

WHO DONE IT? A MURDER INVESTIGATION REPORT

BY: MONIQUE WEBLEY

Overview: A crime has taken place and detectives need 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 January 15, 2018, and it took place in SQL City. Start by retrieving the corresponding crime scene report from the police department's database. Write SQL queries to figure out who committed the crime.

This report utilizes information from the SQL City police database to examine and cross-reference data for the purpose of solving a murder that occurred on January 15, 2018. Given that the police station employs a SQLite database, queries were conducted using sqliteonline.com.

The subsequent steps were used to solve the murder investigation:

STEP 1: I queried the **crime_scene_report** table to see all columns.

```
SELECT *
```

```
FROM crime_scene_report;
```

The resultant table displayed columns with **date**, **type**, **description**, and **city**, respectively.

```
1 SELECT *
2 FROM crime_scene_report;
```

#	date	type	description	city
	20180115	robbery	A Man Dressed as Spider-Man Is on a...	NYC
	20180115	murder	Life? Dont talk to me about life.	Albany
	20180115	murder	Mama, I killed a man, put a gun again...	Reno
	20180215	murder	REDACTED REDACTED REDACTED	SQL City
	20180215	murder	Someone killed the guard! He took an ...	SQL City

STEP 2: I queried murders that occurred on the day of the **murder** in question, **January 15, 2018**, in **SQL City**.

```
SELECT *
```

```
FROM crime_scene_report
```

```
WHERE date = 20180115 AND type = 'murder' and city = 'SQL City';
```

The query showed one murder occurring in SQL City on January 15, 2018.

```

1 SELECT *
2 FROM crime_scene_report
3 WHERE date = 20180115 AND type = 'murder' AND city = 'SQL City';

```

date	type	description	city
20180115	murder	Security footage shows that there wer...	SQL City

The description for the murder states:

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".

STEP 3: Using the most detailed witness identification, I queried the **person table** to find an **"Annabel"** who lives on **"Franklin Ave."**

```

SELECT *
FROM person
WHERE address_street_name LIKE 'Franklin%' AND name LIKE 'Annabel%';

```

```

1 SELECT *
2 FROM person
3 WHERE address_street_name LIKE 'Franklin%' AND name LIKE 'Annabel%';

```

id	name	license_id	address_number	address_street_name	ssn
16371	Annabel Miller	490173	103	Franklin Ave	318771143

STEP 4: To locate the second witness, I queried the **person table** to find individuals living on **Northwestern Dr**, **ordering** results in **descending order** because the murder description stated that the witness lived in the last house on the street.

```

SELECT *
FROM person
WHERE address_street_name LIKE 'Northwestern_Dr%'
ORDER BY address_number DESC;

```

```

1 SELECT *
2 FROM person
3 WHERE address_street_name LIKE 'Northwestern_Dr%'
4 ORDER BY address_number DESC;

```

id	name	license_id	address_number	address_street_name	ssn
14887	Morty Schapiro	118009	4919	Northwestern Dr	111564949
17729	Lasonya Wildey	439686	3824	Northwestern Dr	917817122
53890	Sophie Tiberio	957671	3755	Northwestern Dr	442830147
73368	Torie Thalmann	773862	3697	Northwestern Dr	341559436

The results of the **person table** showed the occupant of the **last house** on **Northwestern Dr** was **Morty Schapiro**.

STEP 5: The person table showed that the two witnesses were **Annabel Miller** and **Morty Schapiro**. **Miller's id number** was **16371** and **Schapiro's 14887**. Using their id numbers, I queried the **interview table** to find their statements.

```
SELECT *  
  
FROM interview  
  
WHERE person_id = 16371 OR person_id = 14887;
```

```
1 SELECT *  
2 FROM interview  
3 WHERE person_id = 16371 OR person_id = 14887;
```

person_id	transcript
14887	I heard a gunshot and then saw a man run out. He had a "Get Fit Now Gym" bag. ...
16371	I saw the murder happen, and I recognized the killer from my gym when I was wor...

Morty Schapiro's transcript:

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's transcript:

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

STEP 6: Using the information provided by Miller and Schapiro, I joined the **get_fit_now_check_in** and **get_fit_now_member** tables via the **id/membership_id** keys to find **gold status** members who checked in on **January 9, 2018**, and had a membership number starting with **48Z**.

```
SELECT *  
  
FROM get_fit_now_member JOIN get_fit_now_check_in ON get_fit_now_member.id =  
get_fit_now_check_in.membership_id  
  
WHERE id LIKE '48Z%' AND membership_status IS 'gold' AND check_in_date =  
20180109;
```

```
1 SELECT *  
2 FROM get_fit_now_member JOIN get_fit_now_check_in ON get_fit_now_member.id = get_fit_now_check_in.membership_id  
3 WHERE id LIKE '48Z%' AND membership_status IS 'gold' AND check_in_date = 20180109;
```

id	person_id	name	membership_start_date	membership_status	membership_id	check_in_date	check_in_ti...	check_out
48Z7A	28819	Joe Germuska	20160305	gold	48Z7A	20180109	1600	1730
48Z55	67318	Jeremy Bowers	20160101	gold	48Z55	20180109	1530	1700

The result table shows two individuals who match the data points given by Miller and Schapiro, **Joe Germuska** and **Jeremy Bowers**.

STEP 7: Using **person_id** numbers, I cross-referenced the **interview table** to see if there was an interview from either suspect, **Joe Germuska (28819)** or **Jeremy Bowers (67318)**.

```
SELECT *  
  
FROM interview  
  
WHERE person_id IN (28819,67318);
```

```
1 SELECT *  
2 FROM interview  
3 WHERE person_id IN (28819,67318);  
4
```

person_id	transcript
67318	I was hired by a woman with a lot of money. I don't know her name but I know she's...

The **interview table** search returned one of the suspects, **Jeremy Bowers** who admits he committed the murder, however, he disclosed he was hired by a woman. Below is Bower's admission:

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.

STEP 8: Based on the information provided by the self-admitted hit-man, I checked the **drivers_license table** to find **women** who drive a **Tesla Model S** with **red hair** between **65 and 67 inches**.

```
SELECT *  
  
FROM drivers_license  
  
WHERE gender IS 'female' AND car_make IS 'Tesla' AND car_model IS 'Model S' AND  
hair_color IS 'red';
```

```
1 SELECT *  
2 FROM drivers_license  
3 WHERE gender IS 'female' AND car_make IS 'Tesla' AND car_model IS 'Model S' AND hair_color IS 'red';
```

id	age	height	eye_color	hair_color	gender	plate_number	car_make	car_model
202298	68	66	green	red	female	500123	Tesla	Model S
291182	65	66	blue	red	female	08CM64	Tesla	Model S
918773	48	65	black	red	female	917UU3	Tesla	Model S

The result table shows three suspects.

STEP 9: To narrow down the suspect list further, I checked the **facebook_event_checkin** to identify individuals who attended the **SQL Symphony three times** in **December 2017**. Jeremy Bowers stated his contractor attended this event in his police interview.

```
SELECT *  
  
FROM facebook_event_checkin  
  
WHERE event_name LIKE 'SQL Symphony Concert' AND date LIKE '201712%'  
  
GROUP BY person_id  
  
HAVING COUNT(person_id) = 3;
```

```
1 SELECT *  
2 FROM facebook_event_checkin  
3 WHERE event_name LIKE 'SQL Symphony Concert' AND date LIKE '201712%'  
4 GROUP BY person_id  
5 HAVING COUNT(person_id) = 3;
```

person_id	event_id	event_name	date
24556	1143	SQL Symphony Concert	20171207
99716	1143	SQL Symphony Concert	20171206

STEP 10: To identify the only two individuals that attended the **SQL Symphony Concert** in **December 2017**, I queried the **person table** using the **person_id** for both attendees.

```
SELECT *  
  
FROM person  
  
WHERE id IN (24556, 99716);
```

```
1 SELECT *  
2 FROM person  
3 WHERE id IN (24556, 99716);
```

id	name	license_id	address_number	address_street_name	ssn
24556	Bryan Pardo	101191	703	Machine Ln	816663882
99716	Miranda Priestly	202298	1883	Golden Ave	987756388

The **only woman** who attended the SQL Symphony 3 times in December of 2017 was **Miranda Priestly**.

STEP 11: By joining the **person** and **drivers_license tables** using the **id**(drivers_license table) and **license_id** (person table) keys I was able to verify that **Miranda Priestly** does drive a **Tesla Model S** and has **red hair**.

```

SELECT person.id, person.name, drivers_license.car_make, drivers_license.car_model,
drivers_license.height, drivers_license.gender, drivers_license.hair_color, person.license_id

FROM person

JOIN drivers_license ON person.license_id = drivers_license.id

WHERE person.id IN (99716);

```

```

1 SELECT person.id, person.name, drivers_license.car_make, drivers_license.car_model, drivers_license.height,
2 drivers_license.gender, drivers_license.hair_color, person.license_id
3 FROM person
4 JOIN drivers_license ON person.license_id = drivers_license.id
5 WHERE person.id IN (99716);

```

id	name	car_make	car_model	height	gender	hair_color	license_id
99716	Miranda Priestly	Tesla	Model S	66	female	red	202298

STEP 12: I joined the **income table** with the **person table** to see the annual income of **Miranda Priestly** because **Bowers**, the contract killer stated that she gave him “**a lot of money**”. A high income can support his claims. Using **primary key ssn to join income and personal tables** I was able to locate Miranda’s record.

```

SELECT person.name, person.ssn, person.license_id, income.annual_income,
person.address_number, person.address_street_name

FROM income

JOIN person ON income.ssn = person.ssn

WHERE id = 99716;

```

```

1 SELECT person.name, person.ssn, person.license_id, income.annual_income, person.address_number, person.address_street_name
2 FROM income
3 JOIN person ON income.ssn = person.ssn
4 WHERE id = 99716;

```

name	ssn	license_id	annual_income	address_number	address_street_name
Miranda Priestly	987756388	202298	310000	1883	Golden Ave

Conclusion: A **homicide** occurred in **SQL City** on **January 15, 2018**. Statements from witnesses led to the identification of **Jeremy Bowers** as the perpetrator, who confessed to being hired for the crime. Pursuing the leads provided by Bowers, I successfully traced **another suspect, Miranda Priestly**, residing on 1883 Golden Ave. She matches the description given by Bowers, being a **red-haired woman** with a **height** between **65** and **67 inches**, and drives a **Tesla Model S**.

Bowers mentioned that Priestly attended the **SQL Symphony three times in December 2017**, a detail verified by consulting the **facebook_events_checkin table**. Additionally, Bowers disclosed

receiving a substantial sum from the contractor for the murder. By linking the **income table** with the **drivers_license table** using the social security number, it was revealed that **Miranda Priestley** has an **income** of **\$310,000**. This income elucidates her capacity to attend multiple symphony concerts during the holiday season and finance the commissioned act resulting in the demise of the victim.