SQL Project Report by Rajesh Sahu

“Who Killed the CEO”

Understand the the Perfect murder of CEO using the SQL

SQl Murder mystry by IDC

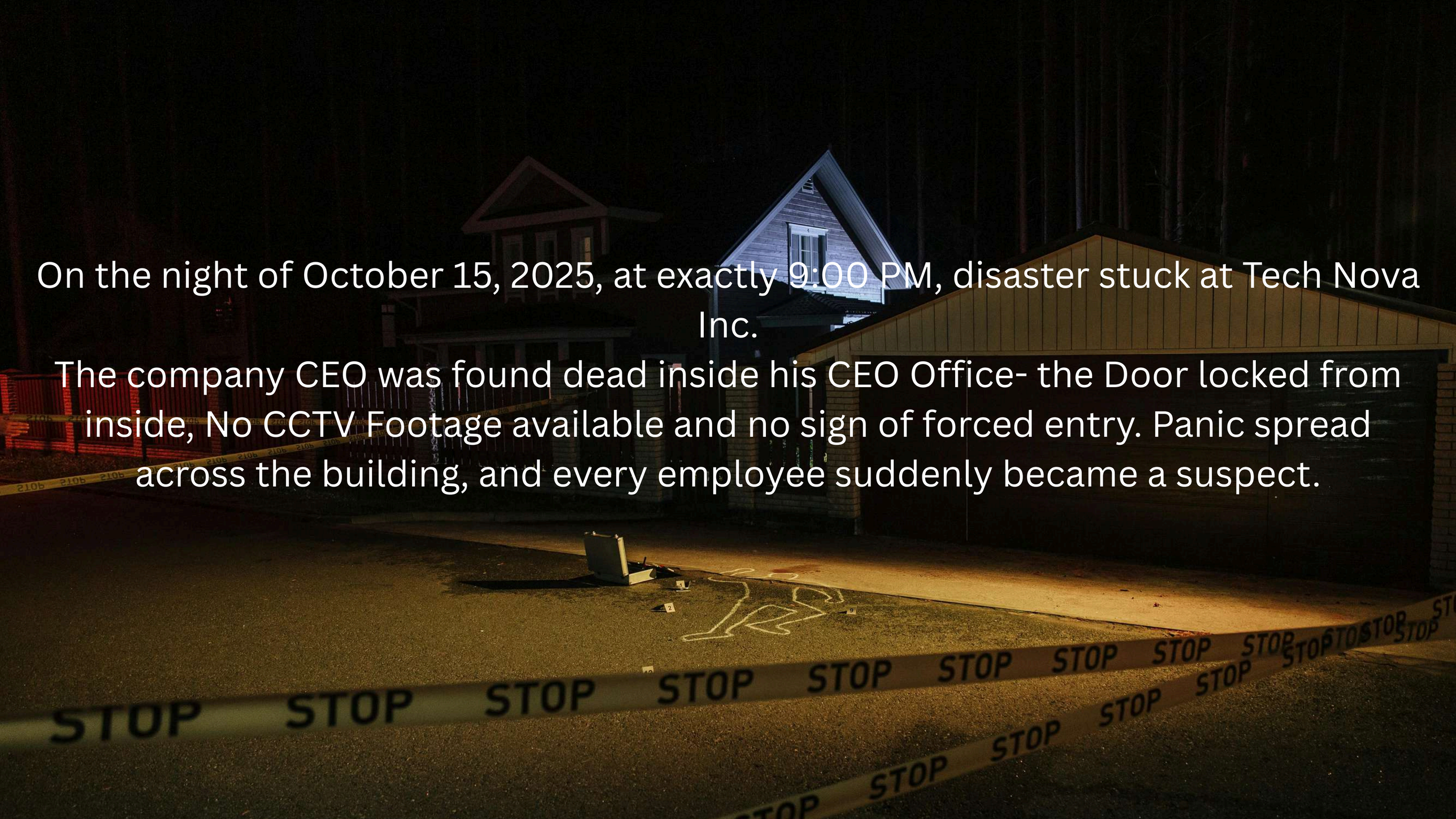
Story Baground

The CEO of TechNova Inc. has been found dead in their office on October 15, 2025, at 9:00 PM. As the lead data analyst tasked with solving this case using SQL. All the clues you need are hidden in the company's databases:

- Keycard logs
- Phone call records
- Alibis
- Evidence found in different rooms

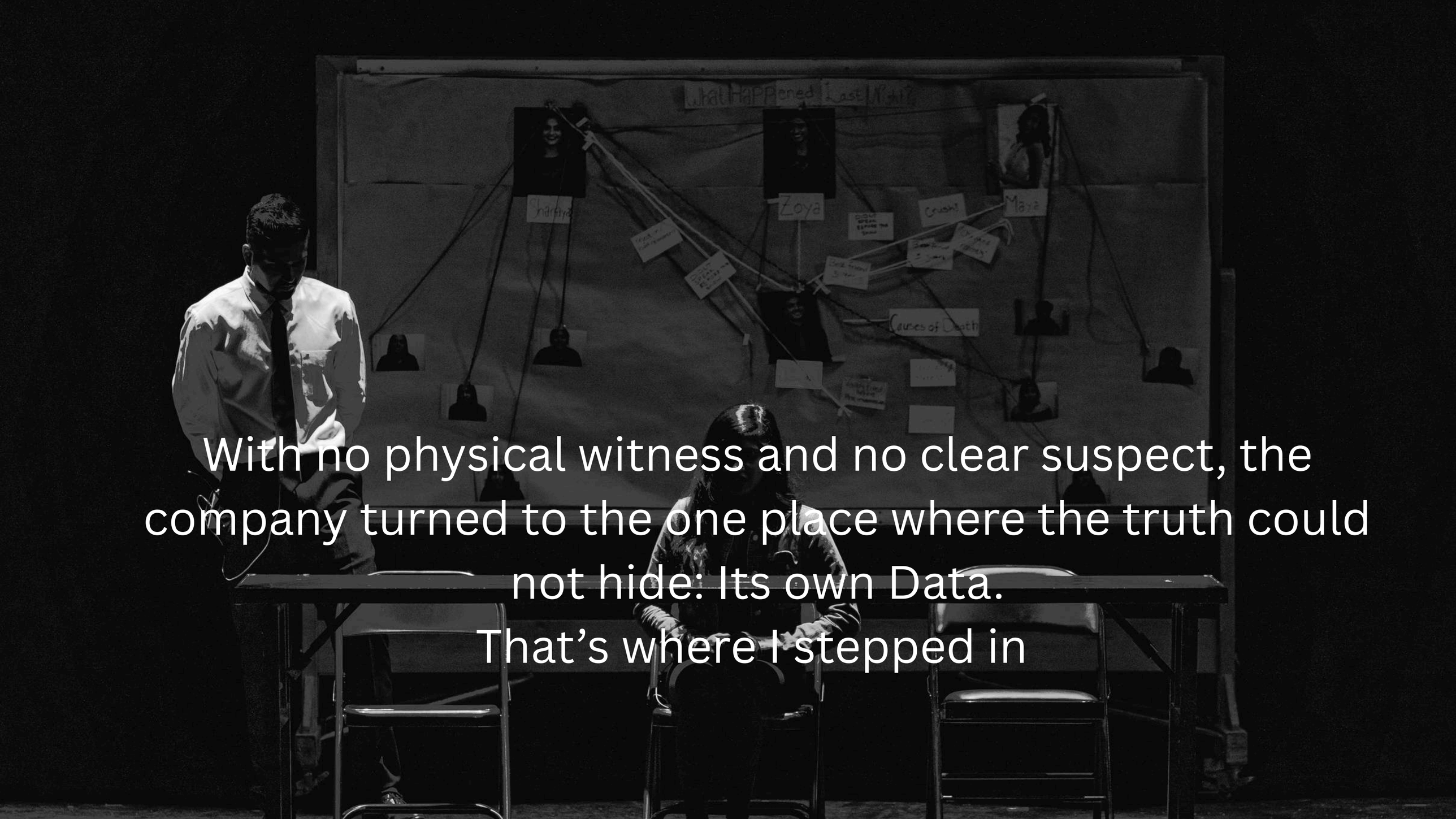
Mission is simple but challenging:

👉 Find out who the killer is, where and when the crime took place, and how it happened — using only SQL queries.



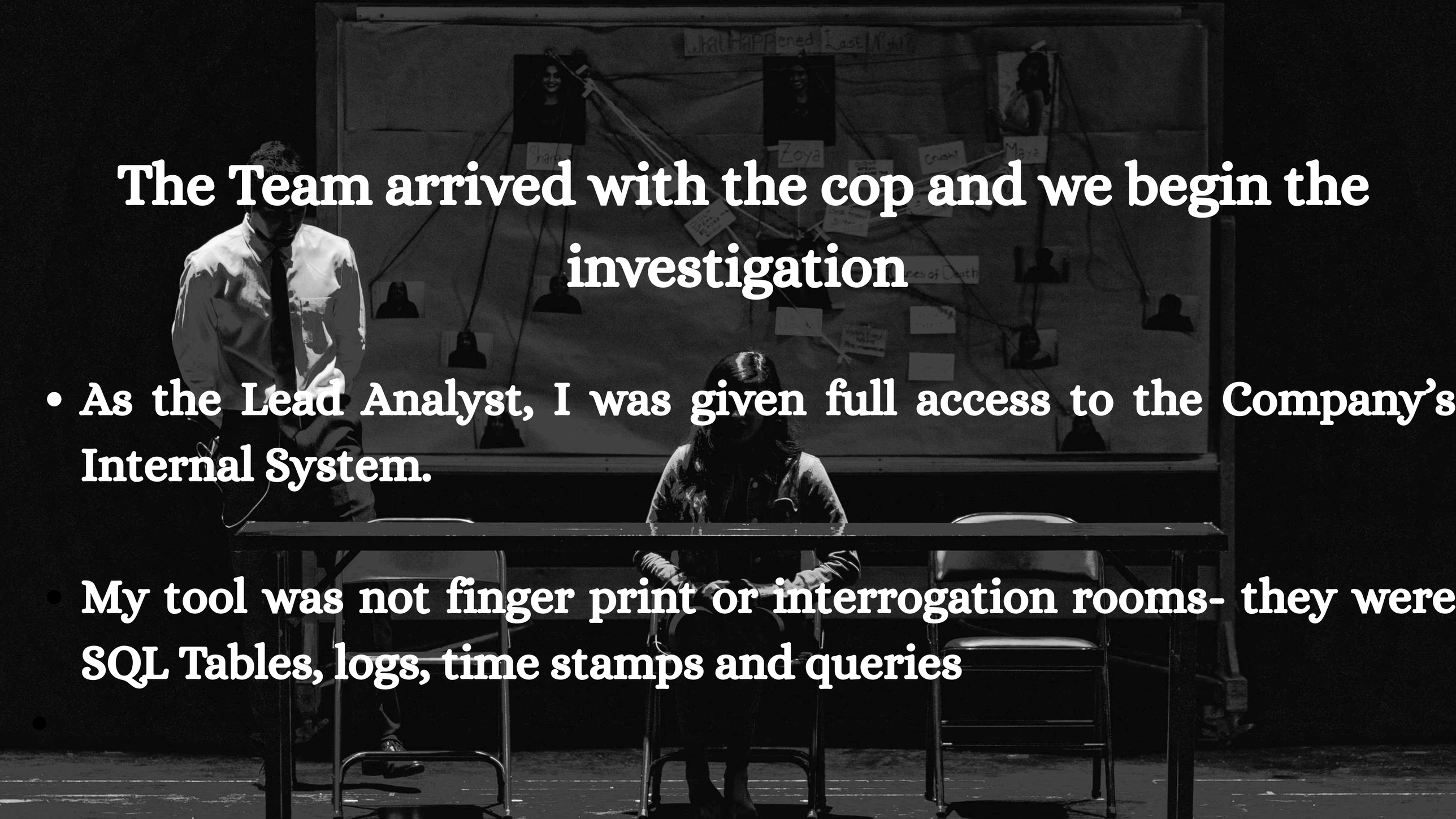
On the night of October 15, 2025, at exactly 9:00 PM, disaster stuck at Tech Nova Inc.

The company CEO was found dead inside his CEO Office- the Door locked from inside, No CCTV Footage available and no sign of forced entry. Panic spread across the building, and every employee suddenly became a suspect.



With no physical witness and no clear suspect, the company turned to the one place where the truth could not hide: Its own Data.

That’s where I stepped in



The Team arrived with the cop and we begin the investigation

- As the Lead Analyst, I was given full access to the Company's Internal System.
- My tool was not finger print or interrogation rooms- they were SQL Tables, logs, time stamps and queries
-

Data Base Overview

The database contains 5 key tables:

Table: employees

Column	Type	Description
employee_id	INT	Unique employee ID
name	VARCHAR	Full name of the employee
department	VARCHAR	Department of the employee
role	VARCHAR	Role or title

Data Base Overview

The database contains 5 key tables:

Table: alibis

Column	Type	Description
alibi_id	INT	Alibi record ID
employee_id	INT	Employee ID
claimed_location	VARCHAR	Where they said they were
claim_time	TIMESTAMP	When they claimed to be there

Data Base Overview

The database contains 5 key tables:

Table: calls

Column	Type	Description
call_id	INT	Call ID
caller_id	INT	ID of caller
receiver_id	INT	ID of receiver
call_time	TIMESTAMP	When the call happened
duration_sec	INT	Duration in seconds

Data Base Overview

The database contains 5 key tables:

Table: keycard_logs

Column	Type	Description
log_id	INT	Log ID
employee_id	INT	ID of employee
room	VARCHAR	Room name
entry_time	TIMESTAMP	Time they entered
exit_time	TIMESTAMP	Time they left

Data Base Overview

The database contains 5 key tables:

Table: evidence

Column	Type	Description
evidence_id	INT	Evidence ID
room	VARCHAR	Where evidence was found
description	VARCHAR	What was found
found_time	TIMESTAMP	When it was found

Investigation Strategy

- Identify the time of death
- Identify who was present in the building
- Analyse the suspicious Keycard_Access
- Match phone call around the incident
- Verify alibis vs actual movements
- Cross check with the physical evidence

“I approached the problem like a real investigation by filtering suspects basedon the location and behavioural pattern”

Analyse where and when crime happened

Query | Query History

```
11  --Analyse where and when crime happened
12
13  Select room as Crime_location,
14  entry_time as Time_entered,
15  exit_time as time_exited
16  from keycard_logs
17  where room='CEO Office'
18  order by entry_time;
19
```

Data Output | Messages | Notifications

Showing rows:

	crime_location character varying (50)	time_entered timestamp without time zone	time_exited timestamp without time zone
1	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00

Cross check who accessed the critical area at that time

Query | Query History

```
-- cross check who accessed the critical area at that time

Select k.employee_Id, e.name,e.role, k.room, k.entry_time,k.exit_time from keycard_logs k
join employees e on k.employee_id=e.employee_id
where k.room='CEO Office'
and k.entry_time='2025-10-15 20:50:00';
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page No: 1 of 1

employee_id integer	name character varying (50)	role character varying (50)	room character varying (50)	entry_time timestamp without time zone	exit_time timestamp without time
4	David Kumar	DevOps Engineer	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00

Cross check the alibis with actual logs

Query | Query History

```
28
29
30  -- Question 3 Cross check the alibis with actual logs
31  Select e.name, a.claimed_location,k.room,a.claim_time from alibis A
32  join keycard_logs k
33  on a.employee_id=k.employee_id
34  join employees e
35  on e.employee_id=a.employee_id
36  where a.claim_time between k.entry_time and k.exit_Time
37  and claimed_location<>K.room;
38
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page

	name character varying (50)	claimed_location character varying (50)	room character varying (50)	claim_time timestamp without time zone
1	David Kumar	Server Room	CEO Office	2025-10-15 20:50:00

Investigating the suspicious calls made around the time.

Query | Query History

```
40
41
42
43  --Investigating the suspicious calls made around the time.
44
45  Select e.name as caller_name, c.receiver_id, c.call_time,c.duration_sec
46  from calls c
47  join employees as e on
48  e.employee_id=c.caller_id
49  where call_time between '2025-10-15 20:45:00' and '2025-10-15 21:15:00';
50
```

Data Output | Messages | Notifications

SQL | Showing rows: 1 to 1 | Page No

	caller_name character varying (50)	receiver_id integer	call_time timestamp without time zone	duration_sec integer
1	David Kumar	1	2025-10-15 20:55:00	45

Match evidence with movement and claims

```
54
55  --Match evidence with movement and claims
56  Select distinct e.name,ev.room,ev.description from evidence ev
57  join keycard_logs k
58  on ev.room=k.room
59  join employees e
60  on e.employee_id=k.employee_id;
61
62
63
```

Data Output | Messages | Notifications

Showing rows: 1 to 4

	name character varying (50)	room character varying (50)	description character varying (255)
1	Henry Wu	Server Room	Unusual access pattern
2	David Kumar	CEO Office	Keycard swipe logs mismatch
3	David Kumar	CEO Office	Fingerprint on desk
4	David Kumar	Server Room	Unusual access pattern

Combine all findings to identify the Killer

Query | Query History

```
64
65  --Combine all findings to identify the killer
66
67  Select distinct e.employee_id,e.name from employees e join keycard_logs k
68  on e.employee_id=k.employee_id
69  join alibis as a
70  on e.employee_id=a.employee_id
71  where k.room='CEO Office'
72  and '2025-10-15 21:00:00' between entry_time and exit_time
73  and a.claimed_location<>'CEO Office';
```

Data Output | Messages | Notifications

Showing rows: 1 to 1 | Page

	employee_id [PK] integer	name character varying (50)
1	4	David Kumar

As per the Database investigation The Killer was found

Using keycard_logs, I identify employees present in the CEO Office at the time of crime then

I cross verified their and found the mismatch between claimed and actual locations

Finally, combining access logs and fales alibi analysis I concluded that David Kumar(Emp_Id 4) was

present at the crime location and fales alibi making him a killer