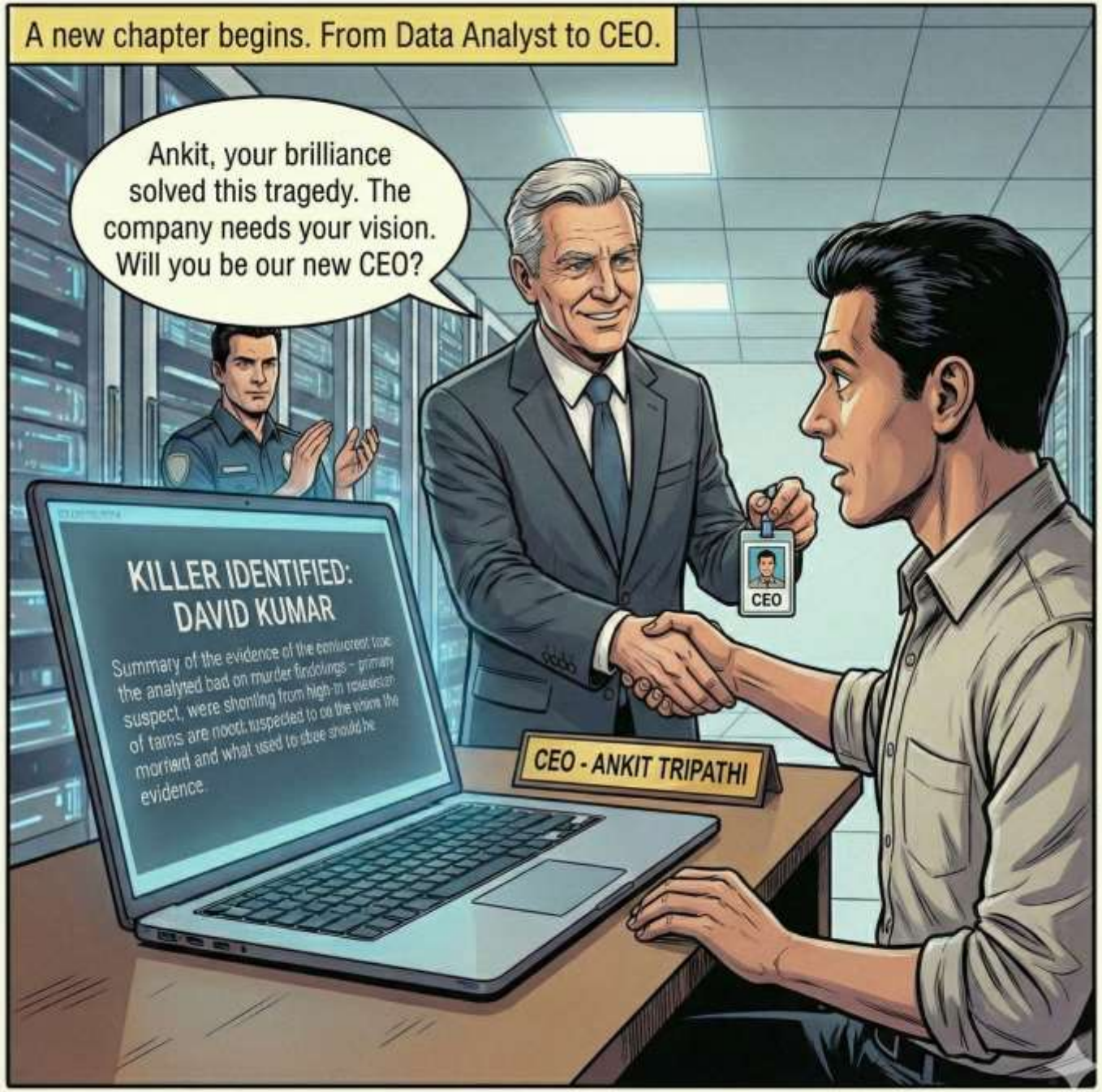
After analyzing every dataset, verifying every clue, and running multi-layer SQL investigations, the mystery finally reached its end.



After analyzing every dataset, verifying every clue, and running multi-layer SQL investigations, the mystery finally reached its end.

Our data analysis uncovered the complete chain of events — and the evidence pointed to one undeniable conclusion:

The killer in this SQL Murder Mystery case is *David Kumar*.
This marks the successful completion of the SQL Murder Mystery project, where structured queries, data cleaning, joins, CTEs, and analytical reasoning came together to solve the case of the CEO's



SQL Murder Mystery: "Who Killed the CEO?"

In this project, Data Analyst **Ankit Tripathi** stepped into the world of investigation — not with weapons, but with **SQL queries, logic, and data evidence.**

After digging through logs, cleaning datasets, analyzing call records, tracing movements, and connecting every clue…

He solved the high-profile case: "Who killed the CEO?"

Through precise SQL analysis, the truth was uncovered — and the mystery was finally cracked.

The successful completion of this investigation didn't just reveal the killer…

It transformed Ankit Tripathi from