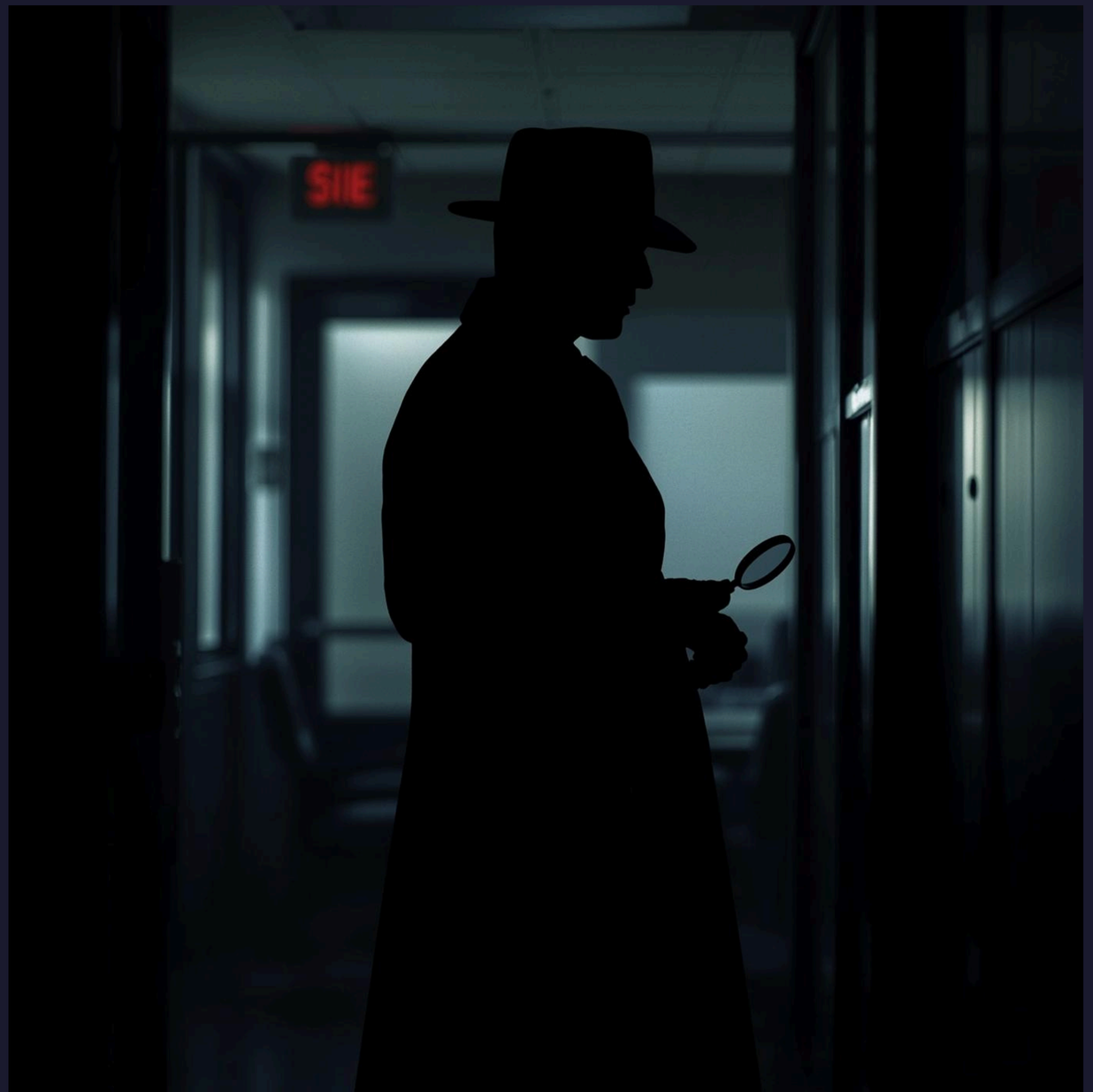
TECHNOVA INC.

SQL Murder Mystery: Solving the Crime at TechNova Inc.

A complete investigation using SQL to identify the killer.



Presented by: Lesie Fernandes

Case Overview

Incident Summary:

- Date: October 15, 2025
- Victim: CEO of TechNova Inc.
- Crime Location: CEO Office
- Time of Death: ~9:00 PM

Goal Use SQL to uncover:

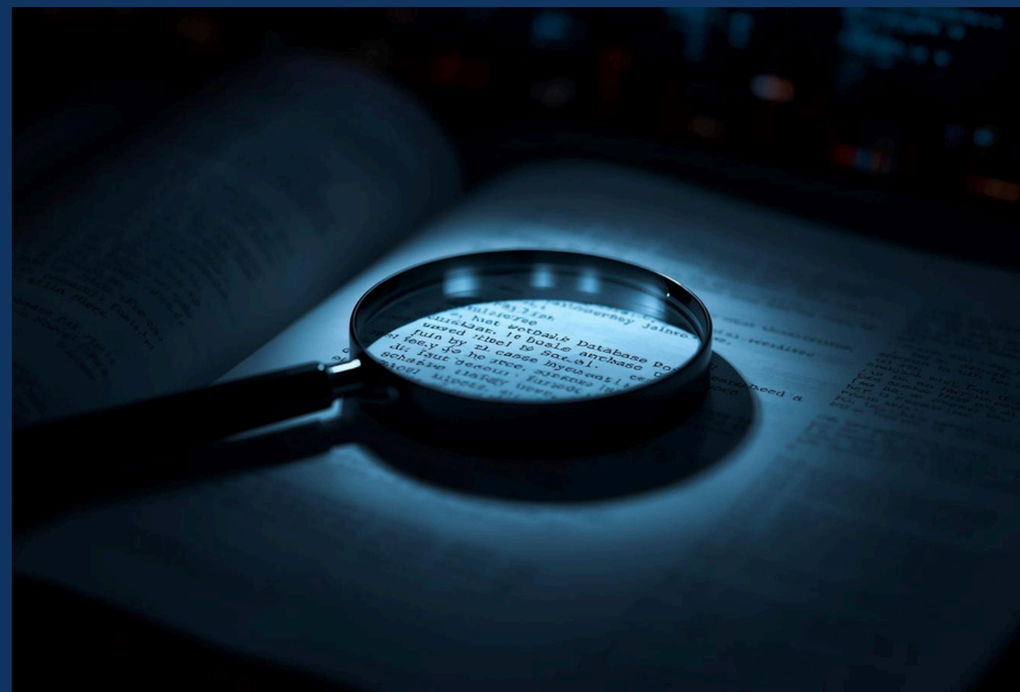
- Who was present
- Who lied
- Who interacted suspiciously
- Who ultimately killed the CEO

Data Used

Datasets Analyzed :

- Employees — Employee details
- Keycard_logs — Entry & exit movements
- Alibis — Claimed locations
- Calls — Calls made around crime time
- Evidence — Proofs found in the rooms

Each dataset acts as a clue in the investigation.



SQL QUERIES :

Identify where and when the crime happened:

```sql
SELECT
    room,
    found_time,
    description
FROM evidence
WHERE found_time BETWEEN '2025-10-15 20:55:00' AND '2025-10-15 21:05:00'
ORDER BY found_time;
```

room	found_time	description
CEO Office	2025-10-15 21:05:00	Fingerprint on desk

Insight:The crime took place in the CEO Office between 21:00 and 21:05.

Analyze who accessed critical areas at the time:

```sql
SELECT
    k.employee_id,
    e.name,
    e.department,
    k.room,
    k.entry_time,
    k.exit_time
FROM keycard_logs k
JOIN employees e
    ON k.employee_id = e.employee_id
WHERE '2025-10-15 21:00:00' BETWEEN k.entry_time AND k.exit_time;
```

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

Insight: David Kumar was in the CEO Office during the critical time.

Cross-check alibis with actual logs

```sql
SELECT
    a.employee_id,
    e.name,
    a.claim_time,
    a.claimed_location,
    k.room AS actual_location,
    k.entry_time,
    k.exit_time,
    CASE
        WHEN k.room IS NOT NULL THEN 'Matched'
        ELSE 'Mismatch'
    END AS match_status
FROM alibis a
JOIN employees e
    ON a.employee_id = e.employee_id
LEFT JOIN keycard_logs k
    ON a.employee_id = k.employee_id
    AND a.claim_time BETWEEN k.entry_time AND k.exit_time;
```

employee_id	name	claim_time	claimed_location	actual_location	entry_time	exit_time	match_status
1	Alice Johnson	2025-10-15 20:50:00	Office	NULL	NULL	NULL	Mismatch
4	David Kumar	2025-10-15 20:50:00	Server Room	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00	Matched
5	Eva Brown	2025-10-15 20:50:00	Marketing Office	NULL	NULL	NULL	Mismatch
6	Frank Li	2025-10-15 20:50:00	Office	NULL	NULL	NULL	Mismatch

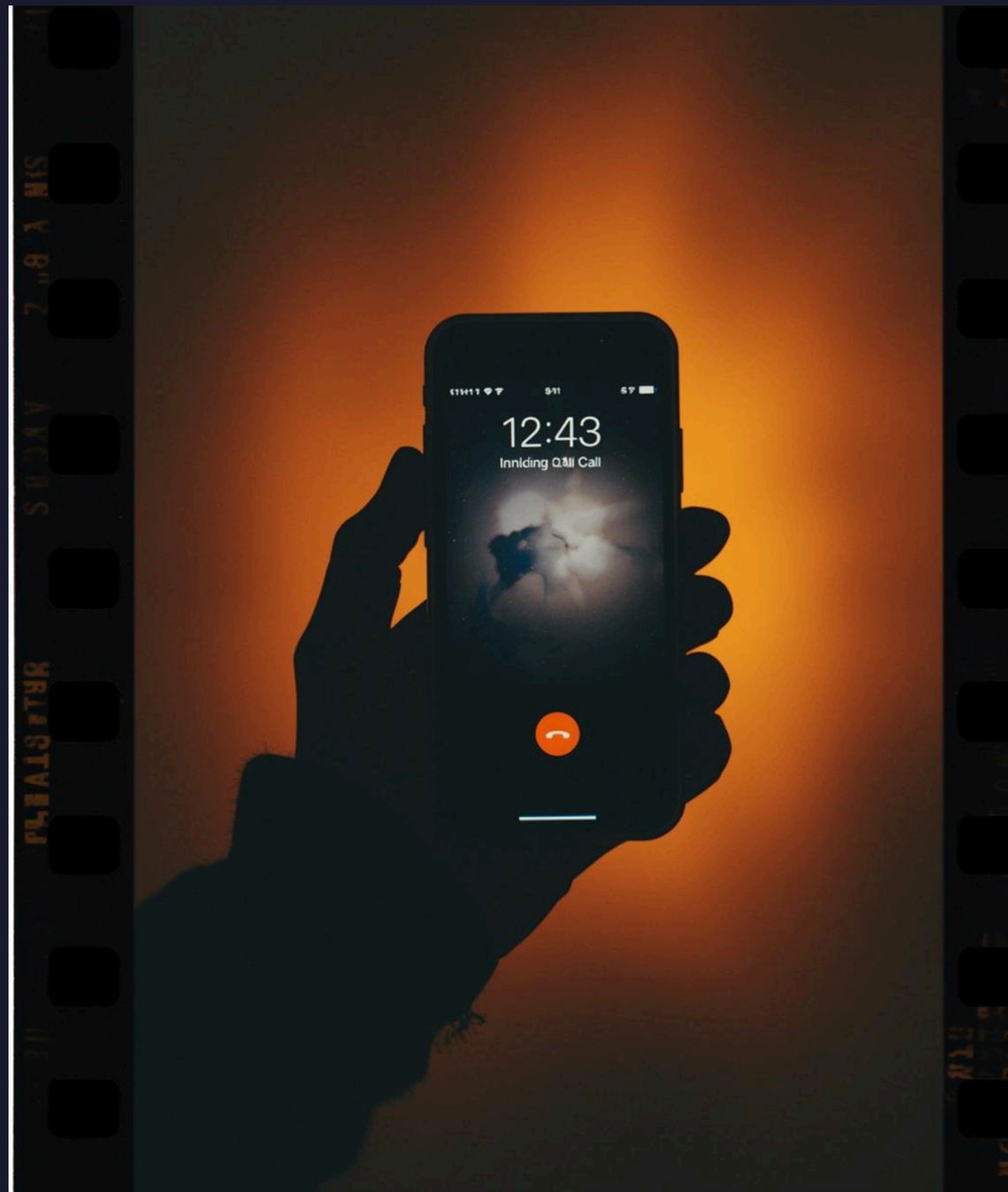
Insight: Several employees' alibis don't line up with keycard records, and David Kumar's claim is inconsistent.

Investigate suspicious calls made around the time:

```sql
SELECT
    c.call_id,
    e.name as caller_name,
    e1.name as receiver_name,
    c.receiver_id,
    c.call_time,
    c.duration_sec
FROM calls c
JOIN employees e
    ON c.caller_id = e.employee_id
join employees as e1
on c.receiver_id = e1.employee_id
WHERE c.call_time BETWEEN '2025-10-15 20:00:00' AND '2025-10-15 21:00:00';
```

call_id	caller_name	receiver_name	receiver_id	call_time	duration_sec
1	David Kumar	Alice Johnson	1	2025-10-15 20:55:00	45
5	David Kumar	Grace Tan	7	2025-10-15 20:40:00	90

Insights: Calls made by David Kumar during the crime raise further suspicion.



Match evidence with movements and claims:

```sql
SELECT
    evi.room AS evidence_room,
    evi.found_time,
    evi.description,
    emp.name,
    k.entry_time,
    k.exit_time,
    a.claim_time,
a.claimed_location
FROM evidence evi
JOIN keycard_logs k
    ON evi.room = k.room
JOIN employees emp
    ON k.employee_id = emp.employee_id
left join alibis as a
on emp.employee_id = a.employee_id;
```

evidence_room	found_time	description	name	entry_time	exit_time	claim_time	claimed_location
Server Room	2025-10-15 21:15:00	Unusual access pattern	David Kumar	2025-10-15 08:50:00	2025-10-15 09:10:00	2025-10-15 20:50:00	Server Room
Server Room	2025-10-15 21:15:00	Unusual access pattern	Henry Wu	2025-10-15 08:40:00	2025-10-15 09:05:00	NULL	NULL
CEO Office	2025-10-15 21:10:00	Keycard swipe logs mismatch	David Kumar	2025-10-15 20:50:00	2025-10-15 21:00:00	2025-10-15 20:50:00	Server Room
CEO Office	2025-10-15 21:05:00	Fingerprint on desk	David Kumar	2025-10-15 20:50:00	2025-10-15 21:00:00	2025-10-15 20:50:00	Server Room

Insight: Evidence from the CEO Office, combined with conflicting claims, points directly to David Kumar.

Combine all findings to identify the killer:

Step 1: Presence

```sql
WITH presence AS (
    SELECT
        k.employee_id,
        e.name,
        k.room,
        k.entry_time,
        DATE_ADD(k.exit_time, INTERVAL 5 MINUTE) AS extended_exit_time
    FROM keycard_logs k
    JOIN employees e
        ON k.employee_id = e.employee_id
),
```

Combine all findings to identify the killer:

Step 2: Crime Room

```
crime_room AS (
    SELECT room, found_time
    FROM evidence
    WHERE found_time BETWEEN '2025-10-15 21:00:00' AND '2025-10-15 21:05:00'
),
```

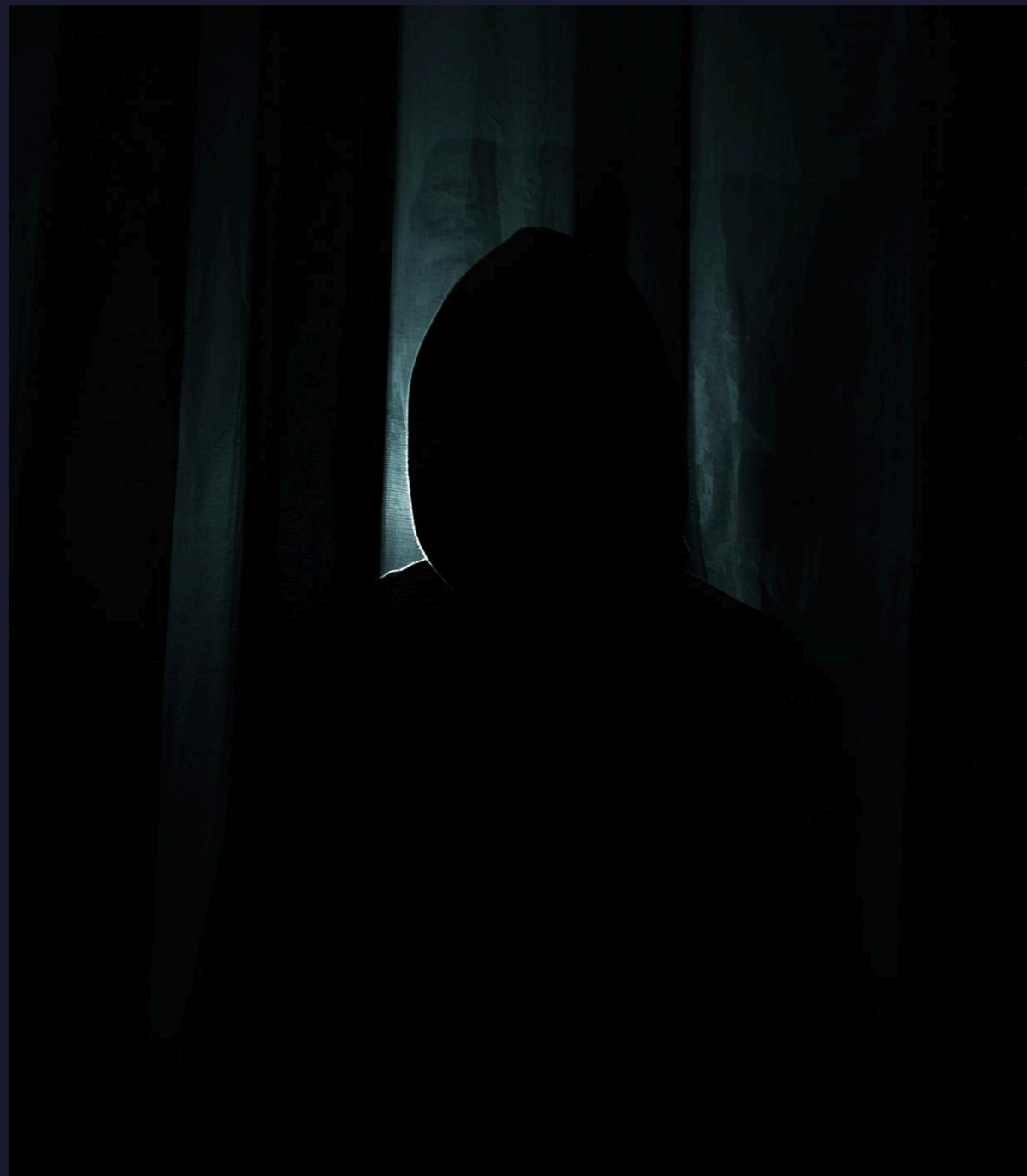
Combine all findings to identify the killer:

Step 3: Suspect Filtration

```sql
suspects AS (
    SELECT DISTINCT
        p.employee_id ,
        p.name as killer,
        p.room,
        p.entry_time,
        p.extended_exit_time
    FROM presence p
    JOIN crime_room c
        ON p.room = c.room
        AND c.found_time BETWEEN p.entry_time AND p.extended_exit_time
)
SELECT killer
FROM suspects;
```

killer
David Kumar

Insight: A step-by-step SQL investigation confirms David Kumar as the culprit.



The Killer Revealed

- Only employee inside the CEO Office during murder window
- Alibi does not match real movement data
- Call before the crime strengthens suspicion
- Evidence, logs, and claims all align perfectly with his timeline

- **Conclusion**: The SQL investigation conclusively identifies **David Kumar as the murderer**.

Thank you