



Referências de tabelas



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



Introdução

Até agora, vimos consultas (SELECT) em uma **tabela**. Porém, também é possível e necessário consultar diferentes tabelas e juntar os resultados.

Por exemplo, um cenário possível seria querer consultar uma tabela em que estão os **dados** do **cliente**, e outra tabela em que estão os **dados** de **vendas** para esses clientes.





Com certeza, na tabela de **vendas**, haverá um campo com o ID do cliente (**customer_id**).

Se quiséssemos mostrar **todas** as vendas de um cliente específico, precisaríamos usar os dados de **ambas as tabelas** e **vinculá-los** a algum campo que eles **compartilham**. Nesse caso, o **cliente_id**.

CLIENTES

id (PK)

nome

sobrenome

VENDAS

id (PK)

data

local

cliente_id (FK)



Consulta SQL

```
SELECT cliente.id AS id, nome.cliente, data.venda  
FROM clientes, vendas  
WHERE cliente.id = vendas.cliente_id;
```





Consulta SQL

```
SELECT clientes.id AS id, clientes.nome, data.venda  
FROM clientes, vendas  
WHERE cliente.id = vendas.cliente_id;
```

Selecionamos aqui:

- A coluna **id** da tabela clientes e atribuímos o alias **id**.
- A coluna **nome** da tabela **clientes**.
- A coluna **data** da tabela **venda**.





Consulta SQL

```
SELECT clientes.id AS id, clientes.nome, data.venda  
FROM clientes, vendas  
WHERE cliente.id = vendas.cliente_id;
```

Fazemos a seleção nas tabelas de **clientes** e **vendas**.
Até aqui a consulta traria **todos** os **clientes** e **todas** as **vendas**. É por isso que ainda precisamos adicionar um **filtro** que mostre **apenas** as **vendas de cada usuário** em particular.





Consulta SQL

```
SELECT clientes.id AS id, clientes.nome, data.venda  
FROM clientes, vendas  
WHERE cliente.id = vendas.cliente_id
```

No **WHERE** criamos uma condição para trazer aqueles registros onde o id do cliente é o mesmo em ambas as tabelas



DigitalHouse>
Coding School