

SQL para análise de dados



MÚLTIPLAS TABELAS



GUIA DA AULA 1



Utilize union – Teoria

• **Introdução**

• **Sobre a
função Union**



Acompanhe aqui
os temas que
serão tratados
na videoaula



Introdução

Nessa aula, usaremos a seguinte tabela:

```
CREATE TABLE transacoes (
    id_cliente INT,
    id_transacao INT,
    data_compra DATE,
    valor FLOAT,
    id_loja varchar(25)
);
```



Também temos os seguintes valores inseridos na tabela:

```
INSERT INTO transacoes VALUES (1,768805383,'2021-06-10',50.74,'magalu');  
INSERT INTO transacoes VALUES (2,768805399,'2021-06-13',30.90,'giraffas');  
INSERT INTO transacoes VALUES (3,818770008,'2021-06-05',110.00,'postoshell');  
INSERT INTO transacoes VALUES (1,76856563,'2021-07-10',2000.90,'magalu');  
INSERT INTO transacoes VALUES (1,767573759,'2021-06-20',15.70,'subway');  
INSERT INTO transacoes VALUES (3,818575758,'2021-06-25',2.99,'seveneleven');  
INSERT INTO transacoes VALUES (4,764545534,'2021-07-11',50.74,'extra');  
INSERT INTO transacoes VALUES (5,76766789,'2021-08-02',10.00,'subway');  
INSERT INTO transacoes VALUES (3,8154567758,'2021-08-15',1100.00,'shopee');
```



Como resultado da função SELECT, temos a seguinte tabela:

id_cliente	id_transacao	data_compra	valor	id_loja
1	768805383	2021-06-10	50.74	magalu
2	768805399	2021-06-13	30.90	giraffas
3	818770008	2021-06-05	110.00	postoshell
1	76856563	2021-07-10	2000.90	magalu
1	767573759	2021-06-20	15.70	subway
3	818575758	2021-06-25	2.99	seveneleven
4	764545534	2021-07-10	50.74	extra
5	76766789	2021-08-02	10.00	subway
3	8154567758	2021-08-15	1100.00	shopee



Além disso, também utilizaremos a tabela:

```
id_cliente INT,  
nome varchar(25),  
data_compra DATE,  
valor_compra float,  
loja_cadastro varchar(25)  
);
```

Teremos os seguintes dados na tabela:

```
INSERT INTO cliente VALUES (5,'jose','2020-07-01',500.43,'cea');  
INSERT INTO cliente VALUES (1,'maria','2019-03-02',150.70,'riachuelo');  
INSERT INTO cliente VALUES (2,'valentina','2020-01-09',210.99,'zara');  
INSERT INTO cliente VALUES (4,'joana','2019-05-11',1300.50,'pontofrio');  
INSERT INTO cliente VALUES (6,'fernando','2020-03-02',86.55,'pontofrio');
```



Como resultado, temos:

id_cliente	nome	data_compra	valor_compra	loja_cadastro
5	jose	2020-07-01	500.43	cea
1	maria	2019-03-02	150.70	riachuelo
2	valentina	2020-01-09	210.99	zara
4	joana	2019-05-11	1300.50	pontofrio
6	fernando	2020-03-02	86.55	pontofrio



Sobre a função Union

A função **UNION** combina colunas e mostra os valores das colunas sequencialmente. A função é utilizada junto com a função SELECT e estabelece algumas regras:

1. As seleções devem possuir o mesmo número de colunas
2. As colunas combinadas possuem o mesmo tipo de dado

No SQL Server:

```
SELECT <colunas1>
FROM <tabela1> UNION
SELECT <colunas2>
FROM <tabela2>;
```



O resultado:

<u>Id_cliente</u>
1
2
3
4
5
6

Pelo resultado anterior, é possível reparar que os valores não se repetem. Mesmo que a tabela transações mostre `id_cliente = 1` em três linhas, a seleção com `UNION` mostra o valor apenas uma vez. Para mostrar todos os valores, use `UNION ALL` em vez de `UNION`.

