

# SQL Queries - SQL Course

Matheus Henrique SantAnna Cardoso  
DRE: 121073530

25 de Junho de 2023

# 1 Curso Iniciante

Queries SQL realizadas para o curso SQL Course online. Estas, para a parte inicial do mesmo curso.

## 1.1 Selecting Data

1.) - Exiba o primeiro nome e a idade de todos os que estão na tabela.

```
1 SELECT first, age FROM empinfo;
```

first	age
John	45
Mary	25
Eric	32
Mary Ann	32
Ginger	42
Sebastian	23
Gus	35
Mary Ann	52
Erica	60
Leroy	22
Elroy	22

2.) - Exiba o nome, o sobrenome e a cidade de todos que não são de Payson.

```
1 SELECT first, last, city FROM empinfo WHERE city <> "Payson";
```

first	last	city
Eric	Edwards	San Diego
Mary Ann	Edwards	Phoenix
Ginger	Howell	Cottonwood
Sebastian	Smith	Gila Bend
Gus	Gray	Bagdad
Mary Ann	May	Tucson
Erica	Williams	Show Low
Leroy	Brown	Pinetop
Elroy	Cleaver	Globe

3.) - Exiba todas as colunas para todas as pessoas com mais de 40 anos.

```
1 SELECT * FROM empinfo WHERE age > 40;
```

first	last	id	age	city	state
John	Jones	99980	45	Payson	Arizona
Ginger	Howell	98002	42	Cottonwood	Arizona
Mary Ann	May	32326	52	Tucson	Arizona
Erica	Williams	32327	60	Show Low	Arizona

4.) - Exiba o nome e o sobrenome de todos cujo sobrenome termina em "ay".

```
1 SELECT first, last FROM empinfo WHERE last LIKE "%ay"
```

first	last
Gus	Gray
Mary Ann	May

5.) - Exiba todas as colunas para todos cujo primeiro nome é igual a "Mary".

```
1 SELECT * FROM empinfo WHERE first = "Mary";
```

first	last	id	age	city	state
Mary	Jones	99982	25	Payson	Arizona

6.) - Exiba todas as colunas para todos cujo primeiro nome contenha "Mary".

```
1 SELECT * FROM empinfo WHERE first LIKE "%Mary%";
```

first	last	id	age	city	state
Mary	Jones	99982	25	Payson	Arizona
Mary Ann	Edwards	88233	32	Phoenix	Arizona
Mary Ann	May	32326	52	Tucson	Arizona

## 1.2 Creating Tables

```
1 CREATE TABLE myemployees(  
2     firstname VARCHAR(50),  
3     lastname  VARCHAR(50),  
4     title     VARCHAR(50),  
5     age       NUMBER(128),  
6     salary    NUMBER(2000000)  
7 );
```

## 1.3 Inserting Into a Table

```
1 INSERT INTO  
2 myemployees  
3 (firstname, lastname, title, age, salary)  
4 VALUES  
5 ("Jonie", "Weber", "Secretary", 28, 19500.00);  
6  
7 INSERT INTO  
8 myemployees  
9 (firstname, lastname, title, age, salary)  
10 VALUES  
11 ("Potsy", "Weber", "Programmer", 32, 45300.00);  
12  
13 INSERT INTO  
14 myemployees  
15 (firstname, lastname, title, age, salary)  
16 VALUES  
17 ("Dirk", "Smith", "Programmer II", 45, 75020.00);
```

1.) - Selecione todas as colunas para todos em sua tabela de funcionários.

```
1 SELECT * FROM myemployees;
```

firstname	lastname	title	age	salary
Jonie	Weber	Secretary	28	19500
Potsy	Weber	Programmer	32	45300
Dirk	Smith	Programmer II	45	75020

2.) - Selecione todas as colunas para todos com um salário acima de 30.000.

```
1 SELECT * FROM myemployees WHERE salary > 30000;
```

firstname	lastname	title	age	salary
Potsy	Weber	Programmer	32	45300
Dirk	Smith	Programmer II	45	75020

3.) - Selecione o nome e o sobrenome de todas as pessoas com menos de 30 anos.

```
1 SELECT firstname, lastname FROM myemployees WHERE age < 30;
```

firstname	lastname
Jonie	Weber

4.) - Selecione o nome, o sobrenome e o salário de qualquer pessoa com “Programmer” no cargo.

```
1 SELECT firstname, lastname FROM myemployees WHERE title LIKE "%Programmer%";
```

firstname	lastname
Potsy	Weber
Dirk	Smith

5.) - Selecione todas as colunas para todos cujo sobrenome contenha “ebe”.

```
1 SELECT * FROM myemployees WHERE lastname LIKE "%ebe%";
```

firstname	lastname	title	age	salary
Jonie	Weber	Secretary	28	19500
Potsy	Weber	Programmer	32	45300

6.) - Selecione o primeiro nome para todos cujo primeiro nome é igual a “Potsy”.

```
1 SELECT firstname FROM myemployees WHERE firstname = "Potsy";
```

firstname
Potsy

7.) - Selecione todas as colunas para todos com mais de 80 anos.

```
1 SELECT * FROM myemployees WHERE age > 80;
```

firstname	lastname	title	age	salary
-----------	----------	-------	-----	--------

8.) - Selecione todas as colunas para todos cujo sobrenome termina em “ith”

```
1 SELECT * FROM myemployees WHERE lastname LIKE "%ith";
```

firstname	lastname	title	age	salary
Dirk	Smith	Programmer II	45	75020

## 1.4 Updating Records

1.) - Jonie Weber acabou de se casar com Bob Williams. Ela solicitou que seu sobrenome fosse atualizado para Weber-Williams.

```

1  UPDATE myemployees
2      SET lastname = "Weber-Williams"
3      WHERE firstname = "Jonie";
4
5  SELECT * FROM myemployees;

```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Secretary	28	19500
Potsy	Weber	Programmer	32	45300
Dirk	Smith	Programmer II	45	75020

2.) - O aniversário de Dirk Smith é hoje, adicione 1 a sua idade.

```

1  UPDATE myemployees
2      SET age = age + 1
3      WHERE firstname = "Dirk";
4
5  SELECT * FROM myemployees;

```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Secretary	28	19500
Potsy	Weber	Programmer	32	45300
Dirk	Smith	Programmer II	46	75020

3.) - Todas as secretárias passam a ser chamadas de “Auxiliar Administrativo”. Atualize todos os títulos de acordo.

```

1  UPDATE myemployees
2      SET title = "Administrative Assistant"
3      WHERE title = "Secretary";
4
5  SELECT * FROM myemployees;

```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Administrative Assistant	28	19500
Potsy	Weber	Programmer	32	45300
Dirk	Smith	Programmer II	46	75020

4.) - Todos que estão ganhando menos de 30.000 receberão um aumento de 3.500 por ano.

```

1  UPDATE myemployees
2      SET salary = salary + 3500
3      WHERE salary < 30000;
4
5  SELECT * FROM myemployees;

```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Administrative Assistant	28	23000
Potsy	Weber	Programmer	32	45300
Dirk	Smith	Programmer II	46	75020

5.) - Todos que estão ganhando mais de 33.500 receberão um aumento de 4.500 por ano.

```

1  UPDATE myemployees
2      SET salary = salary + 4500
3      WHERE salary > 33500;
4
5  SELECT * FROM myemployees;

```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Administrative Assistant	28	23000
Potsy	Weber	Programmer	32	49800
Dirk	Smith	Programmer II	46	79520

6.) - Todos os títulos de “Programador II” agora são promovidos a “Programador III”

```

1  UPDATE myemployees
2      SET title = "Programmer III"
3      WHERE title = "Programmer II" ;
4
5  SELECT * FROM myemployees;
```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Administrative Assistant	28	23000
Potsy	Weber	Programmer	32	49800
Dirk	Smith	Programmer III	46	79520

7.) - Todos os títulos de “Programador” agora são promovidos a “Programador II”

```

1  UPDATE myemployees
2      SET title = "Programmer II"
3      WHERE title = "Programmer" ;
4
5  SELECT * FROM myemployees;
```

firstname	lastname	title	age	salary
Jonie	Weber-Williams	Administrative Assistant	28	23000
Potsy	Weber	Programmer II	32	49800
Dirk	Smith	Programmer III	46	79520

## 1.5 Deleting Records

1.) - Jonie Weber-Williams acabou de sair, remova seu registro da tabela.

```

1  DELETE FROM myemployees
2      WHERE firstname = "Jonie" AND lastname = "Weber-Williams";
3
4  SELECT * FROM myemployees;
```

firstname	lastname	title	age	salary
Potsy	Weber	Programmer II	32	49800
Dirk	Smith	Programmer III	46	79520

2.) - É hora de cortes orçamentários. Remova todos os funcionários que estão ganhando mais de 70.000 dólares.

```

1  DELETE FROM myemployees
2      WHERE salary > 70000;
3
4  SELECT * FROM myemployees;
```

firstname	lastname	title	age	salary
Potsy	Weber	Programmer II	32	49800

## 1.6 Drop a Table

```
1 DROP TABLE myemployees;
```

## 2 Advanced Course

### 2.1 Select Statement

1.) - Na tabela items\_ordered, selecione uma lista de todos os itens comprados para o customerid 10449. Exiba o customerid, item e preço para este cliente.

```
1 SELECT customerid, item, price
2   FROM items_ordered
3  WHERE customerid = 10449;
```

customerid	item	price
10449	Unicycle	180.79
10449	Snow Shoes	45
10449	Bicycle	380.5
10449	Canoe	280
10449	Flashlight	4.5
10449	Canoe paddle	40

2.) - Selecione todas as colunas da tabela items\_ordered para quem comprou uma Tent.

```
1 SELECT *
2   FROM items_ordered
3  WHERE item = "Tent";
```

customerid	order_date	item	quantity	price
10439	18-Sep-1999	Tent	1	88
10438	18-Jan-2000	Tent	1	79.99

3.) - Selecione os valores customerid, order\_date e item da tabela items\_ordered para todos os itens na coluna de itens que começam com a letra "S".

```
1 SELECT customerid, order_date, item
2   FROM items_ordered
3  WHERE item LIKE "S%";
```

customerid	order_date	item
10449	01-Sep-1999	Snow Shoes
10410	28-Oct-1999	Sleeping Bag
10101	08-Mar-2000	Sleeping Bag
10330	19-Apr-2000	Shovel



4.) - Selecione os itens distintos na tabela items\_ordered. Em outras palavras, exiba uma listagem de cada um dos itens exclusivos da tabela items\_ordered.

```
1 SELECT DISTINCT item FROM items_ordered;
```

```
item
Pogo stick
Raft
Skateboard
Life Vest
Parachute
Umbrella
Unicycle
Ski Poles
Rain Coat
Snow Shoes
Tent
Lantern
Sleeping Bag
Pillow
Helmet
Bicycle
Canoe
Hoola Hoop
Flashlight
Inflatable Mattress
Lawnchair
Compass
Pocket Knife
Canoe paddle
Ear Muffs
Shovel
```

5.) - Crie suas próprias declarações selecionadas e envie-as.

```
1 SELECT DISTINCT item FROM items_ordered WHERE price < 10;
```

```
item
Umbrella
Pillow
Compass
Flashlight
```

## 2.2 Aggregate Functions

1.) - Selecione o preço máximo de qualquer item pedido na tabela items\_ordered. Dica: selecione apenas o preço máximo.

```
1 SELECT MAX(price) FROM items_ordered;
```

```
MAX(price)
1250
```

2.) - Selecione o preço médio de todos os itens pedidos que foram comprados no mês de dezembro.

```
1 SELECT AVG(price) FROM items_ordered WHERE order_date LIKE "%Dec%";

      AVG(price)
      174.3125
```

3.) - Qual é o número total de linhas na tabela items\_ordered?

```
1 SELECT COUNT(*) FROM items_ordered;

      COUNT(*)
      32
```

4.) - Para todas as barracas que foram encomendadas na tabela items\_ordered, qual é o preço da barraca mais barata? Dica: Sua consulta deve retornar apenas o preço.

```
1 SELECT MIN(price) FROM items_ordered WHERE item = "Tent";

      MIN(price)
      79.99
```

## 2.3 Group By Clause

1.) - Quantas pessoas estão em cada estado único na tabela de clientes? Selecione o estado e exiba o número de pessoas em cada um. Dica: count é usado para contar linhas em uma coluna, sum funciona apenas em dados numéricos.

```
1 SELECT COUNT(customerid), state FROM customers GROUP BY state;

      COUNT(customerid)  state
      6                   Arizona
      2                   Colorado
      1                   Hawaii
      1                   Idaho
      1                   North Carolina
      2                   Oregon
      1                   South Carolina
      2                   Washington
      1                   Wisconsin
```

2.) - Na tabela items\_ordered, selecione o item, preço máximo e preço mínimo para cada item específico na tabela. Dica: Os itens precisarão ser divididos em grupos separados.

```
1 SELECT item, MAX(price), MIN(price) FROM items_ordered GROUP BY item;
```

item	MAX(price)	MIN(price)
Bicycle	380.5	380.5
Canoe	280	280
Canoe paddle	40	40
Compass	8	8
Ear Muffs	12.5	12.5
Flashlight	28	4.5
Helmet	22	22
Hoola Hoop	14.75	14.75
Inflatable Mattress	38	38
Lantern	29	16
Lawnchair	32	32
Life Vest	125	125
Parachute	1250	1250
Pillow	8.5	8.5
Pocket Knife	22.38	22.38
Pogo stick	28	28
Raft	58	58
Rain Coat	18.3	18.3
Shovel	16.75	16.75
Skateboard	33	33
Ski Poles	25.5	25.5
Sleeping Bag	89.22	88.7
Snow Shoes	45	45
Tent	88	79.99
Umbrella	6.75	4.5
Unicycle	192.5	180.79

3.) - Quantos pedidos cada cliente fez? Use a tabela items\_ordered. Selecione o customerid, o número de pedidos que eles fizeram e a soma de seus pedidos.

```
1 SELECT customerid, COUNT(*) FROM items_ordered GROUP BY customerid;
```

customerid	COUNT(*)
10101	6
10298	5
10299	2
10315	1
10330	3
10339	1
10410	2
10413	1
10438	3
10439	2
10449	6

## 2.4 Having Clause

1.) - Quantas pessoas estão em cada estado único na tabela de clientes que tem mais de uma pessoa no estado? Selecione o estado e exiba o número de quantas pessoas estão em cada, se for maior que 1.

```
1 SELECT COUNT(*), state
2 FROM customers
3 GROUP BY state
4 HAVING COUNT(*) > 1;
```

COUNT(*)	state
6	Arizona
2	Colorado
2	Oregon
2	Washington

2.) - Na tabela items\_ordered, selecione o item, preço máximo e preço mínimo para cada item específico na tabela. Exiba os resultados somente se o preço máximo de um dos itens for maior que 190,00.

```
1 SELECT item, MAX(price), MIN(price)
2 FROM items_ordered
3 GROUP BY item
4 HAVING MAX(price) > 190.00;
```

item	MAX(price)	MIN(price)
Bicycle	380.5	380.5
Canoe	280	280
Parachute	1250	1250
Unicycle	192.5	180.79

3.) - Quantos pedidos cada cliente fez? Use a tabela items\_ordered. Selecione o ID do cliente, o número de pedidos que eles fizeram e a soma de seus pedidos, caso tenham comprado mais de 1 item.

```
1 SELECT customerid, COUNT(*), SUM(price)
2 FROM items_ordered
3 GROUP BY customerid
4 HAVING COUNT(*) > 1;
```

customerid	COUNT(*)	SUM(price)
10101	6	320.75
10298	5	118.88
10299	2	1288
10330	3	72.75
10410	2	281.72
10438	3	95.24
10439	2	113.5
10449	6	930.79

### 3 Order By Clause

1.) - Selecione o sobrenome, o nome e a cidade de todos os clientes na tabela de clientes. Exiba os resultados em ordem crescente com base no sobrenome.

```
1 SELECT * FROM customers ORDER BY lastname;
```

customerid	firstname	lastname	city	state
10298	Leroy	Brown	Pinetop	Arizona
10408	Elroy	Cleaver	Globe	Arizona
10330	Shawn	Dalton	Cannon Beach	Oregon
10413	Donald	Davids	Gila Bend	Arizona
10439	Conrad	Giles	Telluride	Colorado
10429	Sarah	Graham	Greensboro	North Carolina
10101	John	Gray	Lynden	Washington
10338	Michael	Howell	Tillamook	Oregon
10410	Mary Ann	Howell	Charleston	South Carolina
10315	Lisa	Jones	Oshkosh	Wisconsin
10299	Elroy	Keller	Snoqualmie	Washington
10329	Kelly	Mendoza	Kailua	Hawaii
10449	Isabela	Moore	Yuma	Arizona
10419	Linda	Sakahara	Nogales	Arizona
10339	Anthony	Sanchez	Winslow	Arizona
10325	Ginger	Schultz	Pocatello	Idaho
10438	Kevin	Smith	Durango	Colorado

2.) - Igual ao exercício nº 1, mas exiba os resultados em ordem decrescente.

```
1 SELECT * FROM customers ORDER BY lastname DESC;
```

customerid	firstname	lastname	city	state
10438	Kevin	Smith	Durango	Colorado
10325	Ginger	Schultz	Pocatello	Idaho
10339	Anthony	Sanchez	Winslow	Arizona
10419	Linda	Sakahara	Nogales	Arizona
10449	Isabela	Moore	Yuma	Arizona
10329	Kelly	Mendoza	Kailua	Hawaii
10299	Elroy	Keller	Snoqualmie	Washington
10315	Lisa	Jones	Oshkosh	Wisconsin
10338	Michael	Howell	Tillamook	Oregon
10410	Mary Ann	Howell	Charleston	South Carolina
10101	John	Gray	Lynden	Washington
10429	Sarah	Graham	Greensboro	North Carolina
10439	Conrad	Giles	Telluride	Colorado
10413	Donald	Davids	Gila Bend	Arizona
10330	Shawn	Dalton	Cannon Beach	Oregon
10408	Elroy	Cleaver	Globe	Arizona
10298	Leroy	Brown	Pinetop	Arizona

3.) - Selecione o item e o preço de todos os itens na tabela items\_ordered cujo preço seja maior que 10,00. Exiba os resultados em ordem crescente com base no preço.

```
1 SELECT item, price
2 FROM items_ordered
3 WHERE price > 10
4 ORDER BY price;
```

<b>item</b>	<b>price</b>
Ear Muffs	12.5
Hoola Hoop	14.75
Lantern	16
Shovel	16.75
Rain Coat	18.3
Helmet	22
Pocket Knife	22.38
Ski Poles	25.5
Pogo stick	28
Flashlight	28
Lantern	29
Lawnchair	32
Skateboard	33
Inflatable Mattress	38
Canoe paddle	40
Snow Shoes	45
Raft	58
Tent	79.99
Tent	88
Sleeping Bag	88.7
Sleeping Bag	89.22
Life Vest	125
Unicycle	180.79
Unicycle	192.5
Canoe	280
Bicycle	380.5
Parachute	1250