

## SQL Murder Mystery:

A crime has taken place and the detective needs your help. The detective gave you the crime scene report, but you somehow lost it. You vaguely remember that the crime was a **murder** that occurred sometime on **Jan.15, 2018** and that it took place in **SQL City**. Start by retrieving the corresponding crime scene report from the police department's database.

### Starting Information:

1. The crime is a murder
2. The crime took place on January 15, 2018
3. The crime took place in SQL City

### Step 1: Query the crime scene reports to assert that there is a record of a murder taking place on January 15, 2018 in SQL City.

```
SELECT * FROM crime_scene_report WHERE date==20180115 AND city=='SQL City';
```

*In this query we select all of the columns from the crime\_scene\_report table that has the matching date and city of the incident in question.*

date	type	description	city
20180115	assault	Hamilton: Lee, do you yield? Burr: You shot him in the side! Yes he yields!	SQL City
20180115	assault	Report Not Found	SQL City
20180115	murder	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".	SQL City

**Figure 1:** We know that the crime was a murder so it is clear that the highlighted record describes the incident.

**Description:** We are able to discover that the incident in question had two witnesses. The first witness lives in the last house on Northwestern Dr, and the second witness is named Annabel, and she lives on Franklin Ave. Our next task should be to read the interviews of the witnesses to collect more information on the murder suspect(s).

## Step 2: Join the interview table with the persons table to view the transcripts of the witness interviews paired with the witness names.

```
SELECT p.name, p.id, i.transcript FROM person p JOIN interview i ON  
i.person_id == p.id WHERE p.id IN (14887, 16371);
```

*In this query we are joining the persons table with the interview table on the condition that the person\_id in the interview table matches the id in the person table. The purpose of this join is so that we can display the person's name in the output of the query. The person id is not a meaningful number to colleagues that will read our report, for this reason it is important to include the person name in our output table.*

name	id	transcript
Morty Schapiro	14887	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	16371	I saw the murder happen, and I recognized the killer from my gym when I was working out last week on January the 9th.

**Figure 2:** This table combines the name, person id, and corresponding transcript of all the witnesses to the murder.

**Description:** From the resulting output we are able to read the transcripts of the witness reports. We learn several important insights from these interviews. I have listed new information regarding the murder suspect below.

### Suspect Information:

1. The suspect is a man
2. The suspect had a "Get Fit Now Gym" bag
3. The suspect's bag had an ID number starting with "48Z"
4. The suspect was likely a gold member to the "Get Fit Now Gym"
5. The suspect got into a car that had the phrase "H42W" as a subset of the license plate number
6. The suspect was at the "Get Fit Now Gym" on January 9th, 2018

**Step 3: Find the names and IDs of people who were at the gym on January 9th, 2018, and who are gold members with member IDs that start with “48Z”.**

```
SELECT gfnm.id, gfnm.person_id, gfnm.name, gfnm.membership_status,
gfnchi.check_in_date FROM get_fit_now_member gfnm JOIN get_fit_now_check_in
gfnchi ON gfnm.id==gfnchi.membership_id WHERE membership_status=='gold' AND
gfnchi.check_in_date==20180109 AND gfnm.id LIKE '48Z%';
```

*In this query we collect metadata describing people who were at the gym on January 9, 2018, and were gold members and had member ids starting with '48Z'. We made a join on the get\_fit\_now\_member table with the get\_fit\_now\_check\_in table based on matching membership id numbers. The purpose of the join was to be able to include the check in date in the resulting output. Including the check in date would be important for anyone who was reading the report who maybe did not have access to the database.*

id	person_id	name	membership_status	check_in_date
48Z7A	28819	Joe Germuska	gold	20180109
48Z55	67318	Jeremy Bowers	gold	20180109

**Figure 3:** This output table holds records of people that match all of the criteria outlined in the witness interviews. It is safe to say that either Joe Germuska or Jeremy Bowers committed the murder.

**Step 4: Check if either of the suspects are registered to a vehicle that was the getaway car. The getaway car had ‘H42W’ in its license plate number.**

```
SELECT p.name, p.id, dl.plate_number, dl.car_make, dl.car_model FROM
drivers_license dl JOIN person p ON dl.id == p.license_id WHERE
dl.plate_number LIKE '%H42W%'
```

*In this query we join the drivers\_license table with the person table to collect information on people who have vehicles that contain the substring 'H42W' somewhere in its license plate number. Please note that the LIKE operator in SQL is case insensitive.*

name	id	plate_number	car_make	car_model
Tushar Chandra	51739	4H42WR	Nissan	Altima
Jeremy Bowers	67318	0H42W2	Chevrolet	Spark LS
Maxine Whitely	78193	H42W0X	Toyota	Prius

**Figure 4:** This output table contains the names and id numbers of all the people who are registered to vehicles with the 'H42W' substring in their license plate numbers.

**Description:** In this step, we can definitively say that Jeremy Bowers matched all of the criteria outlined by the witnesses, and he has registered a car that contains the phrase 'H42W' which was part of a witness testimony. This is enough evidence to reasonably accuse Jeremy Bowers of the murder.

### Step 5: For further investigation, search for interview transcripts from Jeremy Bowers.

```
SELECT p.name, p.id, i.transcript FROM interview i JOIN person p ON
p.id=i.person_id WHERE person_id==67318;
```

*In this query we are joining the interview and the person table on person id in order to view the transcript from Jeremy Bowers*

name	id	transcript
Jeremy Bowers	67318	I was hired by a woman with a lot of money. I don't know her name but I know she's around 5'5" (65") or 5'7" (67"). She has red hair and she drives a Tesla Model S. I know that she attended the SQL Symphony Concert 3 times in December 2017.

**Figure 5:** This output table contains the transcript from the interview from Jeremy Bowers.

**Description:** From the Jeremy Bowers transcript we learn several key insights regarding the circumstances of his crime. We learn that he was hired by a woman who had a lot of money, who was between 65 and 67 inches tall, with red hair, drives a Tesla Model S, and attended the SQL Symphony Concert three times in December 2017.

## 6. Search for the name of the woman who hired Jeremy Bowers to commit the crime.

```
SELECT p.id, p.name, dl.height, dl.hair_color, dl.car_make,
dl.car_model, fec.event_name, fec.date
FROM person p
JOIN drivers_license dl ON dl.id==p.license_id
JOIN facebook_event_checkin fec ON fec.person_id==p.id
WHERE dl.height >= 65 AND dl.height <= 67
AND dl.gender=='female' AND dl.hair_color=='red'
AND dl.car_make=='Tesla' AND dl.car_model=='Model S'
AND fec.event_name LIKE '%SQL Symphony Concert%';
```

*In this query we have to aggregate data from three different tables, and that is why we use two join statements. We join the person table and the drivers license table based on the license id, and we also join the person table to the facebook\_event\_checkin table based on the person id. We then filter our results based on the attributes of Jeremy Bowers' boss from his interview transcript. Note that we have skipped over the clue that the boss has a lot of money, since this is a subjective piece of data it is hard to quantify what "a lot" means.*

id	name	height	hair_color	car_make	car_model	event_name	date
99716	Miranda Priestly	66	red	Tesla	Model S	SQL Symphony Concert	20171206
99716	Miranda Priestly	66	red	Tesla	Model S	SQL Symphony Concert	20171212
99716	Miranda Priestly	66	red	Tesla	Model S	SQL Symphony Concert	20171229

**Figure 6:** This table shows three instances of Jeremy Bowers' boss, Miranda Priestly, attending the SQL Symphony Concert in December of 2017.

## **Conclusion**

We have gathered enough evidence to accuse Jeremy Bowers of the murder in SQL City on January 15, 2018. Upon further investigation it has become clear that he was not acting alone, and that he was put up to the crime by a red haired woman named Miranda Priestly. The motive behind the crime is unclear, other than that Miranda is a wealthy woman and she likely wanted to set up Jeremy with the blame in case the authorities were able to solve the crime.