



MiniCurso de SQL

“A memória é a melhor amiga e a pior inimiga do homem.”
- Gilbert Parket

Prof. MSc. Cloves Rocha

#ClovesRochaDigital #SQL #MySQL #DDL #DML #DQL



Agenda

- Introdução ao minicurso
- Principais comandos da SQL
 - DQL
 - DDL e
 - DML
- Bônus
- Dúvidas
- Referências Bibliográficas.

Introdução ao minicurso

Bem-vindos ao minicurso sobre os Principais Comandos do SQL. SQL, ou Linguagem de Consulta Estruturada (do inglês, Structured Query Language), é a espinha dorsal de muitas aplicações de bases de dados no mundo moderno. Utilizada para criar, gerir, e manipular bases de dados relacionais, a compreensão profunda do SQL abre portas para a análise de dados, desenvolvimento de aplicações, administração de sistemas de informação, entre muitas outras oportunidades. Neste minicurso, centrar-nos-emos em categorias específicas dos comandos SQL, nomeadamente:

1. **DQL** - Data Query Language: Os comandos DQL permitem-nos consultar dados. Aqui, iremos explorar o fundamental comando ``SELECT``, que nos permite selecionar dados de uma ou mais tabelas.
2. **DML** - Data Manipulation Language: Os comandos DML afetam os dados nas tabelas. Vamos estudar comandos essenciais como ``INSERT``, para inserir novos dados; ``UPDATE``, para modificar dados existentes; e ``DELETE``, para remover dados.
3. **DDL** - Data Definition Language: Os comandos DDL permitem criar, modificar e eliminar estruturas de banco de dados, como tabelas. Vamos nos concentrar em comandos como ``CREATE TABLE``, ``ALTER TABLE`` e ``DROP TABLE``.

Introdução ao minicurso

Ao longo deste minicurso de SQL, você irá não só aprender a usar eficazmente estes comandos para manipular dados e estruturas de banco de dados, mas também irá entender como podem ser aplicados em cenários do mundo real. Resolve-se assim, não apenas a técnica, mas a aplicabilidade prática dos conceitos aprendidos.

Dúvidas serão bem-vindas a qualquer momento. Nosso objetivo é garantir que você saia deste minicurso com uma compreensão sólida dos comandos SQL e como usá-los efetivamente.

Referências Bibliográficas: Os materiais recomendados incluem uma seleção de livros, documentação online e artigos que fornecem uma base sólida para o entendimento e aprofundamento do tema.

Prepare-se para mergulhar no mundo do SQL, fortalecer suas habilidades e abrir novos horizontes na sua carreira ou nos seus projetos pessoais!

Mas afinal, você sabe o que é **SQL**?



SQL ou Structured Query Language (Linguagem de Consulta Estruturada) é uma linguagem padrão de gerenciamento de dados que interage com os principais bancos de dados baseados no modelo relacional.

Alguns dos principais sistemas que utilizam SQL são: **Oracle**, PostgreSQL, Firebird, **MySQL**, entre outros.

Mas afinal, você sabe o que é SQL?



Sendo assim, utilizando a linguagem SQL, os desenvolvedores podem ter uma comunicação com o banco de dados de maneira simples e ágil a partir dos seus comandos.

DQL, DDL e DML



DQL, DDL e DML são consideradas subconjuntos da linguagem SQL.

Estes comandos são os que permitem a escrita de códigos que irão criar novas estruturas como tabelas, índices, visões, entre outros, permitindo que dados sejam inseridos ou modificados para consultas posteriores.

Linguagem de Consulta de Dados

```
select * from departments;
```

DQL ou Data Query Language possui apenas um único comando: O SELECT.

O **SELECT** é um dos principais comandos utilizados em SQL, pois com ele é possível realizar consultas aos dados que pertencem a uma determinada tabela. É um comando composto de várias opções que permite a elaboração de consultas das mais simples a mais elaboradas.

Um exemplo que podemos fornecer a cerca do uso do SELECT é quando possuímos em nosso banco de dados vários usuários e precisamos listar todos eles de uma só vez. Desta forma utilizaremos o seguinte comando:

```
SELECT * FROM usuarios;
```

Ao ser executado, o comando acima irá retornar todos os usuários armazenados no banco de dados.



Linguagem de Definição de Dados

DDL ou Data Definition Language (Linguagem de Definição de dados) permite ao usuário definir as novas tabelas e os elementos que serão associados a elas. É responsável pelos comandos de criação e alteração no banco de dados, sendo composto por três comandos: CREATE, ALTER e DROP.

O comando CREATE DATABASE é responsável pela criação de um novo banco de dados vazio, conforme podemos ver abaixo:

```
CREATE DATABASE banco_teste;
```

Ao ser executado, estaremos criando o banco de dados chamado "banco_teste".



Linguagem de Definição de Dados

```
create table DEPARTMENTS (  
  deptno      number,  
  name        varchar2(50) not null,  
  location    varchar2(50),  
  constraint pk_departments primary key  
    (deptno)  
);
```

Já o comando CREATE TABLE irá criar uma nova tabela. Os bancos de dados relacionais guardam seu dados dentro de tabelas que são divididas em colunas. Desta forma, veremos abaixo a criação de uma tabela de usuário. Ao criar, especificaremos as suas colunas e quais tipos de dados elas irão receber (neste caso, um ID e o nome do usuário).

O comando CREATE DATABASE é responsável pela criação de um novo banco de dados vazio, conforme podemos ver abaixo:

```
CREATE TABLE usuario (id INT, nome VARCHAR (255));
```

O comando ALTER, por sua vez, é o comando utilizado para alterar uma tabela ou um banco de dados já existente.



Linguagem de Definição de Dados



No exemplo abaixo estaremos **adicionando** uma nova coluna a nossa tabela de usuário criada acima. Esta nova coluna “idade” será criada após a nossa coluna “nome”, desta forma, usaremos:

```
ALTER TABLE usuario ADD idade INT AFTER nome;
```



Linguagem de Definição de Dados

O comando DROP é utilizado para remoção de uma tabela ou do banco de dados por completo. Desta forma para **remover** um banco de dados por completo, basta inserirmos o seguinte comando:

```
DROP DATABASE banco_teste;
```

Ou **excluir** uma tabela utilizando o seguinte comando:

```
DROP TABLE usuario;
```



Linguagem de Manipulação de Dados

```
insert into departments (name, location)
values
    ('Finance', 'New York');
```

```
insert into departments (name, location)
values
    ('Development', 'San Jose');
```

O DML ou Data Manipulation Language (Linguagem de Manipulação de Dados) interage diretamente com os dados dentro das tabelas. Possui três comandos para esta manipulação: **INSERT**, **UPDATE** e **DELETE**.

Estas instruções são utilizadas nas consultas e modificações dos dados que estarão armazenados dentro do banco de dados.

No comando **INSERT** estaremos inserindo dados a uma ou mais tabela de um banco de dados. Desta forma, abaixo veremos a sua sintaxe:

```
INSERT INTO usuario (id, nome, idade) VALUES (1, 'Cloves', 50);
```

O comando acima irá criar o usuário Cloves, com idade de 50 anos e ID 1.



Linguagem de Manipulação de Dados



O **UPDATE** é utilizado para atualizar os dados de uma ou mais tabelas.

```
UPDATE usuario SET nome = 'Cloves Rocha' WHERE id = 1;
```

Ao executar o comando acima, estaremos alterando o nome do usuário que possui o ID 1 para “Cloves Rocha”.

Já o comando **DELETE**, como seu próprio significado já diz, utilizaremos para excluir os dados de uma ou mais tabela em nosso banco de dados.

```
DELETE FROM usuario WHERE id = 1
```

Desta forma, ao executar o comando acima, estaremos excluindo o usuário que possui o ID 1 do nosso banco de dados.



BÔNUS



dbdiagram.io



O dbdiagram.io é uma ferramenta visual para criar diagramas de banco de dados. Enquanto o uso direto de SQL não é o principal foco dessa ferramenta, ela permite que você exporte o diagrama de banco de dados que criou para o formato SQL. Abaixo, estão alguns exemplos de como você poderia usar SQL gerado a partir de um diagrama criado no dbdiagram.io ou interpretar as tabelas e relações criadas na ferramenta para o SQL.

1. Criar Tabela

Imagine que você criou um diagrama com uma tabela chamada `Users`. No dbdiagram.io, você teria definido os campos e os tipos de dados visualmente. O SQL gerado para criar esta tabela pode parecer com:

```
```sql
CREATE TABLE Users (
 id INT PRIMARY KEY,
 name VARCHAR(255),
 email VARCHAR(255) UNIQUE,
 created_at TIMESTAMP
);
```
```

2. Criar Relacionamentos

Se você estabeleceu uma relação de chave estrangeira entre `Users` e outra tabela chamada `Orders`, isso indica que um usuário pode ter várias ordens. O SQL para a tabela `Orders` poderia ser:

```
```sql
CREATE TABLE Orders (
 id INT PRIMARY KEY,
 user_id INT,
 order_date DATE,
 total DECIMAL(10,2),
 FOREIGN KEY (user_id) REFERENCES Users(id)
);
```
```

3. Inserir Dados

Após a criação das tabelas, você poderia querer inserir alguns dados nelas. Isso não é feito diretamente no dbdiagram.io, mas o SQL resultante pode ser:

```
```sql
INSERT INTO Users (id, name, email, created_at) VALUES (1, 'Cloves', 'cloves@example.com', NOW());
INSERT INTO Orders (id, user_id, order_date, total) VALUES (101, 1, '2023-04-01', 299.99);
```
```

4. Consultar Dados

Com as tabelas e os relacionamentos criados, você poderá querer realizar consultas SQL para buscar dados específicos, como encontrar todas as ordens de um usuário:

```
```sql
SELECT * FROM Orders WHERE user_id = 1;
```
```

5. Atualizar Dados

Em algum momento, você pode precisar atualizar os dados existentes. Supondo a necessidade de atualizar o nome do usuário:

```
```sql
UPDATE Users SET name = 'Cloves Rocha' WHERE id = 1;
```
```

6. Deletar Dados

Caso seja necessário remover dados, o seguinte comando SQL pode ser utilizado:

```
```sql
DELETE FROM Orders WHERE id = 101;
```
```

Embora o dbdiagram.io seja uma ferramenta para modelagem visual, entender como as estruturas criadas nela se traduzem para SQL é essencial para a implementação e manipulação desses modelos no seu banco de dados.

DÚVIDAS?



Digitalize-me!

Referências Bibliográficas

Artigo - Principais comandos SQL

Disponível em:

<https://www.treinaweb.com.br/blog/principais-comandos-sql>



@clovesrocha



eBook -
Descomplicando Git e
GitHub

...



eBook -
Descomplicando
Programação com
Python

...



eBook - Machine
Learning com Python
(Códigos)

...



eBook - Machine
Learning - Primeiros
Passos (Conceitos)

...