



CS50x

Challenge SQL - Fiftyville

Italo Guasti

Ouro Preto
22 de outubro de 2024

Sumário

1	Introdução	1
2	Problema a resolver	1
2.1	Linguagem Utilizada	1
2.2	Resolução do Mistério	1
2.3	Resultados	7
2.4	Conclusão	8

1 Introdução

Este relatório apresenta a solução desenvolvida para o meu desafio preferido do curso **Introduction to Computer Science** da **Harvard Online**. O desafio é sobre a semana de estudos referentes a **SQL** onde consistiu em através de um banco de dados investigar e solucionar o mistério de Fiftyville onde o pato CS50 foi roubado!

2 Problema a resolver

O Pato **CS50** foi roubado! A cidade de Fiftyville chamou você para resolver o mistério do pato roubado. As autoridades acreditam que o ladrão roubou o pato e, logo depois, pegou um voo para fora da cidade com a ajuda de um cúmplice. Seu objetivo é identificar:

- Quem é o ladrão,
- Para qual cidade o ladrão fugiu e
- Quem é o cúmplice do ladrão que os ajudou a escapar
- Tudo o que você sabe é que o roubo ocorreu em 28 de julho de 2023 e aconteceu na Humphrey Street .

Como resolver esse mistério? As autoridades de Fiftyville pegaram alguns registros da cidade da época do roubo e prepararam um banco de dados SQLite para você, fiftyville.db, que contém tabelas de dados de toda a cidade. Você pode consultar essa tabela usando SELECT queries SQL para acessar os dados de seu interesse. Usando apenas as informações do banco de dados, a tarefa é resolver o mistério.

2.1 Linguagem Utilizada

A linguagem de programação utilizada neste desafio por mim foi o **SQL** utilizando o **SQLite3**. SQL foi selecionada pelo curso e também por ser uma linguagem amplamente utilizada na manipulação e consulta de bancos de dados, especialmente em tarefas que envolvem a recuperação e análise de grandes volumes de dados. Sua sintaxe declarativa e estruturada permite a execução eficiente de consultas complexas, além de oferecer funções poderosas para filtragem, agregação e junção de dados. Além disso, **SQL** e **SQLite3** são amplamente adotados na indústria de tecnologia para o gerenciamento de dados, o que os torna uma escolha ideal para este tipo de desafio técnico.

2.2 Resolução do Mistério

Utilizando o comando `.tables` obtemos a lista das tabelas disponíveis para solucionarmos o nosso problema.

```
sql/ $ cd fiftyville/
sql/fiftyville/ $ sqlite3 fiftyville.db
sqlite> .tables
airports          crime_scene_reports  people
atm_transactions  flights              phone_calls
bakery_security_logs  interviews
bank_accounts      passengers
sqlite> █
```

Primeiramente, vamos averiguar as **informações iniciais do crime**, através dos relatórios de cena do crime em `crime_scene_reports`.

query

```
1 SELECT description
2 FROM crime_scene_reports
3 WHERE month = 7 AND day = 28
4 AND street = 'Humphrey Street';
```

Obtemos:

response

description: O roubo do pato do CS50 ocorreu às 10h15 na padaria da Rua Humphrey. Entrevistas foram conduzidas hoje com três testemunhas que estavam presentes no momento – cada uma das transcrições das entrevistas menciona a padaria.

description: Descarte de lixo ocorreu às 16h36. Nenhuma testemunha conhecida.

Agora analisaremos as **entrevistas ocorridas no dia** (apenas listei abaixo as entrevistas relevantes, pois eram muitas).

query

```
1 SELECT *
2 FROM interviews
3 WHERE month = 7 AND day = 28;
```

Obtemos:

response

Ruth: O ladrão entrou em um carro no estacionamento da padaria. Se você tiver imagens de segurança do estacionamento da padaria, talvez queira verificar os carros que saíram nesse período.

Eugene: O ladrão era alguém reconhecido. Vi ele sacando dinheiro pelo caixa eletrônico na rua Leggett 10 minutos após o roubo.

Raymond: Quando o ladrão estava saindo da padaria, ele ligou para alguém e falou com essa pessoa por menos de um minuto. Horário do assalto 10:15am. Na chamada, ouvi o ladrão dizer que planejava pegar o primeiro voo saindo de Fiftyville amanhã.

Sendo assim, vamos investigar cada nova pista obtida nas entrevistas.

Em primeiro, **Ruth**, placas de carro que saíram da padaria em até 10 minutos após o crime:

query

```
1 SELECT *
2 FROM bakery_security_logs
3 WHERE month = 7
4     AND activity = 'exit'
5     AND day = 28
6     AND hour = 10
7     AND minute BETWEEN 15 AND 25;
```

response:

	id	year	month	day	hour	minute	activity	license_plate
1	260	2023	7	28	10	16	exit	5P2BI95
2	261	2023	7	28	10	18	exit	94KL13X
3	262	2023	7	28	10	18	exit	6P58WS2
4	263	2023	7	28	10	19	exit	4328GD8
5	264	2023	7	28	10	20	exit	G412CB7
6	265	2023	7	28	10	21	exit	L93JTIZ
7	266	2023	7	28	10	23	exit	322W7JE
8	267	2023	7	28	10	23	exit	0NTHK55

Em segundo, **Eugene**, transações (saques) feitos após o roubo em Leggett Street:

query

```
1 SELECT *
2 FROM atm_transactions
3 WHERE year = 2023
4       AND month = 7
5       AND day = 28
6       AND transaction_type = 'withdraw'
7       AND atm_location = 'Leggett Street';
```

response:

	id	account_number	year	month	day	atm_location	transaction_type	amount
1	246	28500762	2023	7	28	Leggett Street	withdraw	48
2	264	28296815	2023	7	28	Leggett Street	withdraw	20
3	266	76054385	2023	7	28	Leggett Street	withdraw	60
4	267	49610011	2023	7	28	Leggett Street	withdraw	50
5	269	16153065	2023	7	28	Leggett Street	withdraw	80
6	288	25506511	2023	7	28	Leggett Street	withdraw	20
7	313	81061156	2023	7	28	Leggett Street	withdraw	30
8	336	26013199	2023	7	28	Leggett Street	withdraw	35

Vamos conferir também as contas no banco de acordo com os numeros das contas listados acima.

query

```
1 SELECT bank_accounts.*, people.*
2 FROM bank_accounts
3 JOIN people ON bank_accounts.person_id = people.id
4 WHERE bank_accounts.account_number IN (
5     28500762,
6     28296815,
7     76054385,
8     49610011,
9     16153065,
10    25506511,
11    81061156,
12    26013199
13 );
```

response:

1	+	+	+	+	+	+	+	+	+								
2		account_number		person_id		creation_year		id		name		phone_number		passport_number		license_plate	
3	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
4		49610011		686048		2010		686048		Bruce		(367) 555-5533		5773159633		94KL13X	
5		26013199		514354		2012		514354		Diana		(770) 555-1861		3592750733		322W7JE	
6		16153065		458378		2012		458378		Brooke		(122) 555-4581		4408372428		QX4YZN3	
7		28296815		395717		2014		395717		Kenny		(826) 555-1652		9878712108		30G67EN	
8		25506511		396669		2014		396669		Iman		(829) 555-5269		7049073643		L93JTIIZ	
9		28500762		467400		2014		467400		Luca		(389) 555-5198		8496433585		4328GD8	
10		76054385		449774		2015		449774		Taylor		(286) 555-6063		1988161715		1106N58	
11		81061156		438727		2018		438727		Benista		(338) 555-6650		9586786673		8X428L0	
12	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

Observe que já podemos visualizar alguns nomes suspeitos. Vamos em frente.

Por último, Raymond, vamos checar as chamadas efetuadas com menos de um minuto de duração:

query

```
1 SELECT *
2 FROM phone_calls
3 WHERE year = 2023
4     AND month = 7
5     AND day = 28
6     AND duration < 60;
```

response:

1	+	+			+			+	+	+	+	+	+	+			
2		id		caller			receiver			year		month		day		duration	
3	+	+			+			+	+	+	+	+	+	+			
4		221		(130)	555-0289		(996)	555-8899		2023		7		28		51	
5		224		(499)	555-9472		(892)	555-8872		2023		7		28		36	
6		233		(367)	555-5533		(375)	555-8161		2023		7		28		45	
7		251		(499)	555-9472		(717)	555-1342		2023		7		28		50	
8		254		(286)	555-6063		(676)	555-6554		2023		7		28		43	
9		255		(770)	555-1861		(725)	555-3243		2023		7		28		49	
10		261		(031)	555-6622		(910)	555-3251		2023		7		28		38	
11		279		(826)	555-1652		(066)	555-9701		2023		7		28		55	
12		281		(338)	555-6650		(704)	555-2131		2023		7		28		54	
13	+	+			+			+	+	+	+	+	+	+			

Seguindo nossas pistas, vamos buscar o id de origem dos aeroportos.

query

```
1 SELECT *
2 FROM airports;
```

response:

	id	abbreviation	full_name	city
1	1	ORD	O'Hare International Airport	Chicago
2	2	PEK	Beijing Capital International Airport	Beijing
3	3	LAX	Los Angeles International Airport	Los Angeles
4	4	LGA	LaGuardia Airport	New York City
5	5	DFW	Dallas/Fort Worth International Airport	Dallas
6	6	BOS	Logan International Airport	Boston
7	7	DXB	Dubai International Airport	Dubai
8	8	CSF	Fiftyville Regional Airport	Fiftyville
9	9	HND	Tokyo International Airport	Tokyo
10	10	CDG	Charles de Gaulle Airport	Paris
11	11	SFO	San Francisco International Airport	San Francisco
12	12	DEL	Indira Gandhi International Airport	Delhi

Adiante, iremos procurar qual o primeiro voo saindo do aeroporto de Fiftyville no dia posterior ao roubo, dia 29.

query

```
1 SELECT *
2 FROM flights
3 WHERE year = 2023
4       AND month = 7
5       AND day = 29
6       AND origin_airport_id = 8;
```

response:

	id	origin_airport_id	destination_airport_id	year	month	day	hour	minute
1	18	8	6	2023	7	29	16	0
2	23	8	11	2023	7	29	12	15
3	36	8	4	2023	7	29	8	20
4	43	8	1	2023	7	29	9	30
5	53	8	9	2023	7	29	15	20

Podemos observar que o **primeiro voo foi às 8:20**. Então vamos identificar esses passageiros.

query

```
1 SELECT passengers.passport_number, people.name
2 FROM passengers
3 JOIN people ON passengers.passport_number = people.passport_number
4 WHERE passengers.flight_id IN (SELECT id FROM flights WHERE flight_id =
    36 AND year = 2023 AND month = 7 AND day = 29);
```

response:

1	+	+	+	+
2		passport_number		name
3	+	+	+	+
4		7214083635		Doris
5		1695452385		Sofia
6		5773159633		Bruce
7		1540955065		Edward
8		8294398571		Kelsey
9		1988161715		Taylor
10		8496433585		Luca
11	+	+	+	+

Agora que já temos uma gama de informações vamos investigar alguns detalhes para encontrarmos o vilão e seu cúmplice.

Observando os dados podemos notar que a transação feita pela conta **49610011** nos leva a conta bancária de **Bruce** e nos mostra a placa de seu carro **94KL13X** que é uma das placas registradas na saída da padaria e o seu número de telefone **(367) 555-5533** que efetuou uma chamada para **(375) 555-8161**.

Vamos aprofundar um pouco mais! Verificando o nome das pessoas presentes na chamada acima.

query

```
1 SELECT *
2 FROM people
3 WHERE phone_number IN (
4     '(031) 555-6622',
5     '(910) 555-3251'
6 );
```

response:

1	+	+	+	+	+	+
2		id		name		phone_number
3	+	+	+	+	+	+
4		686048		Bruce		(367) 555-5533
5		864400		Robin		(375) 555-8161
6	+	+	+	+	+	+

Voilà.

2.3 Resultados

Após essa severa investigação podemos encontrar o ladrão e seu cúmplice. De acordo com os nossos dados, no dia 28 de julho de 2023, na Humphrey Street, se encontrava Bruce. **Bruce (367) 555-5533** efetuou o roubo do **Pato CS50** e às 10:15 na saída foi visto ligando para o seu cúmplice **Robin (375) 555-8161** e no estacionamento da padaria entrou em seu carro com a **placa 94KL13X**. O vilão informou ao seu parceiro que pegaria o primeiro voo saindo de Fiftyville na manhã seguinte às 8:20. O voo tinha a **id 38** e com destino a **LaGuardia Airport** de **Nova York City**. Nesse sentido, 10 minutos após roubar, o perigoso marginal Bruce foi até a agência mais próxima na **Leggett Street** onde efetuou um saque no valor de \$50 em sua conta **49610011** e o mesmo nunca mais foi visto. No entanto, agora nós o pegamos!

The THIEF is: **Bruce**

The city the thief ESCAPED TO: **New York City**

The ACCOMPLICE is: **Robin**



2.4 Conclusão

O desafio de resolver o mistério do pato roubado em Fiftyville foi uma experiência enriquecedora e estimulante que não apenas aprimorou minhas habilidades em SQL, mas também aprofundou minha compreensão sobre a manipulação e análise de dados. Através deste relatório, busquei esclarecer e documentar as etapas que segui para desvendar esse intrigante caso, evidenciando a importância da prática no desenvolvimento de competências técnicas.

Agradeço à **Harvard University** e ao professor **David J. Malan** pelo ensino gratuito e de alta qualidade que torna desafios como este acessíveis. Sua dedicação à educação online tem um impacto significativo em muitos estudantes, permitindo-nos explorar e expandir nossos conhecimentos em ciência da computação. Este desafio foi, sem dúvida, o meu preferido, e o relatório é um reflexo do meu apreço pela oportunidade de aprender e aplicar o que foi ensinado.

