das vendas.

employees departments

employee_id	first_name	last_name	department_id
100	Steven	King	2
101	Neena	Kochhar	9
102	Lex	De Haan	9
103	Alexander	Hunold	5
104	Bruce	Ernst	6

script.sql

```sql
SELECT *
FROM
employees
WHERE
department_id IN
(
 SELECT department_id
 FROM departments
 WHERE department_name =
'Marketing'
  AND department_name = 'Sales'
);
```

= " . ; ( ) {

Selecione todos os funcionários que trabalham no departamento de Marketing ou Vendas.

employees departments

department_id	department_name	budget
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400
5	Sales	1500

script.sql

```
SELECT *
FROM
employees
WHERE
department_id IN
(
 SELECT department_id
 FROM departments
 WHERE department_name =
'Marketing'
```

= " . ; ( ) {

department_id	department_name	budget
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400
5	Shipping	1500

script.sql Result

```
SELECT *
FROM
employees
WHERE
department_id IN
(
 SELECT department_id
 FROM departments
 WHERE budget > 1800
);
```

CONTINUAR

Selecione todos os funcionários cujo departamento tenha um orçamento superior a 1800.

employees departments

department_id	department_name	budget
1	Administration	1700
2	Marketing	1800
3	Purchasing	1700
4	Human Resources	2400
5	Shipping	1500

script.sql

```
SELECT *
FROM
employees
WHERE
department_id IN
(
```

NOT > department_id

1800 departments budget

Selecione todos os funcionários cujo departamento tenha um orçamento superior a 1800.

employees departments

employee_id	first_name	last_name	department_id
100	Steven	King	4
101	Neena	Kochhar	9
102	Lex	De Haan	9
103	Alexander	Hunold	6
104	Bruce	Ernst	6

script.sql

```
SELECT *
FROM
employees
WHERE
department_id IN
(
```

NOT > department_id

1800 departments budget

por meio de um `ID` compartilhado. Encontre as informações de macro para uma maçã.

snack macros

id	carbs_g	protein_g	fat_g
001	14	0.2	0.2
455	2	15	12
111	0	25	7

script.sql

```
SELECT carbs_g, protein_g, fat_g
FROM macros
WHERE id =
(
 SELECT
 FROM
)
```

62%

Encontre as informaçoes de macro para uma maçã.

snack macros

id	name	type
001	apple	vegan
133	Cheerios	vegan
455	chocolate	vegetarian
060	cashews	vegan

script.sql

```
SELECT carbs_g, protein_g, fat_g
FROM macros
WHERE id =
(
 SELECT
 FROM
)
```

*	snack	=	'apple'
WHERE	name	id	

Selecione o `name` do jogo mais antigo usando uma subconsulta com `MIN` para encontrar o `release` mais antigo.

mario_games

name	release
Mario Bros.	1983
Mario Clash	1995
Super Mario Galaxy	2007
Super Mario 3D World	2013

script.sql Result

name
Mario Bros.

Você entrou `SELECT MIN (release) FROM mario_games` . A resposta correta é `SELECT MIN(release) FROM mario_games` .

CONTINUAR

Selecione o `name` do jogo mais recente.

mario_games

name	release
Mario Bros.	1983
Mario Clash	1995
Super Mario Galaxy	2007
Super Mario 3D World	2013

script.sql

```
SELECT name
FROM mario_games
WHERE release =
(
 SELECT MAX(release)
 FROM mario_games
);
```

MIN

BOM TRABALHO!

de 2.000.

departments employees

id	department_name	budget
1	Administration	4033
2	Marketing	223
3	Purchasing	7678
4	Human Resources	2342

script.sql Result

```sql
SELECT *
FROM employees
where department_id IN
(
 SELECT id
 FROM departments
 WHERE budget > 2000
);
```

CONTINUAR

BOM TRABALHO!

departments employees

employee_id	name	department_id
100	Steven King	9
101	Neena Kochhar	9
102	Lex De Haan	1
103	Alexander Hunold	2
104	Bruce Ernst	6

script.sql Result

```
SELECT *
FROM employees
where department_id IN
(
 SELECT id
 FROM departments
 WHERE budget > 2000
);
```

CONTINUAR

BOM TRABALHO!

id	department_name	budget
1	Administration	4033
2	Marketing	223
3	Purchasing	7678
4	Human Resources	2342

script.sql

```
SELECT *
FROM employees
where department_id IN
(
 SELECT budget
 FROM departments
 WHERE id > 2000
);
```

Certifique-se de escolher `(` seguido por `id` e depois por `departments`, `budget`, `2000`, `)`.

PULAR

DE NOVO

7 SEGUIDOS!

points prizes

score	prize
5	Pen
6	Badge
7	Trinket
8	Watch
8	Bag

script.sql Result

name	alias	score
Lucy	gaminglucy	5
Anonymous	heretowin	8

As pessoas que não vão ganhar uma Caneta ou um Relógio

As pessoas que ganharam uma Caneta ou Relógio

As pessoas que ganharam algum prêmio

CONTINUAR

O que essa consulta seleciona?

points prizes

name	alias	score
Lucy	gaminglucy	5
Henry	hengames	3
Selene	selovesgames	4
Anonymous	heretowin	8

script.sql

```
SELECT *
FROM points
WHERE score IN
(
 SELECT score
 FROM prizes
 WHERE prize = 'Pen' OR prize =
'Watch'
);
```

As pessoas que não vão ganhar uma Caneta ou um Relógio

As pessoas que ganharam uma Caneta

ENVIAR

Selecionando apenas clientes com um `customer_id` presente na tabela orders, obtemos apenas clientes que fizeram um pedido.

customers orders

id	first_name	last_name	email
1	James	Butt	jbutt@gmail.com
2	Josephine	Darakjy	josephine_darakjy@da
4	Lenna	Paprocki	lpaprocki@hotmail.co
5	Artie	Foller	donette.foller@cox.ne
6	Artie	Morasca	simona@morasca.co

script.sql Result

```
SELECT first_name, last_name
FROM customers
WHERE customers.id IN
(
 SELECT customer_id
 FROM orders
);
```

CONTINUAR

64%

Usaremos `IN` para selecionar os clientes com um ID presente na lista de IDs retornada por nossa subconsulta.

customers orders

id	first_name	last_name	email
1	James	Butt	jbutt@gmail.com
2	Josephine	Darakjy	josephine_darakjy@da
4	Lenna	Paprocki	lpaprocki@hotmail.co
5	Artie	Foller	donette.foller@cox.ne
6	Artie	Morasca	simona@morasca.com

script.sql Result

first_name	last_name
James	Butt

Você entrou `IN` . A resposta correta é `IN` .

CONTINUAR

5	Artie	Foller	donette.foller@cox.ne
6	Artie	Morasca	simona@morasca.com

script.sql

```
SELECT first_name, last_name
FROM customers
WHERE customers.id IN
(
 SELECT customer_id
 FROM orders
);
```

consulta que selecione todos os valores `customer_id` em `orders` .

customers orders

customer_id	orderDate	total
1	1/20/2022	300.43
3	1/01/2022	101.23
7	1/02/2022	25.03
7	2/02/2022	525.03

script.sql Result

customer_id
1
3
7
7

CONTINUAR

64%

Esta consulta mostra os clientes que fizeram pedidos, selecionando apenas os clientes cujo ID aparece na tabela `orders` .

customers orders

id	first_name	last_name	email
1	James	Butt	jbutt@gmail.com
2	Josephine	Darakjy	josephine_darakjy@da
4	Lenna	Paprocki	lpaprocki@hotmail.co
5	Artie	Foller	donette.foller@cox.ne
6	Artie	Morasca	simona@morasca.com

script.sql Result

first_name	last_name
James	Butt

CONTINUAR

Assim como as consultas, as subconsultas também podem retornar várias linhas. Aqui está uma prévia.

customers orders

id	first_name	last_name	email
1	James	Butt	jbutt@gmail.com
2	Josephine	Darakjy	josephine_darakjy@da
4	Lenna	Paprocki	lpaprocki@hotmail.co
5	Artie	Foller	donette.foller@cox.ne
6	Artie	Morasca	simona@morasca.com

script.sql

```
SELECT first_name, last_name
FROM customers
WHERE customers.id IN
(
  SELECT customer_id
  FROM orders
);
```

Usamos subconsultas para selecionar uma linha e uma coluna. Como esta subconsulta que seleciona o `total` de pedido mais alto.

orders

customer_id	orderDate	total
1	1/20/2022	300.43
3	1/01/2022	101.23
7	1/02/2022	25.03
7	2/02/2022	525.03
8	1/01/2022	101.23
9	1/02/2022	25.03

script.sql Result

customer_id

CONTINUAR

BOM TRABALHO!

Selecione o nome de todas as lojas com vendas acima do esperado.

store sales_goals

description	target
minimum	40000
outstanding	100000

script.sql Result

store_name	phone	sales
Rowlett Bikes	(972) 530-5555	176760

CONTINUAR

BOM TRABALHO!

Selecione o nome de todas as lojas com vendas acima do esperado.

store sales_goals

store_name	phone	sales
Santa Cruz Bikes	(831) 476-4321	23434
Baldwin Bikes	(516) 379-8888	54543
Rowlett Bikes	(972) 530-5555	176760

script.sql Result

store_name	phone	sales
Rowlett Bikes	(972) 530-5555	176760

CONTINUAR

Selecione o nome de todas as lojas com vendas acima do mínimo.

store sales_goals

description	target
minimum	40000
outstanding	100000

script.sql Result

```
SELECT *
FROM store
WHERE sales >
(
 SELECT target
 FROM sales_goals
 WHERE description = 'minimum'
```

CONTINUAR

vendas acima do mínimo.

store sales_goals

description	target
minimum	40000
outstanding	100000

script.sql Result

```
SELECT *
FROM store
WHERE sales >
(
 SELECT target
 FROM sales_goals
 WHERE description = 'minimum'
);
```

CONTINUAR

store sales_goals

store_name	phone	sales
Santa Cruz Bikes	(831) 476-4321	23434
Baldwin Bikes	(516) 379-8888	54543
Rowlett Bikes	(972) 530-5555	87676

script.sql Result

```
SELECT *
FROM store
WHERE sales >
(
 SELECT target
 FROM sales_goals
 WHERE description = 'minimum'
);
```

CONTINUAR

Selecione o nome de todas as lojas com vendas acima do mínimo.

store sales_goals

description	target
minimum	40000
outstanding	100000

script.sql Result

store_name	phone	sales
Baldwin Bikes	(516) 379-8888	54543
Rowlett Bikes	(972) 530-5555	87676

CONTINUAR

Coloque a subconsulta entre parênteses.

age_statistics customer

age	description
91	max user age
33	average user age
18	min user age

script.sql

```
SELECT *
FROM customer
WHERE age >
  (
   SELECT age
   FROM age_statistics
   WHERE description = 'average
user age'
  );
```

Agora podemos encontrar itens `customer` com uma `age` maior que a média usando essa consulta como uma subconsulta. Aqui está uma prévia.

age_statistics customer

first_name	last_name	age
James	Butt	76
Josephine	Darakjy	41
Lenna	Paprocki	46
Donette	Foller	18
Simona	Morasca	33

script.sql

```
SELECT *
FROM customer
WHERE age >
 (
  SELECT age
  FROM age_statistics
  WHERE description = 'average
user age'
 );
```

= " . ; ( ) {

Agora podemos encontrar itens `customer` com uma `age` maior que a média usando essa consulta como uma subconsulta. Aqui está uma prévia.

age_statistics customer

age	description
91	max user age
33	average user age
18	min user age

script.sql

```
SELECT *
FROM customer
WHERE age >
  (
   SELECT age
   FROM age_statistics
   WHERE description = 'average
user age'
  );
```

= " . ; ( ) {

Lenna	Paprocki	lpaprocki@hotmail.com
Donette	Foller	donette.foller@cox.net
Simona	Morasca	simona@morasca.com

script.sql

```
SELECT first_name, last_name,
email
FROM customers
WHERE age >
(
 SELECT AVG(age)
 FROM customers
);
```

As consultas que fazem parte de outra consulta são chamadas **de subconsultas** , como aquela que calcula a média.

customers

first_name	last_name	email
James	Butt	jbutt@gmail.com
Josephine	Darakjy	josephine_darakjy@darakjy
Lenna	Paprocki	lpaprocki@hotmail.com
Donette	Foller	donette.foller@cox.net
Simona	Morasca	simona@morasca.com

script.sql

```
SELECT first_name, last_name,
email
FROM customers
WHERE age >
(
 SELECT AVG(age)
 FROM customers
);
```

Defina `hotels` como a tabela da esquerda e `booking_requests` como a tabela da direita. Observe a mudança no resultado após trocar as posições da tabela.

booking_requests **hotels**

name	location
Hotel Myx	Paris
Backpackers Hotel	Chicago
Hotel Paradise	Istanbul

script.sql

```
SELECT * FROM hotels
RIGHT JOIN booking_requests
ON hotels.location = booking_reque
sts.client_destination;
```

Defina `hotels` como a tabela da esquerda e `booking_requests` como a tabela da direita. Observe a mudança no resultado após trocar as posições da tabela.

booking_requests hotels

client_name	client_destination
Dan	Paris
Samira	Chicago
Misha	Paris

script.sql

```
SELECT * FROM hotels
RIGHT JOIN booking_requests
ON hotels.location = booking_reque
sts.client_destination;
```

6 SEGUIDOS!

Defina `booking_requests` como a tabela da esquerda e `hotels` como a tabela da direita.

booking_requests hotels

client_name	client_destination
Dan	Paris
Samira	Chicago
Misha	Paris

script.sql Result

client_name	client_destination	name
Dan	Paris	Hotel Myx
Samira	Chicago	Backpackers Hotel
Misha	Paris	Hotel Myx
		Hotel Paradise

CONTINUAR

Defina `booking_requests` como a tabela da esquerda e `hotels` como a tabela da direita.

booking_requests hotels

client_name	client_destination
Dan	Paris
Samira	Chicago
Misha	Paris

script.sql Result

```
SELECT * FROM booking_requests
RIGHT JOIN hotels
ON hotels.location =
booking_requests.client_destinatio
n;
```

CONTINUAR

Todas as linhas, correspondentes ou não, são selecionadas na tabela à direita. Aqui, a mesa certa são `students` .

students courses

name	student_id	course_id
Pat	234	112
Reagan	98	102
Ash	712	212

script.sql Result

d	duration_in_years	name	student_id	course_id
	3	Reagan	98	102
	4	Pat	234	112
		Ash	712	212

CONTINUAR

Para a `class` correspondente ao `name` de cada aluno, juntamos as tabelas com base em uma propriedade comum como `id` .

student enrolled

id	name
0147	Lisa Jones
1008	Luz Garcia
0267	Lin Wong

script.sql Result

```sql
SELECT *
FROM student
JOIN enrolled
ON student.id = enrolled.id;
```

CONTINUAR

14 SEGUIDOS!

Use um alias para renomear `COUNT(*)` para `items`.

wishlist

item	price	category
headphones	250	electronics
pencils	37	stationery
phone charger	110	electronics
daily planner	29	stationery

script.sql Result

```
SELECT category, COUNT(*) AS
items
FROM wishlist
GROUP BY category;
```

CONTINUAR

14 SEGUIDOS!

Use um alias para renomear `COUNT(*)` para `items`.

wishlist

item	price	category
headphones	250	electronics
pencils	37	stationery
phone charger	110	electronics
daily planner	29	stationery

script.sql Result

category	items
electronics	2
stationery	2

CONTINUAR

Quantos grupos teremos nos resultados?

wishlist

item	price	category
headphones	250	electronics
pencils	37	stationery
phone charger	110	electronics
daily planner	29	stationery

script.sql Result

category
electronics
stationery

Nenhum

Quatro, `electronics` duas vezes e `stationery` duas vezes

Dois, `electronics` e `stationery`

CONTINUAR

13 SEGUIDOS!

Quantos grupos teremos nos resultados?

wishlist

item	price	category
headphones	250	electronics
pencils	37	stationery
phone charger	110	electronics
daily planner	29	stationery

script.sql Result

```
SELECT category
FROM wishlist
GROUP BY category;
```

Nenhum

Quatro, `electronics` duas vezes e `stationery` duas vezes

Dois, `electronics` e `stationery`

CONTINUAR

O que essa consulta faz?

wishlist

item	price	category
headphones	250	electronics
pencils	37	stationery
phone charger	110	electronics
daily planner	29	stationery

script.sql Result

category	SUM(price)
electronics	360
stationery	66

Ele soma todos os valores `price` em um resultado

Agrupa itens `wishlist` por `price`

Ele agrupa itens `wishlist` por `category` e soma os preços em cada categoria

CONTINUAR

O que essa consulta faz?

wishlist

item	price	category
headphones	250	electronics
pencils	37	stationery
phone charger	110	electronics
daily planner	29	stationery

script.sql Result

```
SELECT category, SUM(price)
FROM wishlist
GROUP BY category;
```

Ele soma todos os valores `price` em um resultado

Agrupa itens `wishlist` por `price`

Ele agrupa itens `wishlist` por `category` e soma os preços em cada categoria

CONTINUAR

O que essa consulta faz?

user

email	country
stuart.m@mail.com	England
lo_daniel@mail.com	Spain
ann23@mail.com	England

script.sql Result

```
SELECT country, COUNT(*)
FROM user
GROUP BY country;
```

Conta todos os itens `user` na tabela

Ele divide a contagem de usuários entre os grupos `country`

CONTINUAR

Para listar as tabelas, usaremos `sqlite_schema` , pois estamos usando SQLite. `sqlite_schema` é uma tabela que armazena o esquema de um banco de dados.

Um esquema contém informações sobre um banco de dados, incluindo detalhes sobre as tabelas.

ⓘ Toque nos trechos de código abaixo

past_events events crew

name	entry	attendees
Pride Party	10	79
Classical Day	15	76
Wine tasting	8	43

script.sql

```
SELECT * FROM sqlite_schema;
```

Quais linhas essa instrução UPDATE mudará?

Songs

ame	artist	streams	certifications
ise	The Jets	120	
ay	Stevie J	100	
e Sunrise	Harry	90	
rnia Highway	Joan & Jerry	85	
ster Meg	Couches	70	

script.sql

```sql
UPDATE Songs
SET certifications = 'Platinum'
WHERE streams >= 100;
```

- As duas primeiras filas
- Apenas a primeira linha

ENVIAR

Por exemplo, imagine que queremos preencher a coluna `mealservice` com `True` para todos os voos de 6 horas ou mais.

Flights

	origin	destination	duration	mealservice
l	Paris	London	1.2	
	New York	Los Angeles	5.5	
l	New York	London	7.0	
	Honolulu	Boston	9.5	

script.sql

```
UPDATE Flights
SET mealservice = True
WHERE duration >= 6;
```

```
SELECT * FROM Flights;
```

Por exemplo, imagine que queremos preencher a coluna `mealservice` com `True` para todos os voos de 6 horas ou mais.

Flights

number	type	origin	destination
PA84	International	Paris	London
PA56	Domestic	New York	Los Angeles
PA67	International	New York	London
PA76	Domestic	Honolulu	Boston

script.sql

```
UPDATE Flights
SET mealservice = True
WHERE duration >= 6;
```

```
SELECT * FROM Flights;
```

Não queremos simplesmente `INSERT` uma nova entrada na tabela, caso contrário, teremos reservas duplicadas com o mesmo nome. Isso pode ficar confuso!

Excluir e reinserir uma entrada também não é elegante. Em vez disso, podemos usar a sintaxe `UPDATE` do SQL.

Reservations

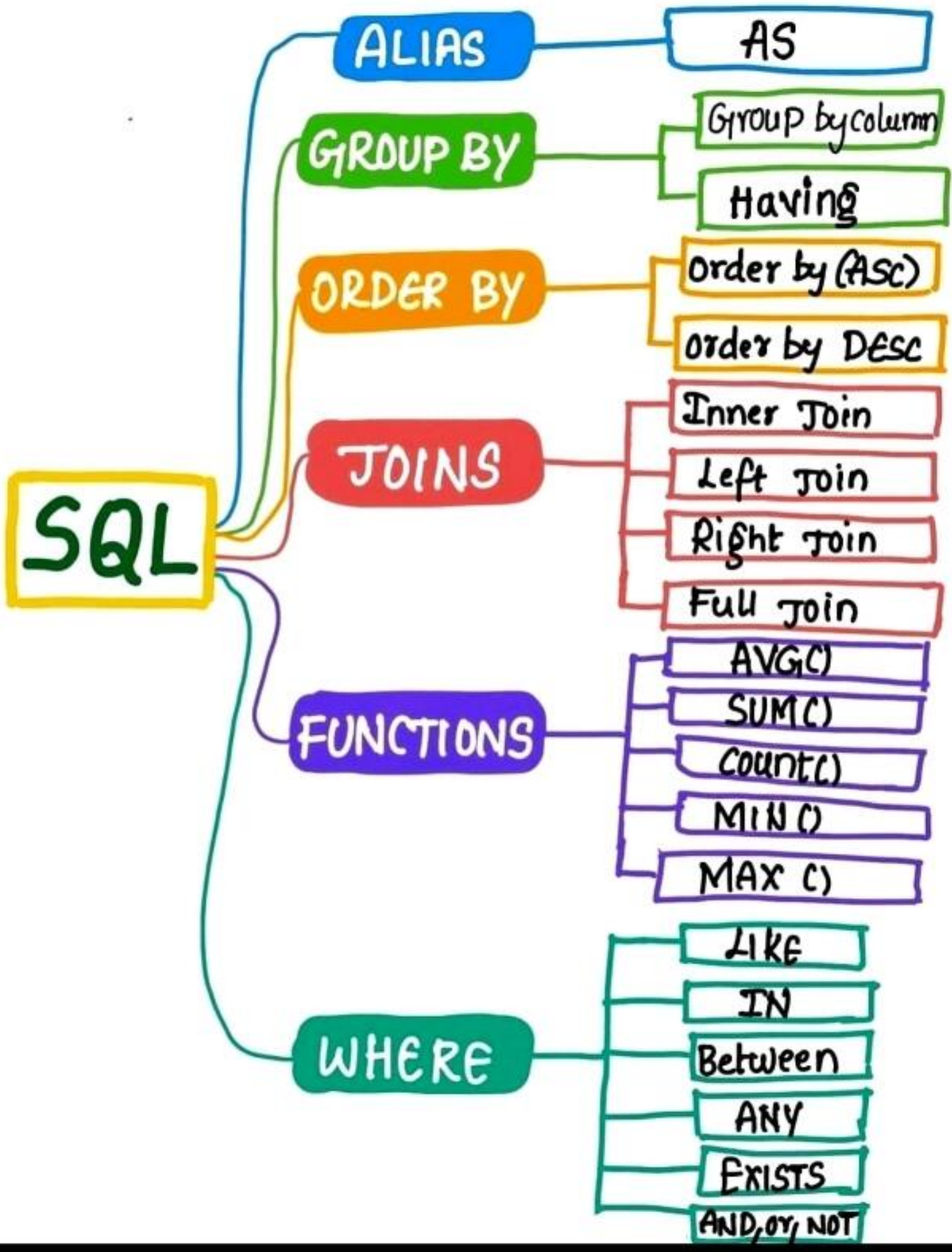
id	name	phonenumber	time	partysize
1	Powers	+16352637463	18:00	4
2	Miranda	+17487652839	19:00	5
3	Smith	+17648373572	18:30	3

script.sql

```
UPDATE Reservations SET time =
'19:00' WHERE name = 'Smith';

SELECT * FROM Reservations;
```

SQL ZERO TO HERO!



Use `DISTINCT` para descobrir quais gêneros musicais esse usuário ouve.

favorites

song	genre
All Blues	jazz
What goes on	art rock
Sure Shot	hip hop
Giant Steps	jazz