CREATE DATABASE / SCHEMA: que serve para criar um banco de dados, que é onde vamos armazenar tabelas.

CREATE DATABASE nome\_database;

USE meu\_bd : utilizar o comando USE para selecionar o banco de dados.

USE nome\_database;

CREATE TABLE: esse comando estamos criando uma tabela com colunas e podemos definir o tipo de cada coluna.

CREATE TABLE UF(

  Id INT,

  cidade VARCHAR(30) NOT NULL,

);

SHOW TABLES: Mostram as tabelas do banco de dados atual.

SHOW TABLES;

DESCRIBE minha\_tabela: É um atalho para o comando SHOW COLUMNS FROM;

DESCRIBE tbl\_cidade;

ALTER TABLE: Para alterar as características de uma tabela existente e de suas colunas. É possível incluir, eliminar e alterar colunas, além de configurar manualmente o valor de campos do tipo INCR.

ALTER TABLE UF SET INCR value;

INSERT: se não tiver informações na sua tabela e queira adicioná-las.

INSERT INTO UF (cidade) VALUES (null, "Ilhéus");

SELECT: É um comando para consulta de dados. Ele deve ser acompanhado de uma indicação específica do local onde o dado será consultado (junto do FROM)

SELECT \* FROM UF;

SELECT cidade FROM UF;

ORDER BY:Ordena os dados de acordo com alguma lógica, algum critério específico. Para auxiliar a ordenar as informações dentro de uma coluna para que fique mais fácil a visualização.

SELECT \* FROM UF ORDER BY cidade;

UPDATE: Serve para atualizar dados nas tabelas, usa o WHERE para especificar qual é o dado.

UPDATE UF SET nome="São Paulo" WHERE cidade;

WHERE: Usado para consultar, ajudar a especificar os dados que serão consultados e manipulados.

UPDATE UF SET nome="São Paulo" WHERE cidade;

DELETE: Exclui dados de acordo com um critério definido ou então um dado em específico.

DELETE FROM UF WHERE cidade="São Paulo";

COUNT: Vai contar o número de registros retornados, independentemente de valores nulos.

select count(\\*) from cidade;

GROUP BY: Identifica uma coluna escolhida para fins de agrupamento dos dados. Os dados são então separados em grupos com base nos valores encontrados nessa coluna específica, resultando em uma linha de resultados para cada grupo.

SELECT DEPT, AVG(SALARY)

FROM Q.STAFF

GROUP BY DEPT

LIMIT: Serve para limitar o número de recursos.

SELECT \* FROM tabela LIMIT 10 OFFSET 20;

ASC: Ordenação ascendente.

SELECT \* FROM tbl\_Livro

ORDER BY Nome\_Livro ASC;

DESC: Ordenação descendente.

SELECT Nome\_Livro, Preco\_Livro

FROM tbl\_Livro

ORDER BY Preco\_Livro DESC;

AS: Serve para dar um nome diferente a uma coluna ou tabela ao utilizar o(join) de modo que seja mais fácil ou intuitivo entender os dados retornados.

SELECT Nome\_Livro

AS Livro

FROM tbl\_Livro;

MAX: Retorna o maior valor dentro de um conjunto de valores.

SELECT embalagem , MAX(preco\_de\_lista) AS maior\_preco FROM tabela\_de\_produtos;

MIN: Retorna o menor valor de um conjunto.

SELECT \* FROM FUNCIONARIO

WHERE SALARIOFUNCIONARIO = (SELECT MIN(SALARIOFUNCIONARIO) FROM FUNCIONARIO);

AVG: Retorna o valor médio de um grupo de registros. Só pe aplicada em valores numéricos.

SELECT

   cidade,

   AVG(nota1) AS nota\_media;

SUM: Retorna a somatória de conjunto de registros de umat tabela.

SELECT

   cidade,

   SUM(valor\_mensalidade) AS total\_mensalidade;

BETWEEN: Serve para selecionar registros dentro de um intervalo específico em uma tabela.

SELECT nome\_aluno, data\_conclusao FROM alunos WHERE data\_conclusao BETWEEN '2020-01-01' AND '2020-12-31';

LIKE: Serve para seleção de atributos específicos em uma tabela. Sempre usando o WHERE junto.

SELECT nome FROM alunos WHERE nome LIKE 'J%';

AND: Serve para mostrar um registro se ambas as condições forem verdadeiras.

SELECT \* FROM tbl\_Livro

WHERE ID\_Livro > 2 AND ID\_Autor < 3;

JOIN: Serve para combinar colunas de uma ou mais tabelas em um banco de dados relaciona.