



# DOMINANDO CRUD

## Guia Rápido para Devs



# 1. OBJETIVO

Bem-vindo ao nosso Ebook sobre CRUD em bancos de dados! Se você é um desenvolvedor em busca de aprimorar suas habilidades e entender melhor como criar, ler, atualizar e deletar dados de maneira eficiente, você está no lugar certo.

CRUD é uma sigla que representa as quatro operações básicas para gerenciar dados em um banco de dados: **Create** (Criar), **Read** (Ler)\*\* , **Update** (Atualizar) e **Delete** (Deletar). Essas operações são fundamentais para qualquer aplicação que manipule informações, seja ela uma pequena aplicação pessoal ou um sistema corporativo complexo.

Dominar o CRUD é essencial para construir sistemas robustos e eficientes. Este Ebook foi criado pensando em você, desenvolvedor, que deseja aprimorar seus conhecimentos e aplicar as melhores práticas em seus projetos.

Ao longo deste Ebook, vamos explorar de forma prática e direta cada operação do CRUD, fornecendo exemplos claros e dicas valiosas. Nosso objetivo é que você termine a leitura com confiança para implementar e otimizar essas operações em suas próprias aplicações.

Prepare-se para mergulhar no mundo do CRUD e elevar o nível dos seus projetos de desenvolvimento!

# 1. CREATE

A operação Create é usada para criar tabelas onde podemos inserir novos dados em um banco de dados. Esta operação é fundamental para adicionar registros e expandir a quantidade de informações armazenadas. Vamos ver como isso funciona com exemplos de código.

## CRIANDO A TABELA:

Primeiro, precisamos criar a tabela onde os dados serão armazenados. Suponha que queremos armazenar informações de usuários em uma tabela chamada usuarios. Para criar essa tabela, usamos o seguinte comando SQL:

```
CREATE TABLE usuarios (  
    id INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(100),  
    email VARCHAR(100)  
);
```

## INSERINDO DADOS NA TABELA:

Depois de criar a tabela, podemos inserir novos registros usando a instrução INSERT INTO. Veja como adicionar um novo usuário:

```
INSERT INTO usuarios (nome,  
email)  
VALUES ('Lucas Silva',  
'lucas@example.com');
```

## CONCLUSÃO

Dominar a operação CREATE é essencial para adicionar registros ao seu banco de dados de maneira eficaz. Com os exemplos fornecidos, você aprendeu a criar uma tabela e inserir dados usando comandos SQL. Esses fundamentos são a base para construir sistemas robustos e eficientes.

## 2. READ

A operação Read é usada para consultar e recuperar dados de um banco de dados. Quando você lê dados, está extraindo informações específicas de uma tabela. Vamos ver como isso funciona na prática com exemplos de código em SQL.

### CONSULTANDO TODOS OS

#### DADOS:

Para recuperar todos os registros de uma tabela, usamos a instrução `SELECT`. Suponha que queremos ler todos os dados da tabela `usuarios`:



```
SELECT * FROM usuarios;
```

### CONSULTANDO DADOS

#### ESPECÍFICOS:

Podemos filtrar os resultados usando a cláusula `WHERE`. Por exemplo, para encontrar o usuário com o nome "Lucas Silva":




```
SELECT * FROM usuarios WHERE  
nome = 'Lucas Silva';
```

## 2. READ

### SELECIONANDO COLUNAS

#### ESPECÍFICAS:

Às vezes, você só precisa de algumas colunas específicas. Por exemplo, para recuperar apenas os nomes e emails dos usuários:



```
SELECT nome, email FROM  
usuarios;
```

### CONCLUSÃO

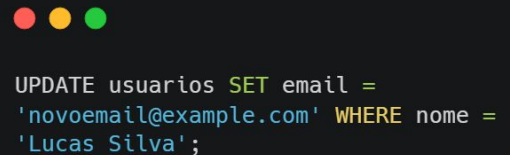
A operação Read é crucial para consultar e recuperar dados de um banco de dados. Com os exemplos fornecidos, você aprendeu a selecionar todos os registros, filtrar dados específicos, selecionar colunas específicas e ordenar resultados usando comandos SQL. Esses fundamentos são essenciais para extrair informações úteis de seus dados de maneira eficiente.

# 3. UPDATE

A operação Update é usada para modificar dados existentes em um banco de dados. Quando você atualiza um registro, está alterando informações específicas em uma tabela. Vamos ver como isso funciona na prática com exemplos de código em SQL.

## ATUALIZANDO UM REGISTRO:

Para atualizar um registro específico, usamos a instrução UPDATE com a cláusula SET para definir os novos valores e WHERE para especificar qual registro deve ser atualizado. Por exemplo, para atualizar o email do usuário "Lucas Silva":

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top left corner. It contains SQL code for updating a user's email.

```
UPDATE usuarios SET email =  
'novoemail@example.com' WHERE nome =  
'Lucas Silva';
```

## CONCLUSÃO


A operação Update é essencial para modificar dados existentes em um banco de dados de maneira eficiente. Com os exemplos fornecidos, você aprendeu a atualizar registros específicos e múltiplos registros com comandos SQL. Esses fundamentos são cruciais para manter os dados atualizados e precisos em suas aplicações.

# 4. DELETE

A operação DELETE (Deletar) é usada para remover registros de um banco de dados. Quando você deleta um registro, está removendo completamente suas informações da tabela. Vamos ver como isso funciona na prática com exemplos de código em SQL.

## DELETANDO UM REGISTRO:


Para deletar um registro específico, usamos a instrução DELETE FROM com a cláusula WHERE para especificar qual registro deve ser removido. Por exemplo, para deletar o usuário com o nome "Lucas Silva":



```
DELETE FROM usuarios WHERE nome =  
'Lucas Silva';
```

## DELETANDO TODOS OS REGISTROS:

Se você quiser deletar todos os registros de uma tabela, pode usar a instrução DELETE FROM sem a cláusula WHERE. Por exemplo, para deletar todos os usuários:



```
DELETE FROM usuarios;
```

## CONCLUSÃO

O DELETE é essencial para remover registros indesejados de um banco de dados. Com os exemplos fornecidos, você aprendeu a deletar registros específicos e todos os registros de uma tabela usando SQL. Esses fundamentos são vitais para manter seu banco de dados organizado e sem dados desnecessários.

# AGRADECIMENTOS

Agradecemos por acompanhar este Ebook sobre operações CRUD em bancos de dados! Ao longo desta jornada, exploramos as operações Create, Read, Update, e Delete em detalhes, aprendendo como adicionar, ler, atualizar e deletar dados de maneira eficiente.

Esperamos que os exemplos e explicações fornecidos tenham sido úteis para o seu aprendizado e para aprimorar suas habilidades como desenvolvedor de sistemas. O conhecimento adquirido nestas páginas servirá como base sólida para seus projetos futuros e para a construção de sistemas robustos e organizados.

Lembre-se sempre da importância de aplicar esses fundamentos de forma responsável, mantendo seus bancos de dados organizados e livres de dados desnecessários. Continue explorando e aprimorando seu conhecimento, pois a jornada de aprendizado na área de desenvolvimento de software é infinita e repleta de desafios estimulantes.

Desejamos a você muito sucesso em sua jornada como desenvolvedor e esperamos vê-lo novamente em futuros materiais de aprendizado. Obrigado por escolher este Ebook e boa sorte em seus projetos!



