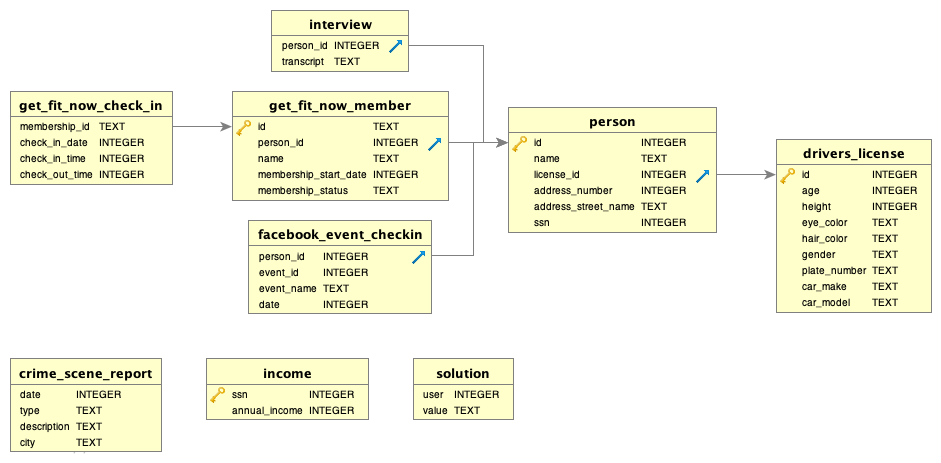
**This is a solo case project regarding the data analysis, especially with SQL language.**

The database has been located at the quarters of **https://mystery.knightlab.com/**

Without further do, let’s jump to it!

**THE DATASET…**

The dataset itself, it consists of multiple relation table (of course, it’s SQL anyway) as we can see from schema below:



**The SQL Murder Mystery — Background Problem**

***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.***

***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.***

**The aim of this challenge is to find the killer with only using information from the given dataset of tables.**

**Let’s get to work sherlock!**

**Crime Scene**

Ας real detectives, our very first step is to go to the crime scene and gathering information.

As we know from the background story, the murder occured on Jan 15, 2018 and took place in SQL City. So we need to get information from crime\_scene\_report table with our very first query:

SELECT date,type,city,description

FROM crime\_scene\_report

where date='20180115'

and city = 'SQL City'

and type='murder'

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Type** | **city** | **description** |
| 20180115 | Murder | SQL City | 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". |

The query result telling us that there was two witnesses when the murder happened:

* First witness. Lives at the last house of Northwestern Dr
* Second witness. Named Annabel, lives somewhere on Franklin Ave

With this information, we can move forward to find our witnesses identities.

**1st Witness Information**

To get this info we have to match it with person table with query:

SELECT date,type,city,description,id,name,address\_number, address\_street\_name

FROM crime\_scene\_report, person

where date='20180115'

and city = 'SQL City'

and type='murder'

and address\_street\_name in ('Northwestern Dr','Franklin Ave')

group by address\_number

The above query has presented the 2 address that are witnesses are staying but we don’t know the address number. Grouping by address number has helped me to find the last number for 'Northwestern Dr' street, as we know our first witness is staying at .

And then …..

SELECT date,type,city,description,id,name,address\_number, address\_street\_name transcript

FROM crime\_scene\_report, person, interview

where date='20180115'

and city = 'SQL City'

and type='murder'

and address\_street\_name ='Northwestern Dr'

and address\_number='4919'

group by address\_number

| **date** | **type** | **city** | **Description** | **id** | **name** | **address\_number** | **transcript** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 20180115 | murder | SQL City | 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". | 14887 | Morty Schapiro | 4919 | Northwestern Dr |

**2nd Witness Information**

SELECT date,type,city,description,id,name,address\_number, address\_street\_name ,person\_id,transcript

FROM crime\_scene\_report, person, interview

where date='20180115'

and city = 'SQL City'

and type='murder'

and address\_street\_name ='Franklin Ave'

and person\_id ='16371'

and name ='Annabel Miller'

| **date** | **type** | **city** | **description** | **id** | **name** | **address\_number** | **address\_street\_name** | **person\_id** | **transcript** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20180115 | murder | SQL City | 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". | 16371 | Annabel Miller | 103 | Franklin Ave | 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. |

**Witnesses Interview**

The next step is we conduct interview with our witnesses, so we can access interview table with query:

SELECT person.name, interview.transcript

FROM person JOIN interview

ON person.id = interview.person\_id

WHERE person.id = 14887 OR person.id = 16371;

| **Name** | **Transcript** |
| --- | --- |
| Morty Schapiro | 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 | I saw the murder happen, and I recognized the killer from my gym when I was working out last week on January the 9th. |

So, both our witnesses saw about what happened at the murder scene and gives us insights:

* Gun murder
* Suspect is a gym member ‘Get Fit Now’ Gym, Gold member, member number starts with ‘48Z’
* Suspect escaped with car. The car plate has ‘H42W’ characters
* Suspect last visit to the gym before murder was on January 9th 2018

These are very good tip!

We can move on to analyze the information.

**Dig Deeper**

We can start from the Gym data, using get\_fit\_now\_member table and get\_fit\_now\_check\_in table. We join this both table using JOIN clause:

SELECT \* FROM get\_fit\_now\_member AS gm

JOIN get\_fit\_now\_check\_in AS gc

ON gm.id = gc.membership\_id

WHERE id LIKE '%48Z%'

AND membership\_status = 'gold';

After we JOIN the table using membership id as the key, we can filter the table using witnesses information. Using LIKE clause to match the ‘48Z’ member code from the bag and the membership status ‘gold’ we get the output of the query:

| **id** | **person\_id** | **name** | **membership\_start\_date** | **membership\_status** | **membership\_id** | **check\_in\_date** | **check\_in\_time** | **check\_out\_time** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 48Z7A | 28819 | Joe Germuska | 20160305 | gold | 48Z7A | 20180109 | 1600 | 1730 |
| 48Z55 | 67318 | Jeremy Bowers | 20160101 | gold | 48Z55 | 20180109 | 1530 | 1700 |

This gives us two suspects that match witnesses testimony. Both checked in the same date in 9th January 2018, same membership status, and also has similiar code of gym membership ID.

With using the same witnesses info and gym info, we know that:

* Suspect is male
* Suspect escaped with car.
* The car plate number has ‘H42W’ characters

We need to narrow down this information using drivers\_license table with query:

SELECT \* FROM drivers\_license

WHERE plate\_number LIKE '%H42W%'

AND gender = 'male'

| **id** | **age** | **height** | **eye\_color** | **hair\_color** | **gender** | **plate\_number** | **car\_make** | **car\_model** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4233 | 2730 | 70 | brown | brown | male | 0H42W2 | Chevrolet | Spark LS |
| 6647 | 6021 | 71 | black | black | male | 4H42WR | Nissan | Altima |

To avoid confusion regarding id number, here’s some useful remarks:

* Id in person table is like identification number of a person. Consist of 5 digits of integer
* Driver license id in drivers\_license table is the drivers license unique ID of a person. Consist of 6 digits of integer.

Next step is we need to match this two data with the person table in order to get the person’s information that own these cars using the license id. The query for it:

SELECT p.id, name, plate\_number FROM drivers\_license AS d   
JOIN person AS p  
ON d.id = p.license\_id  
WHERE license\_id IN  
(SELECT id FROM drivers\_license  
WHERE plate\_number LIKE '%H42W%'  
AND gender = 'male');

For this case we using subquery method to directly filter the result with using previous query in driver license table.

* First, we JOIN the drivers\_license table and person table using license\_id as key.
* Second, using WHERE clause to filter the license\_id with IN clause that consist of SELECT clause from previous query inside the parentheses.

This resulting the matched id to be filtered in the first SELECT clause.

| **Id** | **name** | **plate\_number** |
| --- | --- | --- |
| 51739 | Tushar Chandra | 4H42WR |
| 67318 | Jeremy Bowers | 0H42W2 |

We got matched person! Jeremy Bowers is our main suspect!

## Moment of Truth

Now we need to check to the solution table whether our suspect is right or not.

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

SELECT value FROM solution;

| **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. |

Uh oh! We only got the executioner. We must get the mastermind who ordered the murder as well!! So what are we waiting for?? Let’s get to work lads!

## Aim For The Head

## If we want to capture the mastermind, we need to do it fast. Timing is essential because he/she will flee sooner or later.

To to this, we can again using subquery to minimize our search to be packed in a single query:

First query, to get the interview transcript from our executioner, in order to get information regarding our mastermind.

Second query, to get the the mastermind identity.

## 1st Query

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. |

We got insights of our criminal mastermind identity:

* SHE is a rich woman
* Around 65 to 67 inch height
* Red haired
* Driving Tesla model S
* Attend SQL Symphony Concert for 3 times in Dec 2017

## 2nd Query

With all those information from the executioner we can narrow down our search. In order to make subquery easier, we can breakdown the query from ‘child’ subquery and then stacked it up until ‘parent’ query, resulting a greater single query.

For this query we can narrow down from driver\_license table.

SELECT height,hair\_color,car\_model,id,gender

from drivers\_license

where height between 65 and 67

and hair\_color = 'red'

and car\_model = 'Model S'

and gender ='female'

| **height** | **hair\_color** | **car\_model** | **Id** | **gender** |
| --- | --- | --- | --- | --- |
| 66 | red | Model S | 202298 | female |
| 66 | red | Model S | 291182 | female |
| 65 | red | Model S | 918773 | female |

This query get us detail about which driver id that are female, red hair, and driving Tesla. Then we can continue to our ‘parent’ subquery that using child query to get info of a person id that has previous details in child query, and also attend in ‘SQL Symphony Concert’ from facebook\_event\_checkin table.

seLECT \* from facebook\_event\_checkin

where event\_name='SQL Symphony Concert'

and date between '20171201' and '20171231'

and person\_id IN

(SELECT person.id FROM person

JOIN drivers\_license

ON person.license\_id = drivers\_license.id

WHERE height BETWEEN 65 AND 67

AND gender = 'female'

AND hair\_color = 'red'

AND car\_make = 'Tesla')

| **person\_id** | **event\_id** | **event\_name** | **date** |
| --- | --- | --- | --- |
| 99716 | 1143 | SQL Symphony Concert | 20171206 |
| 99716 | 1143 | SQL Symphony Concert | 20171212 |
| 99716 | 1143 | SQL Symphony Concert | 20171229 |

This subquery resulted only a person id, but we need to know the person name with person table. Thus completed our subquery:

SELECT id, name FROM person

WHERE id =

(SELECT person\_id from facebook\_event\_checkin

where event\_name='SQL Symphony Concert'

and date between '20171201' and '20171231'

and person\_id IN

(SELECT person.id FROM person

JOIN drivers\_license

ON person.license\_id = drivers\_license.id

WHERE height BETWEEN 65 AND 67

AND gender = 'female'

AND hair\_color = 'red'

AND car\_make = 'Tesla'

)

);

| **id** | **name** |
| --- | --- |
| 99716 | Miranda Priestly |

Miranda Priestly. Is this the mastermind that we looking for?? Time to check to the solution table!

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

SELECT value FROM solution;

| **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! |

VOILA! Finally we apprehend the executioner and the mastermind of this murder. Congratulations detective! We deserve a long break for a day!

