Duygu Jones

SQL Murder Mystery

Can you find out whodunnit?

Walkthrough for SQL Beginners

If you're comfortable with SQL, you can skip these explanations and put your skills to the test! Below we introduce some basic SQL concepts, and just enough detail to solve the murder. If you'd like a more complete introduction to SQL, try Select Star SQL.

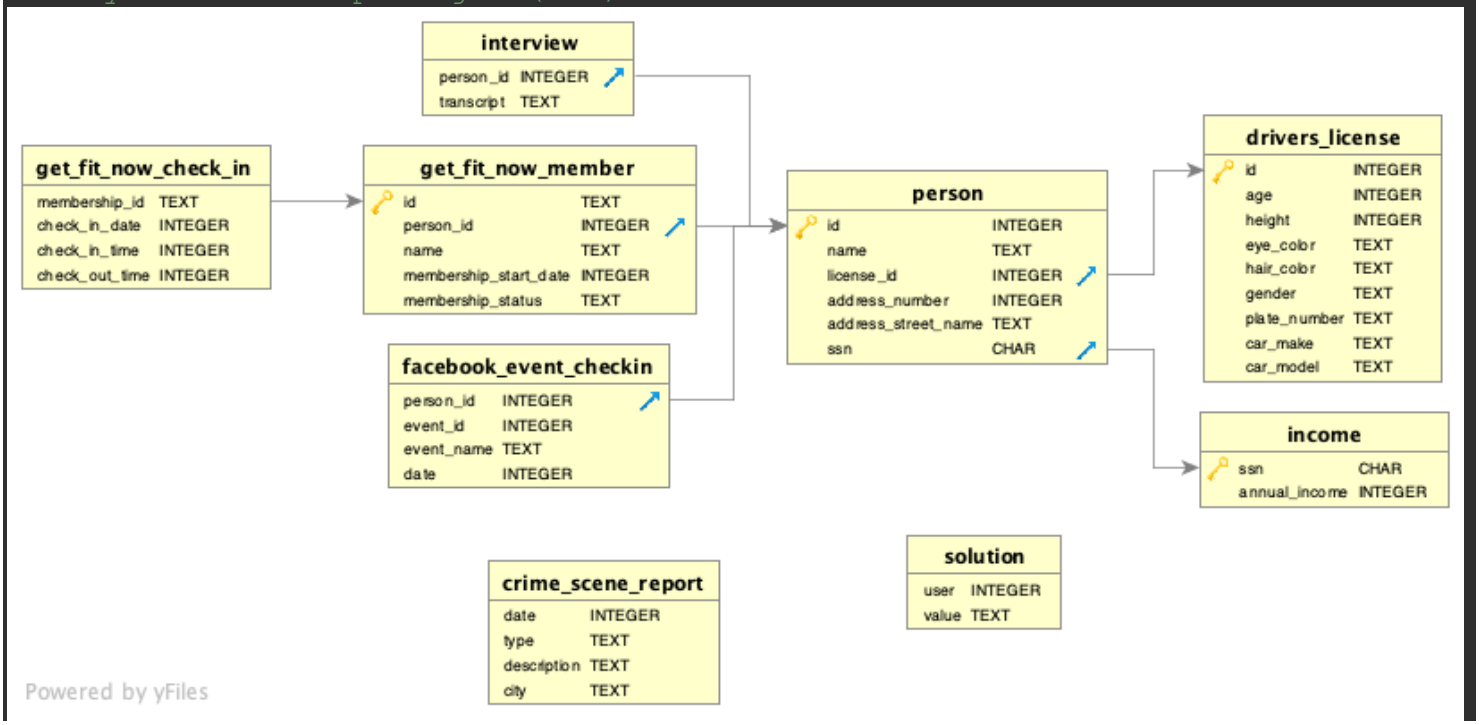
A crime has taken place and the detective needs your help. The detective gave you the crime scene report, but you somehow lost it. You vaguely remember that the crime was a **murder** that occurred sometime on **Jan.15, 2018** and that it took place in **SQL City.** Start by retrieving the corresponding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

All the clues to this mystery are buried in a huge database, and you need to use SQL to navigate through this vast network of information.

Your first step to solving the mystery is to retrieve the corresponding crime scene report from the **police department's database.

Resource: https://mystery.knightlab.com/walkthrough.html

Entity Relationship Diagram(ERD)



```sql
--1. Clue:
-- murder that occurred sometime on Jan.15, 2018
-- and that it took place in SQL City

-- Find Information about Witnesses in crime_scene_report table;
SELECT *
FROM crime_scene_report
WHERE type = 'murder'
AND city = 'SQL City'
AND date = 20180115;
```

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

```sql
--2. Clue:
-- Security footage shows that there were 2 witnesses.
-- 1st witness lives at the last house on "Northwestern Dr".
-- 2nd witness, named Annabel, lives somewhere on "Franklin Ave".

-- Find details about the witnesses in person table;
-- 1st Witness:
SELECT *
FROM person
WHERE address_street_name = 'Northwestern Dr'
ORDER BY address_number DESC
LIMIT 1;
```

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

```sql
-- 2nd Witness:
SELECT *
FROM person
WHERE address_street_name = 'Franklin Ave'
AND LOWER(name) LIKE '%annabel%'
```

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

```sql
--3. Clue:
-- Check the Witnesses with their person_id from the interview table
-- to get the information from the transcript;
SELECT *
FROM interview
WHERE person_id IN (14887,16371);
```

person_id	transcript
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".
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.

```sql
--4. Clue:
-- "Get Fit Now Gym" bag started with "48Z" -> gold member
-- car with a plate that included "H42W"
-- Witness saw the guy at the gym last week on January the 9th

-- Using details from interview table find the killer;
SELECT
     person.*,
     gym_cin.*
FROM drivers_license AS license
INNER JOIN person ON license.id = person.license_id
INNER JOIN get_fit_now_member AS gym ON gym.person_id = person.id
INNER JOIN get_fit_now_check_in AS gym_cin ON gym_cin.membership_id = gym.id
WHERE plate_number LIKE '%H42W%'
AND membership_status ='gold'
AND check_in_date =20180109;
```

id	name	license_id	address_number	address_street_name	ssn	membership_id	che
67318	Jeremy Bowers	423327	530	Washington Pl, Apt 3A	871539279	48Z55	20

```sql
-- Check the solution;
INSERT INTO solution VALUES (1, 'Jeremy Bowers');

SELECT value FROM solution;
```

Did you find the killer?

```sql
INSERT INTO solution VALUES (1, 'Jeremy Bowers');

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.

```sql
--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.


--5. Clue:
-- Check out the killer's interview transcript
SELECT *
FROM interview
WHERE person_id = 67318;
```

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 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.

```sql
--6. Clue:
-- Murderer hired by a woman with a lot of money
-- she's around 5'5" (65") or 5'7" (67") height
-- red hair and she drives a Tesla Model S
-- she attended the SQL Symphony Concert 3 times in December 2017.

-- Using details find the women who hired the Murderer;
-- Check out'drivers_license' table to find the car, height, hair color etc.
SELECT *
FROM drivers_license
WHERE hair_color ='red'
AND height BETWEEN 65 AND 67
AND gender = 'female'
AND car_make = 'Tesla'
AND car_model = 'Model S'
```

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

```sql
-- Check out 'facebook_event_checkin' table for concert and person_id etc.
SELECT
     person_id,
     event_name,
     COUNT(*) times
FROM facebook_event_checkin
WHERE date BETWEEN 20171201 AND 20171231
AND event_name ='SQL Symphony Concert'
GROUP BY person_id
HAVING COUNT(*) >= 3;
```

person_id	event_name	times
24556	SQL Symphony Concert	3
99716	SQL Symphony Concert	3

```sql
-- Join the tables; person + facebook_event_checkin

SELECT person.*, concert.*
FROM person
INNER JOIN facebook_event_checkin AS concert
ON concert.person_id = person.id
WHERE date BETWEEN 20171201 AND 20171231
AND event_name ='SQL Symphony Concert'
GROUP BY person_id
HAVING COUNT(*) >= 3
```

name	license_id	address_number	address_street_name	ssn	person_id	event_id	event_n
Bryan Pardo	101191	703	Machine Ln	816663882	24556	1143	SQL Symphc Concert
Miranda Priestly	202298	1883	Golden Ave	987756388	99716	1143	SQL Symphc Concert

```sql
-- YOU ARE SO CLOSE!
-- One more step--->


-- You can use Common Table Expression (CTE) to find the information about
concert
-- Then joined it to person table filtered by description given;

WITH CTE AS(
SELECT
person_id,
event_name,
COUNT(*) times
FROM facebook_event_checkin
WHERE date BETWEEN 20171201 AND 20171231
AND event_name ='SQL Symphony Concert'
GROUP BY person_id
HAVING COUNT(*) >= 3
)

SELECT person.*, concert.*
FROM drivers_license as license
INNER JOIN person ON license.id = person.license_id
INNER JOIN CTE AS concert ON concert.person_id = person.id
WHERE hair_color ='red'
AND height BETWEEN 65 AND 67
AND gender = 'female'
AND car_make = 'Tesla'
AND car_model = 'Model S'
```

name	license_id	address_number	address_street_name	ssn	person_id	event_name	time
Miranda Priestly	202298	1883	Golden Ave	987756388	99716	SQL Symphony Concert	3

```
-- Check the solution;
INSERT INTO solution VALUES (1, 'Miranda Priestly');

SELECT value FROM 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!
