

# SQL Mastery

**Domine a Arte dos Dados Corporativos**



**Julio Okuda**

**SQL**



# Introdução

**SQL (Structured Query Language)** é a linguagem padrão para gerenciar e manipular bancos de dados relacionais. Neste eBook, você aprenderá os principais comandos SQL, essenciais para qualquer profissional que trabalha com dados no mundo corporativo.

## **Exemplo de Banco de Dados:**

Para seguir os exemplos deste eBook, você pode usar o script de exemplo `exampleDB.sql` para criar o banco de dados.



# 01

# Linguagem de Definição de Dados (DDL)

---

# Criando Tabelas com CREATE TABLE

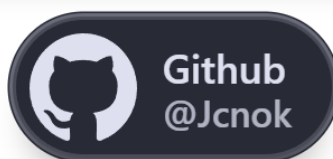


Use CREATE TABLE para criar uma nova tabela.

Exemplo:

```
CREATE by Julio Okuda

1 CREATE TABLE projects (
2     project_id INT PRIMARY KEY,
3     project_name VARCHAR(100),
4     start_date DATE,
5     end_date DATE
6 );
7
8
```



Esta consulta cria uma tabela de projetos com colunas para ID do projeto, nome, data de início e data de término.

# Alterando Tabelas com ALTER TABLE

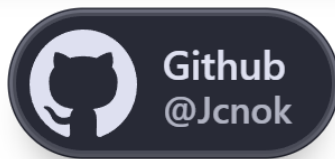


Modifique a estrutura de uma tabela existente.

Exemplo:

```
CREATE by Julio Okuda

1 ALTER TABLE employees
2 ADD COLUMN phone_number VARCHAR(15);
3
4
5
```



Aqui, estamos adicionando uma coluna de número de telefone à tabela de funcionários.

# Excluindo Tabelas com DROP TABLE

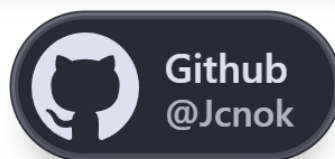


Modifique a estrutura de uma tabela existente.

Exemplo:

```
CREATE by Julio Okuda

1 DROP TABLE old_projects;
2
3
4
5
```



Esta consulta exclui a tabela old\_projects.

# 02

## Linguagem de Manipulação de Dados (DML)

---

# Inserindo Novos Registros com INSERT

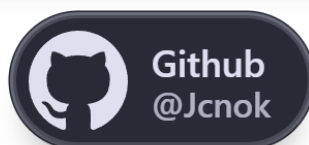


Adicione novos registros em uma tabela.

Exemplo:

CREATE by Julio Okuda

```
1 INSERT INTO
2   EMPLOYEES (FIRST_NAME, LAST_NAME, DEPARTMENT, SALARY)
3 VALUES
4   ('John', 'Doe', 'Marketing', 60000);
5
```



Esta consulta adiciona um novo funcionário ao departamento de Marketing com um salário de 60.000.



# Atualizando Registros com UPDATE

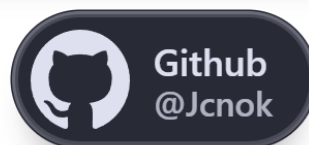


Altere dados existentes em uma tabela.

Exemplo:

```
CREATE by Julio Okuda

1 UPDATE EMPLOYEES
2 SET
3   SALARY = 65000
4 WHERE
5   FIRST_NAME = 'John'
6   AND LAST_NAME = 'Doe';
```



Aqui, estamos atualizando o salário de John Doe para 65.000

# Removendo Registros com DELETE

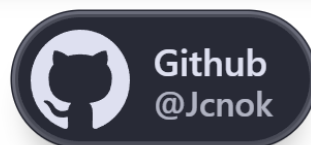


Exclua registros de uma tabela.

Exemplo:

```
CREATE by Julio Okuda

1 DELETE FROM EMPLOYEES
2 WHERE
3   DEPARTMENT = 'HR'
4   AND SALARY < 40000;
```



Esta consulta remove todos os funcionários do departamento de RH que ganham menos de 40.000.

03

# Linguagem de Consulta de Dados (DQL)

---

# Selecionando Dados com SELECT

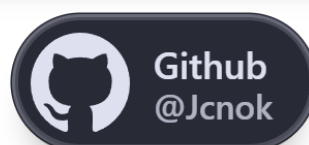


Busque dados de uma tabela.

Exemplo:

```
CREATE by Julio Okuda

1 SELECT
2   FIRST_NAME,
3   LAST_NAME,
4   DEPARTMENT
5 FROM
6   EMPLOYEES
7 WHERE
8   DEPARTMENT = 'Sales';
```



Aqui, estamos buscando os nomes e departamentos dos funcionários que trabalham no departamento de Vendas.

# Ordenando Resultados com ORDER BY

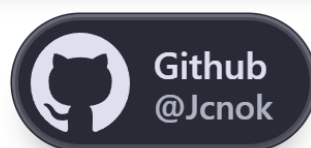


Organize os resultados da sua consulta.

Exemplo:

```
CREATE by Julio Okuda

1 SELECT
2   FIRST_NAME,
3   LAST_NAME,
4   SALARY
5 FROM
6   EMPLOYEES
7 ORDER BY
8   SALARY DESC;
```



Aqui, estamos ordenando os funcionários por salário, do maior para o menor.



# Agrupando Dados com GROUP BY

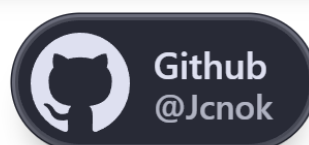


Agrupe resultados para aplicar funções de agregação.

Exemplo:

```
CREATE by Julio Okuda

1 SELECT
2   FIRST_NAME,
3   LAST_NAME,
4   SALARY
5 FROM
6   EMPLOYEES
7 ORDER BY
8   SALARY DESC;
```



Esta consulta agrupa os funcionários por departamento e calcula o número de funcionários e a média salarial em cada departamento.

# Filtrando Dados Agrupados com HAVING

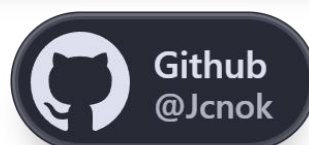


Filtre resultados após o agrupamento.

Exemplo:

```
CREATE by Julio Okuda

1 SELECT
2   DEPARTMENT,
3   AVG(SALARY) AS AVG_SALARY
4 FROM
5   EMPLOYEES
6 GROUP BY
7   DEPARTMENT
8 HAVING
9   AVG(SALARY) > 50000;
```



Aqui, estamos filtrando os departamentos onde a média salarial é superior a 50.000.

# Junção de Tabelas com JOIN

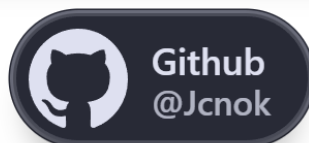


Combine registros de duas ou mais tabelas.

Exemplo:

CREATE by Julio Okuda

```
1 SELECT
2   E.FIRST_NAME,
3   E.LAST_NAME,
4   D.DEPARTMENT_NAME
5 FROM
6   EMPLOYEES E
7 JOIN DEPARTMENTS D ON E.DEPARTMENT=D.DEPARTMENT_NAME;
```



Esta consulta combina a tabela de funcionários com a tabela de departamentos, trazendo o nome do funcionário e o nome do departamento.

# Subconsultas com SUBQUERY

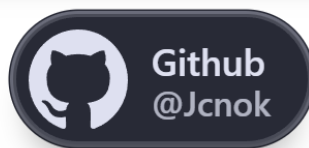


Use uma consulta dentro de outra consulta.

Exemplo:

```
CREATE by Julio Okuda

1  SELECT
2    FIRST_NAME,
3    LAST_NAME
4  FROM
5    EMPLOYEES
6  WHERE
7    SALARY > (
8      SELECT
9        AVG(SALARY)
10     FROM
11       EMPLOYEES);
```



Aqui, estamos buscando os funcionários cujo salário é maior que a média salarial de todos os funcionários.

# 04

## Linguagem de Transação de Dados (DTL)

---



# Iniciando Transações com BEGIN TRANSACTION

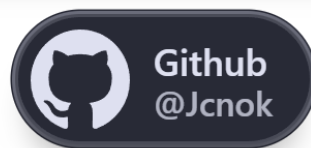


Inicie uma transação para garantir a execução segura de múltiplas operações.

Exemplo:

```
CREATE by Julio Okuda

1 BEGIN TRANSACTION;
2
3 UPDATE ACCOUNTS
4 SET
5     BALANCE = BALANCE - 1000
6 WHERE
7     ACCOUNT_ID = 1;
8
9 UPDATE ACCOUNTS
10 SET
11     BALANCE = BALANCE + 1000
12 WHERE
13     ACCOUNT_ID = 2; COMMIT;
```



Aqui, estamos transferindo 1000 de uma conta para outra de forma segura.

# Desfazendo Transações com ROLLBACK

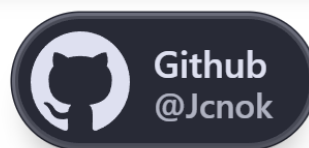


Reverta uma transação em caso de erro.

Exemplo:

```
CREATE by Julio Okuda

1 BEGIN TRANSACTION;
2
3 UPDATE ACCOUNTS
4 SET
5     BALANCE = BALANCE - 1000
6 WHERE
7     ACCOUNT_ID = 1;
8
9 -- Erro ocorre aqui
10 ROLLBACK;
```



Se um erro ocorrer, a transação será revertida e nenhum valor será transferido.

# 05

## Linguagem de Controle de Dados (DCL)

---

# Concedendo Permissões com GRANT



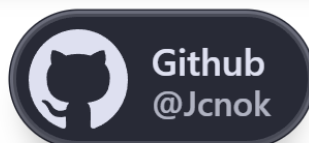
Dê permissões a usuários ou roles.

Exemplo:



CREATE by Julio Okuda

```
1  -- Criar a função user1
2  CREATE ROLE USER1 LOGIN PASSWORD 'password';
3  -- Conceder permissão GRANT
4  GRANT
5  SELECT
6  ,
7  INSERT ON EMPLOYEES TO USER1;
```



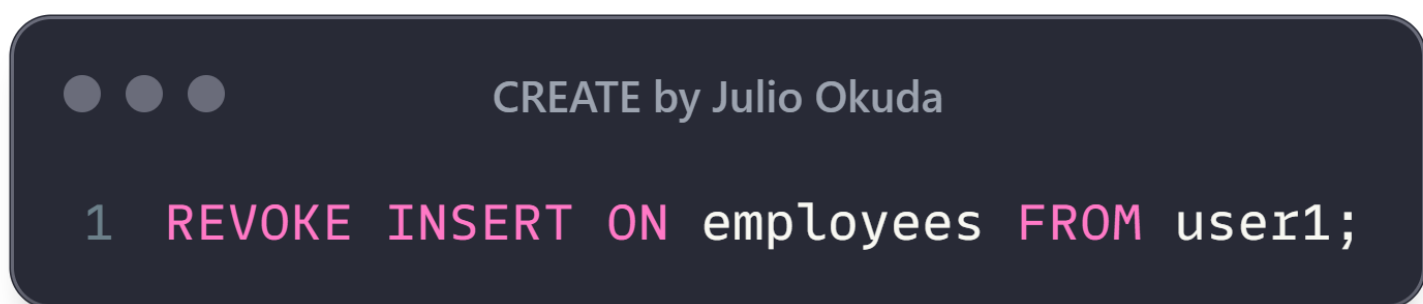
Esta consulta concede permissões de seleção e inserção na tabela de funcionários ao usuário1.

# Revogando Permissões com REVOKE

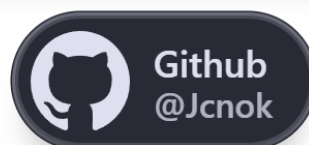


Remova permissões de usuários ou roles.

Exemplo:



```
1 REVOKE INSERT ON employees FROM user1;
```



Aqui, estamos removendo a permissão de inserção na tabela de funcionários do usuário1.



# 06

## Visualizações com VIEW

---

# Criando Views com CREATE VIEW

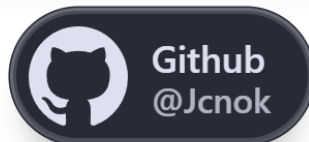


Crie uma view para simplificar consultas complexas.

Exemplo:

```
CREATE by Julio Okuda

1 CREATE VIEW EMPLOYEE_SUMMARY AS
2 SELECT
3   DEPARTMENT,
4   COUNT(*) AS NUM_EMPLOYEES,
5   AVG(SALARY) AS AVG_SALARY
6 FROM
7   EMPLOYEES
8 GROUP BY
9   DEPARTMENT;
```



Esta view resume o número de funcionários e a média salarial por departamento.

# Usando Views

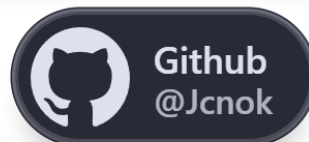


Utilize views em suas consultas.

Exemplo:

```
CREATE by Julio Okuda

1 SELECT
2   *
3 FROM
4   EMPLOYEE_SUMMARY
5 WHERE
6   AVG_SALARY > 50000;
```



Aqui, estamos utilizando a view para buscar departamentos com média salarial superior a 50.000..

# AGRADECIMENTOS

---

# OBRIGADO POR LER ATÉ AQUI



Esse Ebook, foi gerado utilizando ferramentas de IA(chatGPT e DALL-E3) e diagramada por mim.

.

Esse conteúdo foi gerado com fins didáticos de construção, porém os códigos foram todos testados via PGAdmin 4.



<https://github.com/Jcnok>

Meu Perfil DIO



JULIO CESAR OKUDA



Principais Habilidades

1 Soft Skill

2 SQL

3 AWS

4 Machine Learning

Últimas Conquistas





033  
Habilidades Desenvolvidas

251  
Conquistas

027  
Projetos Realizados

005  
Artigos Escritos

SQL Mastery - Julio Okuda

033

002

28