

O que é SQL e por que ele é essencial na análise de dados?

SQL (Structured Query Language) é a linguagem padrão para comunicação com bancos de dados relacionais. Com ele, você pode buscar, filtrar, agrupar, atualizar e manipular dados de forma estruturada e eficiente. Em um mundo movido por dados, o SQL se tornou uma das ferramentas mais poderosas e indispensáveis para analistas. A importância do SQL está na sua capacidade de transformar grandes volumes de dados brutos em informações úteis para tomada de decisão. É uma linguagem acessível, versátil e presente em praticamente todas as empresas que usam dados para gerar



COMANDOS BÁSICOS SQL



1. SELECIONE O QUE IMPORTA — SELECT

O SELECT é a base de toda consulta. Ele busca informações de uma tabela.

Exemplo real: Você quer ver todos os dados da tabela de clientes

```
sql
SELECT * FROM clientes;
```

Quer apenas os nomes e e-mails?

```
SELECT nome, email FROM clientes;
```

🧱 2. FILTRANDO INFORMAÇÕES — WHERE

Use WHERE para filtrar dados com base em condições. Exemplo real:Buscar clientes do estado de SP.

```
sql
SELECT nome, estado
FROM clientes
WHERE estado = 'SP';
```

3. CONTANDO, SOMANDO E MÉDIAS — COUNT, SUM, AVG

Estes são comandos para análise quantitativa. Exemplo real:Contar quantos pedidos foram feitos.

```
sql
SELECT COUNT(*) AS total_pedidos FROM pedidos;
```

Somar o valor total de vendas:

```
sql

SELECT SUM(valor_total) AS faturamento

FROM pedidos;
```

Calcular o valor médio de pedido:

```
sql

SELECT AVG(valor_total) AS ticket_medio

FROM pedidos;
```

120

CRUZANDO TABELAS

JOIN conecta tabelas diferentes através de uma coluna em comum. Exemplo real:Juntar pedidos com dados dos clientes:

```
⑦ Copiar ♡ Editar
SELECT p.id_pedido, c.nome, p.valor_total
FROM pedidos p
JOIN clientes c ON p.id_cliente = c.id_cliente;
```

6 2. ORDENANDO OS RESULTADOS — ORDER BY

Use ORDER BY para organizar os dados por uma coluna. Exemplo real:Listar os 10 maiores pedidos:

```
SELECT * FROM pedidos
ORDER BY valor_total DESC
LIMIT 10;
```

3. QUANDO PRECISAR ESCOLHER CONDIÇÕES — CASE

O CASE unciona como um "se/senão" dentro do SQL. Exemplo real: Classificar pedidos por valor:

```
SELECT id_pedido, valor_total,
  WHEN valor total >= 1000 THEN 'Alto'
   WHEN valor_total >= 500 THEN 'Médio'
   ELSE 'Baixo'
 END AS categoria
FROM pedidos;
```

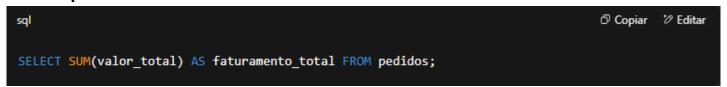


ANÁLISES



1. CRIANDO NOVOS CAMPOS — AS (Alias)

AS serve para renomear colunas ou expressões. Exemplo real:Renomear o valor da soma:





🔀 2. ANALISANDO PERÍODOS — DATE e BETWEEN

Datas são fundamentais em análises. Exemplo real:Buscar pedidos de março de 2025:

```
sql
                                                                       SELECT * FROM pedidos
WHERE data_pedido BETWEEN '2025-03-01' AND '2025-03-31';
```