**COMANDOS – SQL**

**1. CREATE DATABASE**

Este comando é usado para **criar um novo banco de dados.**

**Sintaxe:**

SQL

CREATE DATABASE nome\_do\_banco;

**Exemplo:**

SQL

CREATE DATABASE Floricultura;

Isso cria um banco de dados chamado "Floricultura".

**2. CREATE TABLE**

O comando **CREATE TABLE** é utilizado para **criar uma nova tabela** dentro de um banco de dados, com colunas e seus respectivos tipos de dados.

**Sintaxe:**

SQL

CREATE TABLE nome\_da\_tabela (

nome\_coluna tipo\_dado restrições,

nome\_coluna tipo\_dado restrições,

);

**Exemplo:**

SQL

CREATE TABLE Cliente (

id\_cliente INT AUTO\_INCREMENT PRIMARY KEY,

nome VARCHAR(50),

sobrenome VARCHAR(50),

telefone VARCHAR(20)

);

Isso cria uma tabela "Cliente" com três colunas: `id\_cliente`, `nome`, `sobrenome` e `telefone`.

**3. ALTER TABLE**

O comando **ALTER TABLE** é utilizado para **modificar uma tabela existente**, como adicionar, modificar ou remover colunas.

**Sintaxe:**

SQL

ALTER TABLE nome\_da\_tabela

ADD nome\_coluna tipo\_dado;

**Exemplo:**

SQL

ALTER TABLE Cliente

ADD endereco VARCHAR(100);

Esse exemplo adiciona uma nova coluna `endereco` à tabela "Cliente".

**4. INSERT INTO**

O comando **INSERT INTO** é utilizado para **inserir novos registros (linhas) em uma tabela**.

**Sintaxe:**

SQL

INSERT INTO nome\_da\_tabela (coluna1, coluna2, ...)

VALUES (valor1, valor2, ...);

**Exemplo:**

SQL

INSERT INTO Cliente (nome, sobrenome, telefone)

VALUES ('Maria', 'Silva', '11999999999');

Isso insere um cliente chamado Maria Silva na tabela "Cliente".

**5. SELECT FROM**

O comando **SELECT** é usado para **consultar dados de uma ou mais tabelas.**

**Sintaxe:**

SQL

SELECT colunas

FROM nome\_da\_tabela

WHERE condição;

**Exemplo:**

SQL

SELECT nome, telefone FROM Cliente

WHERE sobrenome = 'Silva';

Esse exemplo retorna os nomes e telefones dos clientes cujo sobrenome é "Silva".

**6. DELETE FROM**

O comando **DELETE FROM** é utilizado para **remover registros de uma tabela**.

**Sintaxe:**

SQL

DELETE FROM nome\_da\_tabela

WHERE condição;

**Exemplo:**

SQL

DELETE FROM Cliente

WHERE id\_cliente = 2;

Isso remove o cliente com `id\_cliente` igual a 2 da tabela "Cliente".

**7. UPDATE**

O comando **UPDATE** é utilizado para **atualizar os dados em uma tabela existente**.

**Sintaxe:**

SQL

UPDATE nome\_da\_tabela

SET coluna = novo\_valor

WHERE condição;

**Exemplo:**

SQL

UPDATE Cliente

SET telefone = '11888888888'

WHERE id\_cliente = 1;

Isso atualiza o telefone do cliente com `id\_cliente` 1.

**8. FUNÇÕES DE AGREGAÇÃO**

As funções de agregação são usadas para **realizar cálculos em conjuntos de dados**.

**Exemplos:**

COUNT: Conta o número de registros.

SUM: Soma valores numéricos.

AVG: Calcula a média.

MAX: Retorna o valor máximo.

MIN: Retorna o valor mínimo.

**Exemplo:**

SQL

SELECT COUNT(\*) AS total\_clientes FROM Cliente;

Este comando conta o total de clientes na tabela "Cliente".

**9. INNER JOIN**

O **INNER JOIN** combina registros de duas tabelas com base em uma **condição de correspondência**.

**Sintaxe:**

SQL

SELECT colunas

FROM tabela1

INNER JOIN tabela2

ON tabela1.coluna = tabela2.coluna;

**Exemplo:**

SQL

SELECT Cliente.nome, Compra.numero\_nota\_fiscal

FROM Cliente

INNER JOIN Compra ON Cliente.id\_cliente = Compra.id\_cliente;

Esse comando retorna os nomes dos clientes e seus respectivos números de nota fiscal.

**10. OUTER JOIN**

O **OUTER JOIN** retorna todos os registros da tabela principal e os correspondentes da tabela secundária, mesmo que não haja correspondência.

- **LEFT JOIN** Retorna todos os registros da tabela à esquerda, mesmo que não haja correspondência na tabela à direita.

- **RIGHT JOIN** Retorna todos os registros da tabela à direita, mesmo que não haja correspondência na tabela à esquerda.

**Exemplo (LEFT JOIN):**

SQL

SELECT Cliente.nome, Compra.numero\_nota\_fiscal

FROM Cliente

LEFT JOIN Compra ON Cliente.id\_cliente = Compra.id\_cliente;

Esse comando retorna todos os clientes e suas compras, mesmo que o cliente não tenha feito nenhuma compra.

**11.TRUNCATE TABLE**

O comando **TRUNCATE TABLE** é utilizado para **remover todos os registros de uma tabela** rapidamente, sem deletar a estrutura da tabela.

**Sintaxe:**

SQL

TRUNCATE TABLE nome\_da\_tabela;

**Exemplo:**

SQL

TRUNCATE TABLE Produto;

Isso remove todos os registros da tabela "Produto", deixando-a vazia.

**12. SUBQUERIES**

Uma **subquery** (subconsulta) é uma consulta dentro de outra consulta. Ela pode ser usada para retornar resultados intermediários.

**Sintaxe:**

SQL

SELECT colunas

FROM nome\_da\_tabela

WHERE coluna = (SELECT valor FROM outra\_tabela WHERE condição);

**Exemplo:**

SQL

SELECT nome\_produto

FROM Produto

WHERE preco > (SELECT AVG(preco) FROM Produto);

Isso retorna todos os produtos cujo preço é maior que a média de todos os preços.

**13. ORDER BY**

O comando **ORDER BY** é usado para **ordenar os resultados de uma consulta**.

**Sintaxe:**

SQL

SELECT colunas

FROM nome\_da\_tabela

ORDER BY coluna [ASC|DESC];

**Exemplo:**

SQL

SELECT nome\_produto, preco

FROM Produto

ORDER BY preco ASC;

Isso retorna os produtos ordenados pelo preço em ordem crescente.

**14. BETWEEN**

O **BETWEEN** é usado para **filtrar registros que estejam dentro de um intervalo de valores**.

**Sintaxe:**

SQL

SELECT colunas

FROM nome\_da\_tabela

WHERE coluna BETWEEN valor1 AND valor2;

**Exemplo:**

SQL

SELECT nome\_produto, preco

FROM Produto

WHERE preco BETWEEN 10 AND 30;

Esse exemplo retorna todos os produtos cujo preço esteja entre 10 e 30.

**15. LIKE e NOT LIKE**

O **LIKE** e **NOT LIKE** são usados para **filtrar registros com base em padrões de texto**.

- **O caractere `%` é usado como coringa para representar zero ou mais caracteres.**

**- O caractere `\_` é usado para representar um único caractere**.

**Sintaxe:**

SQL

SELECT colunas

FROM nome\_da\_tabela

WHERE coluna LIKE padrão;

**Exemplo:**

SQL

SELECT nome\_produto

FROM Produto

WHERE nome\_produto LIKE 'Rosa%';

Esse exemplo retorna todos os produtos cujo nome começa com "Rosa".

**Exemplo (NOT LIKE):**

SQL

SELECT nome\_produto

FROM Produto

WHERE nome\_produto NOT LIKE '%Flor%';

Esse comando retorna todos os produtos cujo nome não contém a palavra "Flor".