



SQL

Uma Odisseia na Análise de Dados

Guia prático de comandos

Introdução

SQL (Structured Query Language) é a linguagem usada para gerenciar bancos de dados relacionais. Neste eBook, você aprenderá os principais comandos de SQL, explicados de forma simples e prática, sempre com exemplos reais.


1. Criando um Banco de Dados e uma Tabela

Antes de armazenar dados, precisamos de um banco e uma tabela.

```
CREATE DATABASE loja;

USE loja;

CREATE TABLE clientes (
    id INT PRIMARY KEY AUTO_INCREMENT,
    nome VARCHAR(100) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    telefone VARCHAR(20)
);
```

 Criamos um banco chamado **loja** e uma tabela **clientes** com um ID automático e campos obrigatórios.



2. Inserindo Dados na Tabela

Agora vamos cadastrar clientes na loja.

```
INSERT INTO clientes (nome, email, telefone)
VALUES ('Maria Silva', 'maria@email.com', '11987654321');
```

Dica: Podemos inserir vários de uma vez:

```
INSERT INTO clientes (nome, email, telefone) VALUES
('Carlos Souza', 'carlos@email.com', '11912345678'),
('Ana Pereira', 'ana@email.com', '11965432100');
```



3. Buscando Dados no Banco

Para visualizar os clientes cadastrados, usamos SELECT.

.

```
SELECT * FROM clientes;
```

Podemos filtrar por nome:

```
SELECT * FROM clientes WHERE nome = 'Maria Silva';
```



4. Atualizando Informações

Se um cliente mudou de telefone, podemos atualizar assim:

```
UPDATE clientes SET telefone = '11999998888' WHERE id = 1;
```



Atualiza o telefone do cliente com id = 1.



5. Excluindo Registros

Se um cliente for removido da loja, excluimos assim:

```
DELETE FROM clientes WHERE id = 2;
```



Remove o cliente com id = 2.

Atenção! Sem WHERE, o comando apagará todos os registros!



6. Ordenando e Limitando Resultados

Para listar os clientes em ordem alfabética:

```
SELECT * FROM clientes ORDER BY nome ASC;
```



Usa ASC para crescente e DESC para decrescente.

Para mostrar apenas os 5 primeiros registros:

```
SELECT * FROM clientes LIMIT 5;
```



7. Criando Relacionamentos

Se tivermos uma tabela de **pedidos**, podemos relacioná-la à tabela **clientes**:

```
CREATE TABLE pedidos (  
  id INT PRIMARY KEY AUTO_INCREMENT,  
  cliente_id INT,  
  produto VARCHAR(100),  
  valor DECIMAL(10,2),  
  FOREIGN KEY (cliente_id) REFERENCES clientes(id)  
);
```



Ligamos os pedidos aos clientes com FOREIGN KEY.

8. Unindo Informações com JOIN

Para listar os pedidos junto com os dados do cliente:

```
SELECT clientes.nome, pedidos.produto, pedidos.valor  
FROM clientes  
JOIN pedidos ON clientes.id = pedidos.cliente_id;
```

→ O JOIN conecta informações das duas tabelas.


9. Contando e Agrupando Dados

Para contar quantos pedidos cada cliente fez:

```
SELECT clientes.nome, COUNT(pedidos.id) AS total_pedidos  
FROM clientes  
LEFT JOIN pedidos ON clientes.id = pedidos.cliente_id  
GROUP BY clientes.nome;
```

→ COUNT() conta registros e GROUP BY agrupa os dados.

Conclusão


Agora você já conhece os principais comandos SQL! Com essas bases, você pode gerenciar bancos de dados de forma eficiente. Continue praticando e explorando novos desafios! 

Créditos e Referências

Este eBook foi criado com o objetivo de oferecer um guia prático sobre SQL, trazendo conceitos fundamentais e exemplos aplicáveis ao dia a dia.

Conteúdo e estrutura: Desenvolvido com assistência de inteligência artificial para organizar e explicar os principais comandos SQL de forma objetiva e didática.

Capa: Gerada com técnicas de arte digital retro-futurista, representando a fusão entre bancos de dados e exploração computacional.

Agradeço a todos que contribuíram direta ou indiretamente para a construção deste material. Que este conteúdo seja útil para aprimorar seus conhecimentos em SQL! 

Autor: Otávio Braz