The SQL Murder Mystery

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.

[M u h a m m a d   A s i f]

Experienced SQL sleuths start here

A crime has taken place and the detective needs your help. The detective gave you the crime scene report, but we 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.

Exploring the Database Structure

Experienced SQL users can often use database queries to infer the structure of a database. But each database system has different ways of managing this information. The SQL Murder Mystery is built using SQLite. Use this SQL command to find the tables in the Murder Mystery database. Run this query to find the names of the tables in this database.

This command is specific to SQLite. For other databases, we'll have to learn their specific syntax.

```
SELECT name
FROM sqlite_master
where type = 'table'
```

Besides knowing the table names, we need to know how each table is structured. The way this works is also dependent upon which database technology we use. Here's how we do it with SQLite.

Entity Relationship Diagram or The schema diagram.
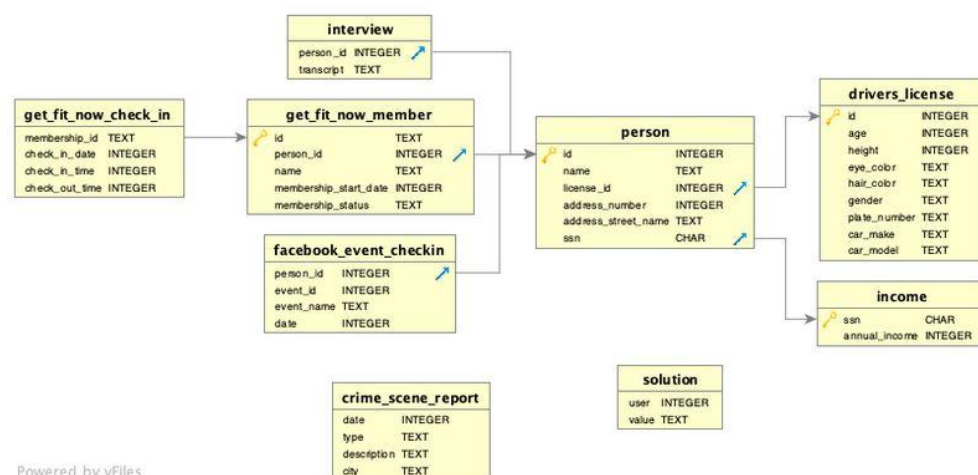
START

Let's find out who committed the murder.

-- Given information
-- Crime Type ::   murder
-- Time   ::   Jan.15, 2018
-- Place   ::   SQL City

Code:01

```sql
select * From crime_scene_report
where type = 'murder'
and date = '20180115'
and city = 'SQL City'
```

OUTOUT:

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

---

-- Find the 2 Witnesses
-- Witnesses   :: 2
-- 1st Witness   :: lives at last house on "Northwestern Dr"
-- 2nd Witness   :: Annabel, lives somewhere on "Franklin Ave"

Code:01

```
select * From person
where address_street_name = 'Northwestern Dr'
order by address_number desc -- as given the last house on Northwestern
limit 100
```

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

Code:02

```
select * From person
where address_street_name = 'Franklin Ave'
and name like '%Annabel%' -- to search for any name contain Annabel word
limit 100
```

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

Code:03

```
Select * From person
where id in (14887, 16371)
```

OUTOUT:

id	name	license_id	address_number	address_street_name	ssn
14887	Morty Schapiro	118009	4919	Northwestern Dr	111564949
16371	Annabel Miller	490173	103	Franklin Ave	318771143

Results:
- 16371 Annabel Miller
- 14887 Morty Schapiro

-- Interview of Two Witnesses

Code:01

```
select * From interview
where person_id in (14887,16371)
```

OUTOUT:

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.

Evidence:
- "Get Fit Now Gym" bag
- membership number on the bag started with "48Z"
- gold members
- Car number plate that included "H42W"
- data : last week on January the 9th

Code:02

```
select p.*, gf.*,ci.* From drivers_license as dl
inner join person as p on dl.id = p.license_id
inner join get_fit_now_member as gf on p.id = gf.person_id
inner join get_fit_now_check_in as ci on gf.id = ci.membership_id
where plate_number like '%H42W%'
and membership_status = 'gold'
```

OUTOUT:

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

Results:

- Name of murder is **Jeremy Bomwers**

Check Code

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

SELECT value FROM solution;

Results:

- we solve the mysery that "**Jeremy Bomwers**" is a murderer

---

OUTOUT:

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

‹   ›

Check your solution

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.

- The Real master-mind behind the murder is not Jeremy Bomwers so lets check his interview
- ID :67318
- Name: Jeremy Browers

Code:01

select * From interview
where person_id = '67318'

OUTOUT:

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.

Eivdence provided by Jeremy Browers

- Gender : woman
- height : 5'5" (65") or 5'7" (67")
- Hair color : red
- car model : Tesla Model S
- she attended the SQL Symphony Concert 3 times in December 2017

Code:02

select * from drivers_license
where hair_color = 'red'
and gender = 'female'
and car_make like '%Tesla%'
and car_model like '%model%'
and height between '65' and '67'
limit 100

OUTOUT:

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

Results:

- id : 202298, 291182, 918773

Code:03

```
select p. * from drivers_license as dl
inner join person as p on dl.id = p.license_id
where hair_color = 'red'
and gender = 'female'
and car_make like '%Tesla%'
and car_model like '%model%'
and height between '65' and '67'
limit 100
```

OUTOUT:

id	name	license_id	address_number	address_street_name	ssn
78881	Red Korb	918773	107	Camerata Dr	961388910
90700	Regina George	291182	332	Maple Ave	337169072
99716	Miranda Priestly	202298	1883	Golden Ave	987756388

- she attended the SQL Symphony Concert 3 times in December 2017

Code:01

```
select
person_id,
event_name,
count(*)
From facebook_event_checkin
where date between 20171201 and 20171231
and event_name = 'SQL Symphony Concert'
group by person_id
having count(*) >=3
```

OUTOUT:

person_id	event_name	count(*)
24556	SQL Symphony Concert	3
99716	SQL Symphony Concert	3

Code:02

```
with cte as
(
select
person_id,
event_name,
count(*)
From facebook_event_checkin
where date between 20171201 and 20171231
and event_name = 'SQL Symphony Concert'
group by person_id
having count(*) >=3
)

select p.*, fb.* from drivers_license as dl
inner join person as p on dl.id = p.license_id
inner join cte as fb on fb.person_id = p.id
where hair_color = 'red'
and gender = 'female'
and car_make like '%Tesla%'
and car_model like '%model%'
and height between '65' and '67'
limit 100
```

OUTOUT:

id	name	license_id	address_number	address_street_name	ssn	person_id	coun
99716	Miranda Priestly	202298	1883	Golden Ave	987756388	99716	3

Check Code

INSERT INTO solution VALUES (1, 'Miranda Priestly');

SELECT value FROM solution;

OUTOUT:

id	name	license_id	address_number	address_street_name	ssn	person_id	coun
99716	Miranda Priestly	202298	1883	Golden Ave	987756388	99716	3

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!

Results:
- the brains behind the murder is **Miranda Priestly**

Congratulation we Solve the SQL Mystery