1. Identified the crime location & exact timeline using evidence timestamps

```sql
4  SELECT
5       *
6  FROM evidence
7  WHERE room = 'CEO Office';
8
```

evidence_id	room	description	found_time
1	CEO Office	Fingerprint on desk	2025-10-15 21:05:00
2	CEO Office	Keycard swipe logs mismatch	2025-10-15 21:10:00

2. Analyzed who accessed the CEO's Office near the murder time

```sql
14 •  SELECT
15        kl.log_id,
16        kl.employee_id,
17        e.name,
18        kl.room,
19        kl.entry_time,
20        kl.exit_time
21    FROM keycard_logs kl
22    JOIN employees e ON kl.employee_id = e.employee_id
23    WHERE kl.entry_time BETWEEN '2025-10-15 20:40:00' AND '2025-10-15 21:10:00'
24    AND kl.room = 'CEO Office';
```

Result Grid | 🔢 | Filter Rows: | Export: | Wrap Cell Contents: 🔲

log_id	employee_id	name	room	entry_time	exit_time
11	4	David Kumar	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00

3. Cross-verified alibis with actual movements using keycard logs

```sql
40 ⊖ WITH
41   claims AS (
42     SELECT employee_id, claimed_location, claim_time
43     FROM alibis
44     WHERE claim_time BETWEEN '2025-10-15 20:40:00' AND '2025-10-15 21:10:00'
45   ),
46 ⊖ actual AS (
47     SELECT employee_id, room, entry_time, exit_time
48     FROM keycard_logs
49     WHERE entry_time BETWEEN '2025-10-15 20:40:00' AND '2025-10-15 21:10:00'
50   )
51   SELECT
52     c.employee_id,
53     e.name,
54     c.claimed_location,
55     a.room AS actual_room,
56     a.entry_time,
57     a.exit_time
58   FROM claims c
59   LEFT JOIN actual a
60     ON c.employee_id = a.employee_id
61     AND (c.claim_time BETWEEN a.entry_time AND a.exit_time)
62   JOIN employees e ON c.employee_id = e.employee_id
63   WHERE c.claimed_location <> a.room
64   ORDER BY c.employee_id;
```

employee_id	name	claimed_location	actual_room	entry_time	exit_time
4	David Kumar	Server Room	CEO Office	2025-10-15 20:50:00	2025-10-15 21:00:00

4. Investigated suspicious phone calls made around the critical 20:50–21:10 window

```sql
68 ●   SELECT
69         c.*,
70         e1.employee_id,
71         e1.name AS caller_name,
72         e2.employee_id,
73         e2.name AS receiver_name
74     FROM calls c
75     JOIN employees e1 ON c.caller_id = e1.employee_id
76     JOIN employees e2 ON c.receiver_id = e2.employee_id
77     WHERE c.call_time BETWEEN '2025-10-15 20:50:00' AND '2025-10-15 21:10:00'
78     ORDER BY c.call_time;
```

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

5. Matched evidence with movement patterns

```sql
86  •  SELECT
87         ev.evidence_id,
88         ev.room,
89         ev.description,
90         ev.found_time,
91         k.employee_id,
92         em.name,
93         k.entry_time,
94         k.exit_time
95     FROM evidence ev
96     LEFT JOIN keycard_logs k ON ev.room = k.room
97        AND k.entry_time BETWEEN ev.found_time - INTERVAL 30 MINUTE AND ev.found_time
98     JOIN employees em ON k.employee_id = em.employee_id
99     WHERE ev.room = 'CEO Office'
100    ORDER BY ev.found_time;
```

	evidence_id	room	description	found_time	employee_id	name	entry_time	exit_time
▶	1	CEO Office	Fingerprint on desk	2025-10-15 21:05:00	4	David Kumar	2025-10-15 20:50:00	2025-10-15 21:00:00
	2	CEO Office	Keycard swipe logs mismatch	2025-10-15 21:00:00	4	David Kumar	2025-10-15 20:50:00	2025-10-15 21:00:00

6. Consolidated all findings to pinpoint the killer

```sql
149  ⊟ WITH
150    employee_details AS (
151        SELECT *
152        FROM employees
153    ),
154  ⊟ key_card_logs AS (
155        SELECT employee_id
156        FROM keycard_logs
157        WHERE entry_time BETWEEN '2025-10-15 20:40:00' AND '2025-10-15 21:20:00'
158    ),
159  ⊟ alibis_found AS (
160        SELECT employee_id, claimed_location
161        FROM alibis
162        WHERE claim_time BETWEEN '2025-10-15 20:40:00' AND '2025-10-15 21:20:00'
163    ),
```

```sql
164  ⊟ sus_call AS (
165        SELECT DISTINCT caller_id AS employee_id
166        FROM calls
167        WHERE call_time BETWEEN '2025-10-15 20:50:00' AND '2025-10-15 21:20:00'
168        UNION
169        SELECT DISTINCT receiver_id AS employee_id
170        FROM calls
171        WHERE call_time BETWEEN '2025-10-15 20:50:00' AND '2025-10-15 21:20:00'
172    )
173    SELECT
174        ed.*
175    FROM employee_details ed
176    JOIN key_card_logs kcl ON ed.employee_id = kcl.employee_id
177    JOIN alibis_found af ON ed.employee_id = af.employee_id
178    JOIN sus_call sc ON ed.employee_id = sc.employee_id;
```

employee_id	name	department	role
4	David Kumar	Engineering	DevOps Engineer