Technical Writing Sample by Patricio Kobek:

SQL Murder Mystery

Introduction:

The SQL Murder Mystery offers an efficient fictional narrative that allows a user to practice the SQL at a basic or intermediate level. This guide assumes that the user has completed a beginner or intermediate SQL course. See my Portfolio Guide by clicking here for more projects.

As always, you should try and complete the problems first yourself, but if you require assistance, following the steps below will help solve the case with a little more explanation as to how and why these SQL queries function like they do.

Tips To Solve The Case On Your Own:

- Pay attention to the details in each challenge, including the clues, the data, and the structure of the database.
- Use the SQL syntax to query the database and retrieve the information you need to solve each challenge.
- Keep track of your progress and the clues you uncover by taking notes or using a spreadsheet. This is especially important because the webpage does not keep your previous queries (Use "/**/" notes).
- Use logical reasoning and deductive skills to eliminate suspects and identify the one who fits all the criteria for the killer.
- Be patient and persistent, as some challenges may require multiple attempts and careful analysis to solve.

Step-by-Step Solution:

- First, what information are we given at the start:
- "Security footage shows that there were 2 witnesses. The first witness lives at the last house on "Northwestern Dr"., The second witness, named Annabel, lives somewhere on "Franklin Ave"."

1. SELECT * FROM person WHERE name LIKE "%Annabel%" AND address_street_name = "Franklin Ave";

- Result: 16371 Annabel Miller 490173 103 Franklin Ave 318771143
- This query selects all the columns of a table named "person", and then filters the results by looking for records where the "name" column contains the string "Annabel" (case-insensitive) and the "address_street_name" column is exactly equal to "Franklin Ave". The next query repeats this for the second witness.

SELECT MAX(address_number) FROM person WHERE address_street_name = "Northwestern Dr";

Result: Address is 4919

SELECT * FROM person WHERE address_number = "4919";

Result: id = 14887 name = Morty Schapiro license_id 118009

address_number = 4919 address_street_name = Northwestern Dr ssn = 111564949

- When examining these results, what do you notice about the fields that could be useful for searching the other available tables of data? (Hint: LEFTJOIN)
- Answer: Interview person_id field is equal to table.person id

2. SELECT * FROM interview WHERE person_id = "14887" and "16371";

Result: Morty Schapiro states: I heard a gunshot and then saw a man run out. He had a "Get Fit Now Gym" bag. The membership number on the bag started with "48Z". Only gold members have those bags. The man got into a car with a plate that included "H42W".

Annabel Miller states: I saw the murder happen, and I recognized the killer from my gym when I was working out last week on January the 9th.

3. SELECT * FROM get_fit_now_member where membership_status = "gold" AND id LIKE "48Z%";

Two results:

id	person_id	name	membership_start_date	membership_status
48Z7A	28819	Joe Germuska	20160305	gold
48Z55	67318	Jeremy Bowers	20160101	gold

- This narrows down our suspects to two individuals.

4. SELECT * FROM drivers_license WHERE plate_number like "%H42W%" AND gender = "male";

- Again, two results:

id	age	height	eye_color	hair_color	gender	plate_number	car_make	car_model
423327	30	70	brown	brown	male	0H42W2	Chevrolet	Spark LS
664760	21	71	black	black	male	4H42WR	Nissan	Altima

5. SELECT * FROM get_fit_now_check_in WHERE membership_id = "48Z7A"

AND check_in_date = "20180109" OR membership_id = "48Z55"

AND check_in_date = "20180109";

Result: Inconclusive, both men were present.

membership_id check_in_date check_in_time check_out_time

48Z7A 20180109 1600 1730

48Z55 20180109 1530 1700

- What should we look up next?
- The time that Annabel Miller was working out, as we want to know which of those two suspects was there at that time. Whoever it is can be considered the killer!
- Unfortunately, they were both present during this time.

6. WITH gym_checkins AS (SELECT person_id, name FROM get_fit_now_member

LEFT JOIN get_fit_now_check_in ON get_fit_now_member.id = get_fit_now_check_in.membership_id

WHERE membership_status = 'gold'

AND id REGEXP '^48Z' -- membership number on the bag started with "48Z"

AND check_in_date = '20180109'

), suspects AS (

SELECT gym_checkins.person_id, gym_checkins.name, plate_number, gender

FROM gym_checkins

LEFT JOIN person ON gym_checkins.person_id = person.id

LEFT JOIN drivers_license ON person.license_id = drivers_license.id

)

SELECT * FROM suspects

WHERE INSTR(plate_number, 'H42W') > 0 AND gender = 'male'

- Here we can use a LEFTJOIN to pull the remaining information.
- This reveals that only Jeremy Bowers fits the parameters to be the killer, so it is time to accuse him.

7. INSERT INTO solution VALUES (1, "Jeremy Bowers");

SELECT value FROM solution

- This confirms that we have found the killer, however, it is not the end of the mystery.
- You are now tasked with an additional challenge, querying the interview transcript of the murderer to find yet another involved person, the one who hired the killer to act.
- As the instructions state, try to do this with two queries.

8.
```
WITH red_haired_tesla_drivers AS (SELECT id AS license_id FROM drivers_license
  WHERE gender = 'female' AND hair_color = 'red'
    AND car_make = 'Tesla' AND car_model = 'Model S'
    AND height >= 64 AND height <= 68 -- she's around 5'5" (65") or 5'7" (67")), rich_suspects AS (
  SELECT person.id AS person_id, name, annual_income
  FROM red_haired_tesla_drivers AS rhtd
  LEFT JOIN person ON rhtd.license_id = person.license_id
  LEFT JOIN income ON person.ssn = income.ssn
), symphony_attenders AS (
  SELECT person_id, COUNT(1) AS n_checkins
  FROM facebook_event_checkin
  WHERE event_name = 'SQL Symphony Concert' -- she attended the SQL Symphony Concert
    AND `date` REGEXP '^201712' -- in December 2017
  GROUP BY person_id
  HAVING n_checkins = 3
)
SELECT name, annual_income
FROM rich_suspects
INNER JOIN symphony_attenders ON rich_suspects.person_id = symphony_attenders.person_id
```

- There we have it! Miranda Priestly is revealed as the mastermind who planned the murder from the beginning.
- You've solved the case, but can you understand exactly what each query has accomplished? A good review activity would be to check each query individually. You may notice that they begin as simple queries, but soon become more and more complex.