



# SQL RELOADED

## O Universo dos Dados

Domine os principais comandos SQL e descubra o poder dos dados, navegando pela "Matrix" da informação.

Raíla Corvalho

# INTRODUÇÃO AO SQL



## O que é SQL?

**SQL (Structured Query Language)** é a linguagem padrão para gerenciamento e manipulação de bancos de dados relacionais. Este eBook apresenta os comandos SQL mais importantes, com exemplos práticos que ajudarão você a entender como utilizá-los em contextos reais.

## Por que aprender SQL?

Conhecer SQL permite trabalhar com uma das habilidades mais exigidas em tecnologia, sendo essencial para desenvolvedores, analistas de dados, administradores de banco de dados, entre outros.





# 01

## CRIANDO E GERENCIANDO TABELAS

---

Aprenda a criar e excluir tabelas com os comandos **CREATE TABLE** e **DROP TABLE**.



# CRIAÇÃO DE TABELAS



Para começar, é essencial saber como criar uma tabela. Utilizamos o comando “**CREATE TABLE**”.

**Exemplo:** Criando uma Tabela de Clientes

```
criar_tabela

CREATE TABLE Clientes (
  ID INT PRIMARY KEY,
  Nome VARCHAR(100),
  Email VARCHAR(100),
  DataDeNascimento DATE
);
```

Neste exemplo, criamos uma tabela chamada "Clientes" com quatro colunas: ID, Nome, Email e DataDeNascimento.



# EXCLUSÃO DE TABELAS



Quando não precisamos mais de uma tabela, podemos excluí-la usando o comando “**DROP TABLE**”.

**Exemplo:** Excluindo uma Tabela

```
exclusao_de_tabelas

DROP TABLE Clientes;
```

Este comando exclui a tabela "Clientes" do banco de dados.





# 02

## MANIPULAÇÃO DE DADOS

---

Descubra como inserir, atualizar e excluir dados usando **INSERT INTO**, **UPDATE** e **DELETE**.



# INSERÇÃO DE DADOS



Para inserir dados em uma tabela, usamos o comando “**INSERT INTO**”.

**Exemplo:** Inserindo Dados na Tabela de Clientes

```
insertao_de_dados

INSERT INTO Clientes (ID, Nome, Email, DataDeNascimento)
VALUES (1, 'João Silva', 'joao.silva@email.com', '1985-03-15');
```

Este comando insere um novo cliente na tabela "Clientes".



# ATUALIZAÇÃO DE DADOS



Para atualizar dados existentes, usamos o comando “**UPDATE**”.

**Exemplo:** Atualizando o Email de um Cliente

```
atualizacao_de_dados

UPDATE Clientes
SET Email = 'joao.novoemail@email.com'
WHERE ID = 1;
```

Este comando atualiza o email do cliente com ID 1.





# EXCLUSÃO DE DADOS



Para excluir dados, utilizamos o comando “DELETE”.

**Exemplo:** Excluindo um Cliente

```
exclusao_de_dados  
  
DELETE FROM Clientes WHERE ID = 1;
```

Este comando exclui o cliente com ID 1 da tabela "Clientes".



# 03

## CONSULTANDO DADOS

---

Entenda como consultar, filtrar e ordenar dados  
com o comando **SELECT**.



# SELEÇÃO DE DADOS



Para consultar dados, utilizamos o comando “SELECT”.

**Exemplo:** Seleccionando Todos os Clientes

```
selecao_de_dados

SELECT * FROM Clientes;
```

Este comando seleciona todos os registros da tabela "Clientes".

**Exemplo:** Seleccionando Clientes pelo Nome

```
selecao_de_dados

SELECT Nome, Email FROM Clientes WHERE Nome = 'João Silva';
```

Aqui, selecionamos apenas o nome e o email do cliente chamado João Silva.





# FILTRANDO DADOS



Podemos filtrar dados utilizando uma cláusula “WHERE”.

**Exemplo:** Filtrando Clientes por Data de Nascimento

```
filtrando_dados  
  
SELECT * FROM Clientes WHERE DataDeNascimento > '1990-01-01';
```

Este comando seleciona todos os clientes nascidos após 1º de janeiro de 1990.



# ORDENANDO DADOS



Para ordenar os resultados de uma consulta, usamos “**ORDER BY**”.

**Exemplo:** Ordenando Clientes por Nome

```
ordenando_dados  
  
SELECT * FROM Clientes ORDER BY Nome ASC;
```

Este ordena comando os clientes em ordem alfabética crescente pelo nome.



# 04

## AGRUPANDO E COMBINANDO DADOS

---

Aprenda a agrupar dados com **GROUP BY** e  
combinar tabelas usando **JOIN**.





# AGRUPANDO DADOS



Para agrupar dados, utilizamos “**GROUP BY**”.

**Exemplo:** Contando Clientes por Ano de Nascimento

```
agrupando_dados

SELECT YEAR(DataDeNascimento) AS Ano, COUNT(*) AS NumeroDeClientes
FROM Clientes
GROUP BY YEAR(DataDeNascimento);
```

Este comando agrupa os clientes pelo ano de nascimento e conta quantos clientes nasceram em cada ano.



# JUNÇÃO DE TABELAS



Para combinar dados de várias tabelas, use “JOIN”.

**Exemplo:** Junção de Clientes e Pedidos

Suponha que tenhamos uma tabela "Pedidos":

```
juncao_de_tabelas

CREATE TABLE Pedidos (
  PedidoID INT PRIMARY KEY,
  ClienteID INT,
  DataDoPedido DATE,
  ValorTotal DECIMAL(10, 2),
  FOREIGN KEY (ClienteID) REFERENCES Clientes(ID)
);
```



# JUNÇÃO DE TABELAS



Para selecionar informações dos clientes junto com seus pedidos, fazemos:

```
juncao_de_tabelas

SELECT Clientes.Nome, Pedidos.DataDoPedido, Pedidos.ValorTotal
FROM Clientes
JOIN Pedidos ON Clientes.ID = Pedidos.ClienteID;
```

Este comando junta a tabela "Clientes" com a tabela "Pedidos" e seleciona os nomes dos clientes, como dados dos pedidos e os valores totais.





# 05

## FUNÇÕES AGREGADAS

---

Aprenda a usar funções como **MAX()**, **MIN()**, **AVG()** e **COUNT()** para realizar cálculos e análises avançadas em seus dados. Utilize uma cláusula **HAVING** para filtrar os resultados agregados necessários conforme.



# SELECIONANDO O MAIOR NÚMERO



Para encontrar o maior valor em uma coluna, use a função MAX().

**Exemplo:** Seleccionando o Maior Número de Vendas

```
maior_numero

SELECT MAX(ValorVenda) AS MaiorVenda FROM Vendas;
```

Este comando retorna o maior valor de venda registrado.



# SELECIONANDO O MENOR NÚMERO



Para encontrar o menor valor em uma coluna, use a função MIN().

**Exemplo:** Seleccionando o Menor Número de Vendas

```
menor_numero

SELECT MIN(ValorVenda) AS MenorVenda FROM Vendas;
```

Este comando retorna o menor valor de venda registrado.





# CALCULANDO A MÉDIA



Para calcular a média de uma coluna, usamos a função `AVG()`.

**Exemplo:** Calculando a Média de Vendas

```
calculador_media

SELECT AVG(ValorVenda) AS MediaVendas FROM Vendas;
```

Este comando calcula a média dos valores de venda.



# CONTAGEM DE REGISTROS



Para contar o número de registros em uma coluna, usamos a função COUNT().

**Exemplo:** Contando o Número de Clientes

```
contagem_de_registros

SELECT COUNT(*) AS NumeroDeClientes FROM Clientes;
```

Este comando conta o número total de clientes na tabela.



# FILTRANDO RESULTADOS AGREGADOS



A cláusula HAVING é usada em conjunto com funções agregadas para filtrar os resultados após a agregação.

## Exemplo: Filtrando Vendas Acima da Média

```
vendas_acima_da_media

SELECT ClienteID, AVG(ValorVenda) AS MediaVendas
FROM Vendas
GROUP BY ClienteID
HAVING AVG(ValorVenda) > 1000;
```

Este comando seleciona os clientes cuja média de vendas é superior a 1000.





# 06

# AGRADECIMENTOS

---



# PRÓXIMOS PASSOS



Recomendações para continuar aprendendo SQL incluem a prática regular, realização de projetos práticos e a exploração de tópicos avançados como subconsultas, transações e índices.

## DICAS DE PRÁTICA



- Crie e manipule suas próprias tabelas em um banco de dados local ou em um serviço de nuvem.
- Resolva desafios de SQL em plataformas online.
- Participe de comunidades e fóruns de discussão sobre SQL.



# OBRIGADO POR LER ATÉ AQUI



Este eBook foi criado com IA e diagramado por humano.  
O passo a passo se encontra no meu GitHub.

Nosso objetivo é fornecer um recurso didático em SQL,  
não foi realizada uma avaliação cuidadosa humana e pode  
conter erros gerados por IA.

Agradecemos sua compreensão e esperamos que este  
material ajude em sua jornada de aprendizado!



<https://github.com/RailaCarvalho/eBook-SQL-Reloaded>



Autora



Raila Carvalho

[GitHub](#) | [LinkedIn](#) | [Instagram](#)







"Sem dados, você é apenas mais  
uma pessoa com uma opinião."  
- W. Edwards Deming

