

SQL Murder Mystery



*gonna put murpy lawden here, cuz why not

Written By : Reza Donsika Putra

Link : <https://mystery.knightlab.com/>

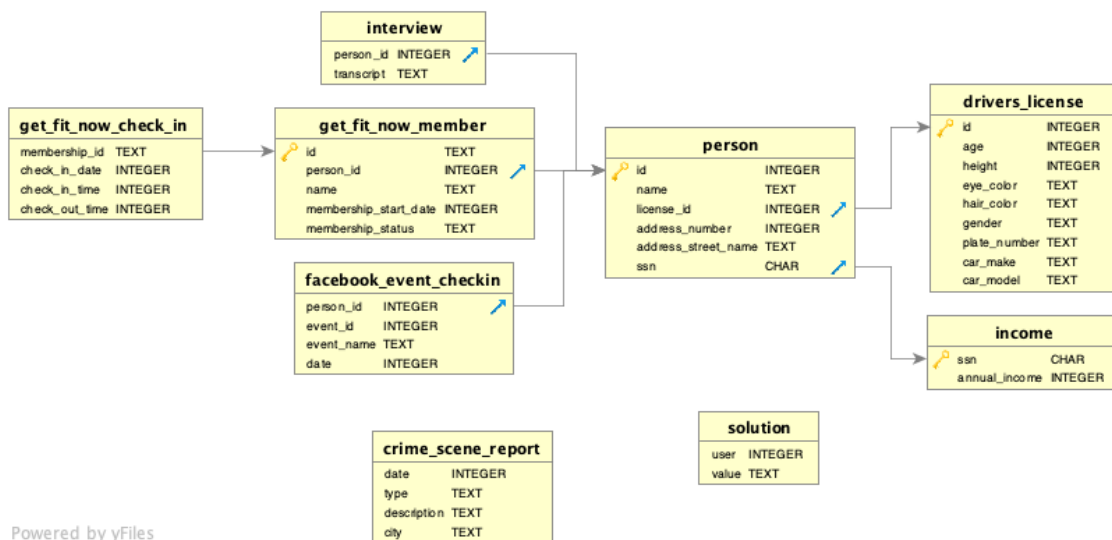
INTRODUCTION

There's been a Murder in SQL City! The SQL Murder Mystery is designed to be both a self-directed lesson to learn SQL concepts and commands and a fun game for experienced SQL users to solve an intriguing crime.

CASE AND OUR EVIDENCE

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.

SCHEMA



Powered by yFiles

I want to interrogate first at **murder, Jan.15, 2018, SQL City**. At crime_scene_report from our evidence.

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

RUN ↴

RESET

date	type	description	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

We get an information at the description, we got 2 information that there were 2 witnesses, **first witness lives at the last house on "Northwestern Dr"**. The **second witness, named Annabel, lives somewhere on "Franklin Ave"**.

We got 2 information:

1. First witness lives at the last house on "Northwestern Dr".
2. The second witness, named Annabel, lives somewhere on "Franklin Ave".

Let's interrogate the first witness, we'll check at the **person table** and **interview table**, so we need to join it.



First witness:

We don't know who was the first witness, so we get to check the address and sort the address_number into the latest.

```
1 SELECT *
2 FROM person
3 WHERE ( address_street_name = 'Northwestern Dr')
4 ORDER BY address_number DESC
5 LIMIT 1;
```

RUN ↴

RESET

id	name	license_id	address_number	address_street_name	ssn
14887	Morty Schapiro	118009	4919	Northwestern Dr	111564949

We get his id, name, license_id, address_number, and ssn.

Let's check his interview at **interview table** using his id and combine it with **person table** to make sure the data is valid.

```
1 SELECT per.name, inter.person_id, inter.transcript
2 FROM interview inter
3 JOIN person per ON inter.person_id = per.id
4 WHERE id = 14887;
```

RUN ↴

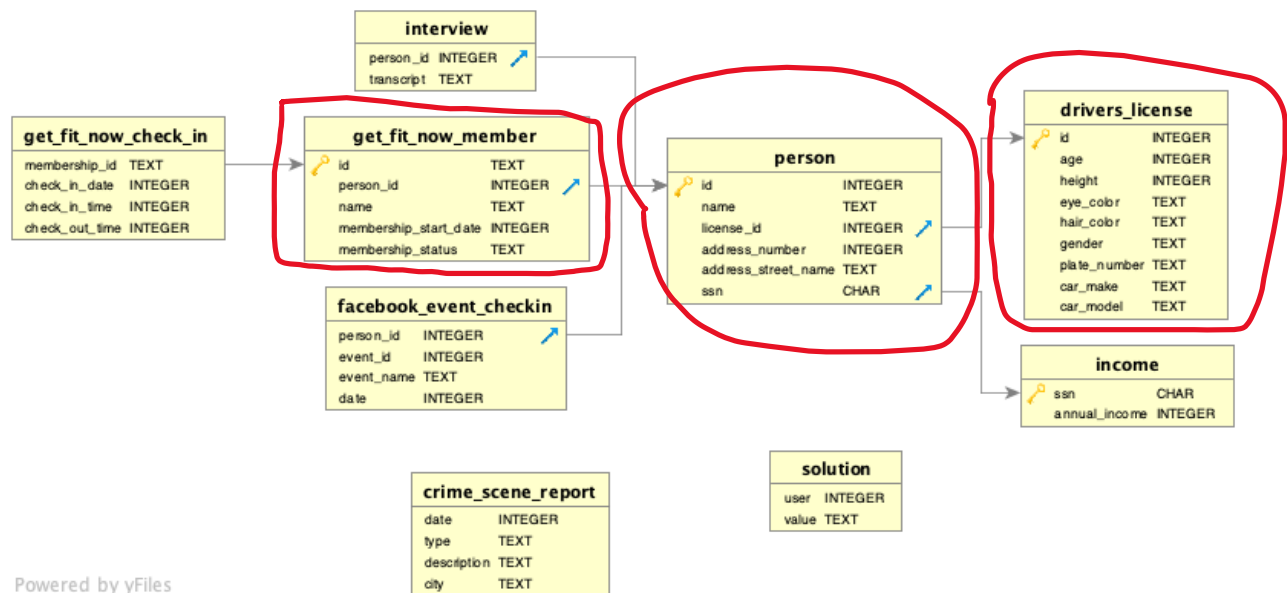
RESET

name	person_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".

Morty Schapiro (14887) said that "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".

We have a new information from Witness 1 which is Morty Schapiro (14887) :

1. That a suspect had a "Get Fit Now Gym" bag.
2. The membership number on the bag started with "48Z".
3. Only gold members have those bags.
4. The man got into a car with a plate that included "H42W".



Three tables that we need right now is **person table**, **get_fit_now_member table** and **drivers_license table**. Person_table to check the identity and validity of the suspect, get_fit_now_member table because the witness said that the suspect had a “Get Fit Now Gym” bag, and drivers_license table to check the plate_number of the car.

```

1 SELECT per.id, per.name, per.license_id, gym.id as gym_id,
2 gym.membership_status, lcs.plate_number
3 FROM person per
4 JOIN get_fit_now_member gym ON per.id = gym.person_id
5 JOIN drivers_license lcs ON per.license_id = lcs.id
6 WHERE gym.id LIKE '48Z%'
7 AND gym.membership_status = 'gold'
8 AND lcs.plate_number LIKE '%H42W%';

```

RUN ↴

RESET

id	name	license_id	gym_id	membership_status	plate_number
67318	Jeremy Bowers	423327	48Z55	gold	0H42W2

We get his id and his name, let's check **his interview**.

```

1 SELECT per.name, inter.person_id, inter.transcript
2 FROM interview inter
3 JOIN person per ON inter.person_id = per.id
4 WHERE id = 67318;

```

RUN ↴

RESET

name	person_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.

Second witness:

1. ~~First witness lives at the last house on "Northwestern Dr".~~
2. The second witness, named Annabel, lives somewhere on "Franklin Ave".

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

RUN ↴

RESET

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

We found her identity, let's check her interview.

```
1 SELECT per.name, inter.person_id, inter.transcript
2 FROM interview inter
3 JOIN person per ON inter.person_id = per.id
4 WHERE id = 16371;
```

RUN ↴

RESET

name	person_id	transcript
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.

She said that she recognized that the killer is from her gym on January the 9th. Let's check it.

```
1 SELECT
2   per.id, per.name, per.license_id, gym.id as gym_id,
3   gym.membership_status, SUBSTR(CAST(gym.membership_start_date AS CHAR), -4)
4   as membership_start_date
5 FROM
6   person per
7 JOIN
8   get_fit_now_member gym ON per.id = gym.person_id
9 WHERE
10  SUBSTR(CAST(gym.membership_start_date AS CHAR), -4) = '0109'
```

RUN ↴

RESET

id	name	license_id	gym_id	membership_status	membership_start_date
62367	Concha Fiveash	378266	300J1	regular	0109

Let's check her interview.

```
1 SELECT per.name, inter.person_id, inter.transcript
2 FROM interview inter
3 JOIN person per ON inter.person_id = per.id
4 WHERE id = 62367;
```

RUN ↴

RESET

name	person_id	transcript
Concha Fiveash	62367	

We don't have any clue on her interview. Let's check on her facebook_event_checkin.

```
1 SELECT *
2 FROM facebook_event_checkin
3 WHERE person_id = 62367
```

RUN ↴

RESET

person_id	event_id	event_name	date
62367	3841	None. The Universe spines the bulb, and the Zen master stays out of	20170120
62367	9288	-- Mark Twain	20170509

She only attend 2 event, and none of it seems suspicious and give us a clue.

Conclusion (Murderer) :

Suspect 1 from Witnesses 1

Did you find the killer?

```
1 INSERT INTO solution VALUES (1, 'Jeremy Bowers');  
2  
3 SELECT value FROM solution;
```

RUN ↴

RESET

value

Congrats, you found the murderer! But wait, there's more... If you think you're up for a challenge, try querying the interview transcript of the murderer to find the real villain behind this crime. If you feel especially confident in your SQL skills, try to complete this final step with no more than 2 queries. Use this same INSERT statement with your new suspect to check your answer.

Suspect 2 from Witnesses 2

Did you find the killer?

```
1 INSERT INTO solution VALUES (1, 'Concha Fiveash');  
2  
3 SELECT value FROM solution;
```

RUN ↴

RESET

value

That's not the right person. Try again!

We found the murderer, he is **Jeremy Bowers**, but we didn't caught up who's the real villain behind this crime.

Find the Villain :

```
1 SELECT per.name, inter.person_id, inter.transcript
2 FROM interview inter
3 JOIN person per ON inter.person_id = per.id
4 WHERE id = 67318;
```

RUN ↴

RESET

name	person_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.

From the murderer, He said that he was hired by a woman with a lot of money, let's collect the information first from his transcript interview:

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

Let's roll, Murphy. We got a new suspect, it's a rich woman.



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

First Suspect (based on Information 1 – 3):

```
1 SELECT per.id, per.name, per.license_id,  
2 lcs.gender, lcs.height, lcs.hair_color, lcs.car_make, lcs.car_model  
3 FROM person per  
4 JOIN drivers_license lcs ON per.license_id = lcs.id  
5 WHERE lcs.gender = 'female'  
6 AND (lcs.height >= 65 AND lcs.height <= 67)  
7 AND lcs.hair_color = 'red'  
8 AND lcs.car_make = 'Tesla'  
9 AND lcs.car_model = 'Model S';
```

RUN ↴

RESET

id	name	license_id	gender	height	hair_color	car_make	car_model
78881	Red Korb	918773	female	65	red	Tesla	Model S
90700	Regina George	291182	female	66	red	Tesla	Model S
99716	Miranda Priestly	202298	female	66	red	Tesla	Model S

Second Suspect (based on Information 4) :

```
1 SELECT
2   per.id, per.name, fb.event_id, fb.event_name,
3   SUBSTR(CAST(fb.date AS CHAR), 1, 6) as date,
4   COUNT(*) as event_attends
5 FROM
6   person per
7 JOIN
8   facebook_event_checkin fb ON per.id = fb.person_id
9 WHERE
10  event_name = 'SQL Symphony Concert'
11  AND
12  SUBSTR(CAST(fb.date AS CHAR), 1, 6) = '201712'
13 GROUP BY
14  per.name
15 HAVING
16  event_attends = 3;
```

RUN ↴

RESET

id	name	event_id	event_name	date	event_attends
24556	Bryan Pardo	1143	SQL Symphony Concert	201712	3
99716	Miranda Priestly	1143	SQL Symphony Concert	201712	3

Let's check the gender

```
1 SELECT
2   per.id, per.name, fb.event_id, fb.event_name,
3   SUBSTR(CAST(fb.date AS CHAR), 1, 6) as date,
4   COUNT(*) as event_attends,
5   lcs.gender
6 FROM
7   person per
8 JOIN
9   facebook_event_checkin fb ON per.id = fb.person_id
10 JOIN
11  drivers_license lcs ON lcs.id = per.license_id
12 WHERE
13  event_name = 'SQL Symphony Concert'
14  AND
15  SUBSTR(CAST(fb.date AS CHAR), 1, 6) = '201712'
16 GROUP BY
17  per.name
18 HAVING
19  event_attends = 3;
```

RUN ↴

RESET

id	name	event_id	event_name	date	event_attends	gender
99716	Miranda Priestly	1143	SQL Symphony Concert	201712	3	female

Final Query (Find the Villain) :

```
1 SELECT
2     per.id, per.name,
3     lcs.gender, lcs.height, lcs.hair_color, lcs.car_make, lcs.car_model,
4     fb.event_id, fb.event_name, SUBSTR(CAST(fb.date AS CHAR), 1, 6) as fb_date,
5     COUNT(*) as event_attends
6 FROM
7     person per
8 JOIN
9     drivers_license lcs ON per.license_id = lcs.id
10 JOIN
11     facebook_event_checkin fb ON per.id = fb.person_id
12 WHERE
13     lcs.gender = 'female'
14     AND
15     (lcs.height >= 65 AND lcs.height <= 67)
16     AND
17     lcs.hair_color = 'red'
18     AND
19     lcs.car_make = 'Tesla'
20     AND
21     lcs.car_model = 'Model S'
22     AND
23     event_name = 'SQL Symphony Concert'
24     AND
25     SUBSTR(CAST(fb.date AS CHAR), 1, 6) = '201712'
26 GROUP BY
27     per.name
28 HAVING
29     event_attends = 3;
```

Result :

id	name	gender	height	hair_color	car_make	car_model	event_id	event_name	fb_date	e
99716	Miranda Priestly	female	66	red	Tesla	Model S	1143	SQL Symphony Concert	201712	e

<		>
---	--	---

e	gender	height	hair_color	car_make	car_model	event_id	event_name	fb_date	event_attends
anda stly	female	66	red	Tesla	Model S	1143	SQL Symphony Concert	201712	3

<		>
---	--	---

Villain :

Check your solution

Did you find the killer?

```
1 INSERT INTO solution VALUES (1, 'Miranda Priestly');  
2  
3 SELECT value FROM solution;
```

RUN ↴

RESET

value

Congrats, you found the brains behind the murder! Everyone in SQL City hails you as the greatest SQL detective of all time. Time to break out the champagne!

Fun Fact :

(I didn't expect to find him in the facebook_event_checkin tbh while I was looking at this table)

Let's check at facebook_event_checkin table

person_id	event_id	event_name	date
28508	5880	Nudists are people who wear one-button suits.	20170913
63713	3865	but that's because it's the best book on anything for the layman.	20171009
63713	3999	If Murphy's Law can go wrong, it will.	20170502
63713	6436	Old programmers never die. They just branch to a new address.	20170926
82998	4470	Help a swallow land at Capistrano.	20171022

What do we got here, Murphy :D