

1. Context

As part of Harvard's introduction to computer science course, I completed a problem set where the task was to identify specific people by analysing multiple tables in a dataset using SQL techniques.

2. Ask

An item of value has been stolen! The townspeople have called upon you to solve the mystery of the stolen duck. Authorities believe that the thief stole a duck and then, shortly afterwards, took a flight out of town with the help of an accomplice. The goal is to identify:

- Who the thief is,
- What city the thief escaped to, and
- Who the thief's accomplice is who helped them escape
- All you know is that the theft took place on July 28, 2021 and that it took place on Humphrey Street.



DATA TABLES

```
CREATE TABLE crime_scene_reports (  
  id INTEGER,  
  year INTEGER,  
  month INTEGER,  
  day INTEGER,  
  street TEXT,  
  description TEXT,  
  PRIMARY KEY(id)
```

```
CREATE TABLE interviews (  
  id INTEGER,  
  name TEXT,  
  year INTEGER,  
  month INTEGER,  
  day INTEGER,  
  transcript TEXT,  
  PRIMARY KEY(id)
```

```
CREATE TABLE flights (  
  id INTEGER,  
  origin_airport_id INTEGER,  
  destination_airport_id INTEGER,  
  year INTEGER,  
  month INTEGER,  
  day INTEGER,  
  hour INTEGER,  
  minute INTEGER,  
  PRIMARY KEY(id),  
  FOREIGN KEY(origin_airport_id)  
  REFERENCES airports(id),  
  FOREIGN  
  KEY(destination_airport_id)  
  REFERENCES airports(id)
```

```
CREATE TABLE passengers (  
  flight_id INTEGER,  
  passport_number INTEGER,  
  seat TEXT,  
  FOREIGN KEY(flight_id)  
  REFERENCES flights(id)
```

```
CREATE TABLE bank_accounts (  
  account_number INTEGER,  
  person_id INTEGER,  
  creation_year INTEGER,  
  FOREIGN KEY(person_id)  
  REFERENCES people(id)
```

```
CREATE TABLE people (  
  id INTEGER,  
  name TEXT,  
  phone_number TEXT,  
  passport_number INTEGER,  
  license_plate TEXT,  
  PRIMARY KEY(id)
```

```
CREATE TABLE airports (  
  id INTEGER,  
  abbreviation TEXT,  
  full_name TEXT,  
  city TEXT,  
  PRIMARY KEY(id)
```

```
CREATE TABLE atm_transactions (  
  id INTEGER,  
  account_number INTEGER,  
  year INTEGER,  
  month INTEGER,  
  day INTEGER,  
  atm_location TEXT,  
  transaction_type TEXT,  
  amount INTEGER,  
  PRIMARY KEY(id)
```

```
CREATE TABLE bakery_security_logs (  
  id INTEGER,  
  year INTEGER,  
  month INTEGER,  
  day INTEGER,  
  hour INTEGER,  
  minute INTEGER,  
  activity TEXT,  
  license_plate TEXT,  
  PRIMARY KEY(id)
```

```
CREATE TABLE phone_calls (  
  id INTEGER,  
  caller TEXT,  
  receiver TEXT,  
  year INTEGER,  
  month INTEGER,  
  day INTEGER,  
  duration INTEGER,  
  PRIMARY KEY(id)
```

By examining the available tables, I noticed there is a crime scene reports table. Based on the information provided about the crime, I started by requesting the relevant crime report filtering by the date and location of the incident.

```
SELECT description FROM crime_scene_reports WHERE month = 7 AND day = 28 AND street = "Humphrey Street";
```

Theft of the CS50 duck took place at 10:15am at the Humphrey Street bakery. Interviews were conducted today with three witnesses who were present at the time - each of their interview transcripts mentions the bakery. Littering took place at 16:36. No known witnesses.

The result seen above provides valuable insight as I now learned the precise time the theft took place and there were three witnesses to the crime. Also, each witness report mentions the bakery. I can use this information to identify the witnesses and review their testimonies.

```
SELECT name FROM interviews WHERE month = 7 AND day = 28 AND transcript LIKE '%bakery%';
```

```
+-----+
| name |
+-----+
| Ruth |
| Eugene |
| Raymond |
+-----+
```

I identified the three witnesses as Ruth, Eugene and Raymond. I can now retrieve their testimonies from the INTERVIEWS table and review for additional information about the crime by writing a query using an IN statement to retrieve multiple variables simultaneously, then filter by the date when the crime occurred.

```
SELECT name, transcript FROM interviews WHERE name IN ('Ruth', 'Eugene', 'Raymond') AND month = 7 AND day = 28;
```

| name | transcript

| Eugene | "I suppose," said Holmes, "that when Mr. Windibank came back from France he was very annoyed at your having gone to the ball."

| Ruth | Sometime within ten minutes of the theft, I saw the thief get into a car in the bakery parking lot and drive away. If you have security footage from the bakery parking lot, you might want to look for cars that left the parking lot in that time frame.

| Eugene | I don't know the thief's name, but it was someone I recognized. Earlier this morning, before I arrived at Emma's bakery, I was walking by the ATM on Leggett Street and saw the thief there withdrawing some money.

| Raymond | As the thief was leaving the bakery, they called someone who talked to them for less than a minute. In the call, I heard the thief say that they were planning to take the earliest flight out of Fiftyville tomorrow. The thief then asked the person on the other end of the phone to purchase the flight ticket.

By examining the testimonies of each of the witnesses, I can see a lot of useful information about where to take my investigation next. There are several queries I will have to execute based on each testimony, so I will go through them one by one starting with Ruth.

Ruth stated that within 10mins of the theft she saw the thief drive away from the bakery parking lot. I can use these parameters to get a list of individuals by joining the BAKERY SECURITY LOGS table with the PEOPLE table on the basis of the license plate field. Then filter by time and date to give me the correct range of data.

4

```
SELECT bakery_security_logs.activity,
bakery_security_logs.license_plate, people.name FROM people
JOIN bakery_security_logs ON bakery_security_logs.license_plate =
people.license_plate
WHERE bakery_security_logs.year = 2021
AND bakery_security_logs.month = 7
AND bakery_security_logs.day = 28
AND bakery_security_logs.hour = 10
AND bakery_security_logs.minute >= 15
AND bakery_security_logs.minute <= 25;
```

activity	license_plate	name
exit	5P2BI95	Vanessa
exit	94KL13X	Bruce
exit	6P58WS2	Barry
exit	4328GD8	Luca
exit	G412CB7	Sofia
exit	L93JTIZ	Iman
exit	322W7JE	Diana
exit	ONTHK55	Kelsey

Now that I have a list of names for those people who exited the parking lot within the timeframe specified by Ruth, I can turn to the next testimony which is Eugene's.

Eugene noted that before he arrived at the bakery, he saw the thief withdrawing money at an ATM on Leggett Street. Since I know the theft took place at 10:15am, I can use these parameters to filter for a list of names of people who withdrew money at that ATM location before the time of the theft.

To do this, I will join the BANK ACCOUNTS table with the ATM TRANSACTIONS table on the basis of the account number field, and then join with the PEOPLE table with the BANK ACCOUNTS table on the basis of the id field. Finally, by applying the relevant date and time filters I can get a list of names.

```
SELECT people.name, atm_transactions.transaction_type FROM people
JOIN bank_accounts ON bank_accounts.person_id = people.id
JOIN atm_transactions ON atm_transactions.account_number =
bank_accounts.account_number
WHERE atm_transactions.year = 2021
AND atm_transactions.month = 7
AND atm_transactions.day = 28
AND atm_transactions.atm_location = "Leggett Street"
AND atm_transactions.transaction_type = "withdraw";
```

```
+-----+-----+
| name   | transaction_type |
+-----+-----+
| Bruce  | withdraw         |
| Diana  | withdraw         |
| Brooke | withdraw         |
| Kenny  | withdraw         |
| Iman   | withdraw         |
| Luca   | withdraw         |
| Taylor | withdraw         |
| Benista| withdraw         |
+-----+-----+
```

Now that I have a list of names based on Eugene's testimony, I can move on to Raymond's. In his testimony, Raymond first noted that he saw the thief call someone for less than a minute after they left the bakery.

This information can help identify the thief's accomplice. I can do this by matching the caller with the receiver from the PHONE CALLS table. However, I need to first connect the phone numbers with their owner. This can be done by making some new columns in the PHONE CALLS table using the SET command to join the relevant name and number for both the caller and receiver.

I can then apply the relevant date, time and duration filters to retrieve a list of names based on Raymond's testimony.

```

UPDATE phone_calls
SET caller_name = people.name
FROM people
WHERE phone_calls.caller = people.phone_number;

UPDATE phone_calls
SET receiver_name = people.name
FROM people
WHERE phone_calls.receiver = people.phone_number;

```

```

SELECT caller, caller_name, receiver, receiver_name FROM phone_calls
WHERE year = 2021
AND month = 7
AND day = 28
AND duration < 60;

```

caller	caller_name	receiver	receiver_name
(130) 555-0289	Sofia	(996) 555-8899	Jack
(499) 555-9472	Kelsey	(892) 555-8872	Larry
(367) 555-5533	Bruce	(375) 555-8161	Robin
(499) 555-9472	Kelsey	(717) 555-1342	Melissa
(286) 555-6063	Taylor	(676) 555-6554	James
(770) 555-1861	Diana	(725) 555-3243	Philip
(031) 555-6622	Carina	(910) 555-3251	Jacqueline
(826) 555-1652	Kenny	(066) 555-9701	Doris
(338) 555-6650	Benista	(704) 555-2131	Anna

In the second part of Raymond's testimony, he stated that he overheard the thief was planning to leave town on the first flight out in the morning and asked the person they were speaking with to make the booking.

I can use this information to identify where the thief went. However, the origin and destination airport IDs in the FLIGHTS table are not currently joined up with their respective names in the AIRPORTS table. To correct this, I will apply the same principle as with the phone calls, and update the FLIGHTS table with two new columns, each connecting the airport name with its corresponding ID.

Finally, I will again apply the relevant date and time filters to get the first flight that left town after the theft occurred and its destination.

```

UPDATE flights
SET origin_airport_id = airports.city
FROM airports
WHERE flights.origin_airport_id = airports.id;

UPDATE flights
SET destination_airport_id = airports.city
FROM airports
WHERE flights.destination_airport_id = airports.id;

```

```

SELECT id, hour, minute, origin_airport_id, destination_airport_id FROM
flights
WHERE year = 2021
AND month = 7
AND day = 29
ORDER BY hour ASC
LIMIT 1;

```

```

+-----+-----+-----+-----+-----+
| id | hour | minute | origin_airport_id | destination_airport_id |
+-----+-----+-----+-----+-----+
| 36 | 8    | 20    | Fiftyville        | New York City         |
+-----+-----+-----+-----+-----+

```

Now that we know the destination of the first flight out of town was New York City, I can identify the passengers of that flight by joining the PASSENGERS and PEOPLE on the basis of the passport number field, and then with the FLIGHTS table on the flight id field.

```

SELECT flights.destination_airport_id, name, phone_number,
license_plate FROM people
JOIN passengers ON people.passport_number = passengers.passport_number
JOIN flights ON flights.id = passengers.flight_id
WHERE flights.id = 36
ORDER BY flights.hour ASC;

```

```

+-----+-----+-----+-----+-----+
| destination_airport_id | name | phone_number | license_plate |
+-----+-----+-----+-----+-----+
| New York City         | Doris | (066) 555-9701 | M51FA04      |
| New York City         | Sofia | (130) 555-0289 | G412CB7      |
| New York City         | Bruce | (367) 555-5533 | 94KL13X      |
| New York City         | Edward | (328) 555-1152 | 130LD9Z      |
| New York City         | Kelsey | (499) 555-9472 | ONTHK55      |
| New York City         | Taylor | (286) 555-6063 | 1106N58      |
| New York City         | Kenny | (826) 555-1652 | 30G67EN      |
| New York City         | Luca  | (389) 555-5198 | 4328GD8      |
+-----+-----+-----+-----+-----+

```

1

After collating a list of names from each testimony, the final step to solving who committed the crime is to create a detailed query which will identify whoever is present within all three of the list of names we collected. The result of this query should be the perpetrator.

Since we have already written each query to identify the list of names from each testimony, we can combine them together in one single query to return the name of the person who is common across all three.

```
SELECT name FROM people
JOIN passengers ON people.passport_number = passengers.passport_number
JOIN flights ON flights.id = passengers.flight_id
WHERE (flights.year = 2021 AND flights.month = 7 AND flights.day = 29
AND flights.id = 36)
AND name IN (SELECT phone_calls.caller_name FROM phone_calls WHERE year
= 2021 AND month = 7 AND day = 28 AND duration < 60)
AND name IN (SELECT people.name FROM people JOIN bank_accounts ON
bank_accounts.person_id = people.id
JOIN atm_transactions ON atm_transactions.account_number =
bank_accounts.account_number
WHERE atm_transactions.year = 2021
AND atm_transactions.month = 7
AND atm_transactions.day = 28
AND atm_transactions.atm_location = "Leggett Street"
AND atm_transactions.transaction_type = "withdraw")
AND name IN (SELECT people.name FROM people JOIN bakery_security_logs
ON bakery_security_logs.license_plate = people.license_plate
WHERE bakery_security_logs.year = 2021
AND bakery_security_logs.month = 7
AND bakery_security_logs.day = 28
AND bakery_security_logs.hour = 10
AND bakery_security_logs.minute >= 15
AND bakery_security_logs.minute <= 25);
```

```
+-----+
| name  |
+-----+
| Bruce |
+-----+
```

Upon executing this final query, we learn that Bruce is the thief. With this knowledge we can also find his accomplice by reviewing the caller and receiver columns in the PHONE CALLS table from earlier.

In summary and to answer the original questions:

- Who the thief is: **Bruce**
- What city the thief escaped to: **New York City**
- Who the thief's accomplice is who helped them escape: **Robin**