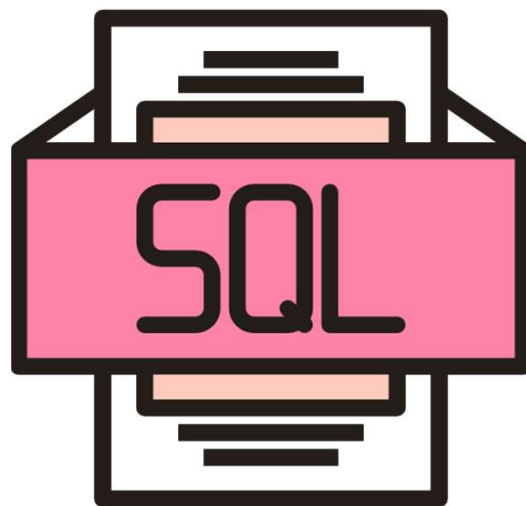


SQL Simples: O Guia



Ana Flávia

Índice

1) Introdução ao SQL

- História e evolução do SQL
- O que é SQL?
- Importância do SQL no desenvolvimento de bancos de dados

2) Conceitos Básicos de SQL

- Estrutura de um banco de dados relacional
- Tabelas, linhas e colunas
- Tipos de dados no SQL
- Criação de um banco de dados e tabelas

3) Manipulação de Dados

- Comando SELECT: Consultas básicas e avançadas
- Comando INSERT: Inserindo dados em tabelas
- Comando UPDATE: Atualizando dados existentes
- Comando DELETE: Removendo dados de tabelas

4) Funções e Operadores em SQL

- Funções agregadas (SUM, AVG, COUNT, MAX, MIN)
- Funções de texto (CONCAT, SUBSTRING, LENGTH)
- Funções de data e hora (NOW, DATEADD, DATEDIFF)
- Operadores lógicos e aritméticos

5) Aplicações Avançadas de SQL

- Joins: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN
- Subconsultas e consultas correlacionadas
- Vistas (Views) e Procedimentos Armazenados (Stored Procedures)
- Índices e Otimização de Consultas



Capítulo 1: Introdução ao SQL

Introdução ao SQL

O SQL, ou Structured Query Language, foi desenvolvido na década de 1970 pela IBM para manipulação e gerenciamento de dados armazenados em bancos de dados relacionais. Desde então, SQL se tornou o padrão para sistemas de gerenciamento de banco de dados relacionais (RDBMS), e continua evoluindo com novas funcionalidades e aprimoramentos.

- O que é SQL? Qual é a sua importância?

SQL é uma linguagem padrão utilizada para realizar consultas, inserções, atualizações e exclusões de dados em um banco de dados. Ele permite a interação com o banco de dados de maneira estruturada e eficiente.

A habilidade de manipular e gerenciar dados é crucial em diversas áreas, incluindo desenvolvimento web, análise de dados, e administração de sistemas. SQL fornece as ferramentas necessárias para essas operações, tornando-se uma habilidade essencial para profissionais de TI.



Capítulo 2: Conceitos Básicos de SQL

Conceitos Básicos de SQL

- **Estrutura de um Banco de Dados Relacional**

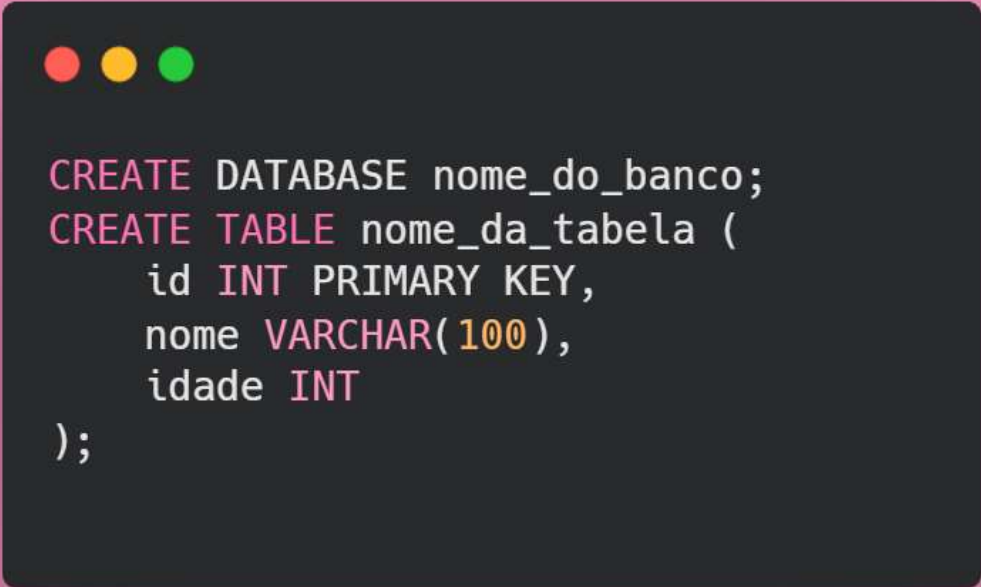
Um banco de dados relacional organiza os dados em tabelas, que consistem em linhas (registros) e colunas (campos). Cada tabela representa uma entidade única, como clientes, produtos ou vendas.

- Tabelas, Linhas e Colunas
- Tabelas: Estruturas que armazenam dados.
- Linhas: Cada registro individual dentro de uma tabela.
- Colunas: Definem os tipos de dados que cada registro pode conter.

- **Tipos de Dados no SQL**

SQL suporta diversos tipos de dados, incluindo inteiros, strings, datas, e booleanos. A escolha do tipo de dado correto é importante para otimização e integridade dos dados.

- Para criar um banco de dados e tabelas, utilizamos comandos SQL específicos:



```
CREATE DATABASE nome_do_banco;  
CREATE TABLE nome_da_tabela (  
    id INT PRIMARY KEY,  
    nome VARCHAR(100),  
    idade INT  
);
```




Capítulo 3: Manipulação de Dados

Manipulação de Dados

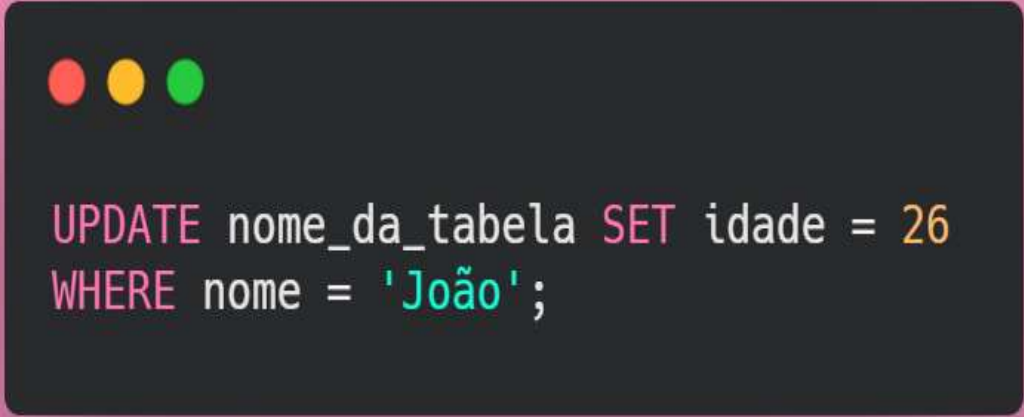
- **Comando SELECT:** Consultas Básicas e Avançadas. O comando SELECT é usado para buscar dados de uma tabela. Exemplos:

```
SELECT * FROM nome_da_tabela;  
SELECT nome, idade FROM nome_da_tabela  
WHERE idade > 30;
```

- **Comando INSERT:** Inserindo Dados em Tabelas. Para adicionar novos registros em uma tabela, usamos o comando INSERT:

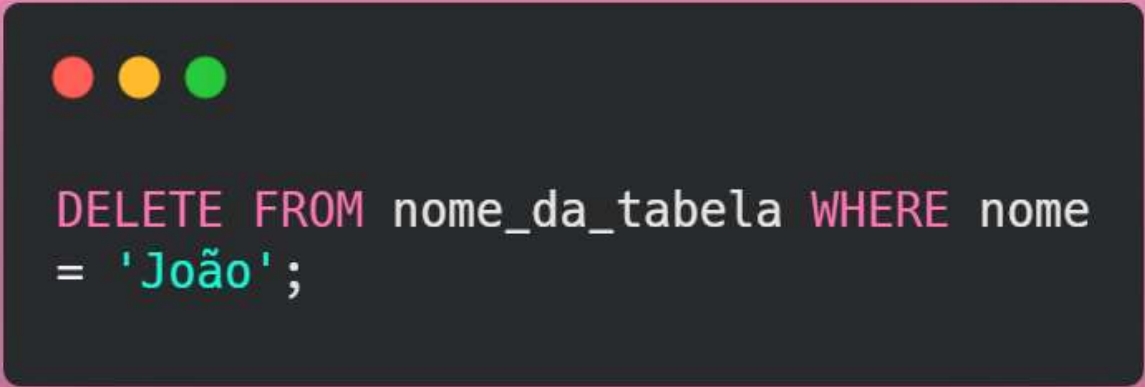
```
INSERT INTO nome_da_tabela (nome,  
idade) VALUES ('João', 25);
```

- **Comando UPDATE:** Atualizando Dados Existentes. O comando UPDATE modifica dados existentes em uma tabela:



```
UPDATE nome_da_tabela SET idade = 26  
WHERE nome = 'João';
```

- **Comando DELETE:** Removendo Dados de Tabelas. Para remover registros, utilizamos o comando DELETE:



```
DELETE FROM nome_da_tabela WHERE nome  
= 'João';
```



Capítulo 4: Funções e Operadores em SQL

Funções e Operadores em SQL

- **Funções Agregadas:**

- **SUM:** Soma dos valores
- **AVG:** Média dos valores
- **COUNT:** Conta o número de registros
- **MAX:** Valor máximo
- **MIN:** Valor mínimo

- **Funções de Texto**

- **CONCAT:** Concatena strings
- **SUBSTRING:** Extrai parte de uma string
- **LENGTH:** Retorna o comprimento da string

- **Funções de Data e Hora:**

- **NOW:** Data e hora atual
- **DATEADD:** Adiciona intervalo de tempo a uma data
- **DATEDIFF:** Diferença entre duas datas

- **Operadores Lógicos e Aritméticos**

- **AND, OR, NOT:** Operadores lógicos
- **+, -, *, /:** Operadores aritméticos




Capítulo 5: Aplicações Avançadas de SQL

Aplicações Avançadas de SQL

Joins: INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN


Joins são usados para combinar registros de duas ou mais tabelas:



```
SELECT a.nome, b.produto FROM clientes  
a INNER JOIN pedidos b ON a.id =  
b.cliente_id;
```

Subconsultas e Consultas Correlacionadas


Subconsultas são consultas dentro de outras consultas:



```
SELECT nome FROM clientes WHERE idade  
> (SELECT AVG(idade) FROM clientes);
```


Vistas (Views) e Procedimentos Armazenados (Stored Procedures)

Vistas são consultas armazenadas:



```
CREATE VIEW clientes_maiores AS SELECT  
* FROM clientes WHERE idade > 30;
```


Procedimentos armazenados são scripts SQL armazenados no banco de dados:



```
CREATE PROCEDURE  
obter_clientes_maiores() BEGIN SELECT  
* FROM clientes WHERE idade > 30; END;
```


Índices e Otimização de Consultas

Índices melhoram a performance das consultas:



```
CREATE INDEX idx_nome ON clientes  
(nome);
```

Fim.