SQL - Uma linguagem estruturada para manipulação de dados. É padronizada para os Bancos de dados relacionais, mas cada gerenciador pode possuir uma extensão própria dessa linguagem.

Conforme figura acima:

Em um banco de dados cliente/servidor, a interface com o usuário fica na estação e se comunica remotamente com o mecanismo de aceso, que é um **sistema gerenciador de banco de dados (SGBD)** rodando no servidor. Quando o **SGBD** recebe um pedido para selecionar alguns dados, ele acessa localmente os dados no servidor e retorna apenas o resultado pedido. No caso de uma atualização, não é necessário nem mesmo retornar um resultado, apenas informar que a atualização foi feita. Por ser uma linguagem de numerosas aplicações, pode manipular objetos de diferentes classes entre as funções de um **SGBD**.

**Definição de Dados (DDL – Data Definition Language )** – permite ao usuário a definição da estrutura e organização dos dados armazenados, e as relações que existem entre eles, palavras chaves: **create**, **revoke**, **grant** e **alter**;

**Manipulação de Dados (DML – Data Manipulation Language**) – permite ao usuário ou a um programa de aplicação manipular dados, a inclusão , remoção, seleção ou atualização de dados previamente armazenados no banco, palavras chaves: **select, insert, uodate e delete**;

**Controle de acesso** – protege os dados de manipulação não autorizada;

**Compartilhamento de Dados** – coordena o compartilhamento dos dados por usuários concorrentes, sem conteúdo interferir na ação de cada um deles.

O diagrama abaixo resume as diferenças entre os ambientes:

Estação

Estação

INTERFACE COM USUARIO

INTERFACE COM USUARIO

MECANISMO DE ACESSO

Servidor

MECANISMO DE ACESSO

Servidor

ARMAZENAMENTO DE DADOS

ARMAZENAMENTO DE DADOS

## **Funções**

Como começamos por lidar com números, a próxima questão natural é perguntar se é possível efetuar cálculos com esses números, tais como somá-los ou calcular a sua média. A resposta é sim! A linguagem SQL tem várias funções aritméticas, sendo:

* **AVG**
* **COUNT**
* **MAX**
* **MIN**
* **SUM**
* **STDEV**
* **STDEVP**
* **VAR**
* **VARP.**

|  |  |
| --- | --- |
| Operador | Função |
| AVG | Média aritmética. |
| COUNT | Conta o numero de ocorrências de linhas. |
| MAX | Maior valor na coluna. |
| MIN | Menor valor na coluna. |
| SUM | Soma todos os valores da coluna. |
| STDEV | Desvio padrão estatístico na coluna. |
| STDEVP | Desvio padrão populacional na coluna. |
| VAR | Variação estatística dos valores da coluna. |
| VARP | Variação populacional dos valores da coluna. |

## **SQL Sintaxe**

Nesta secção, listamos a sintaxe SQL para cada um dos comandos SQL presentes nesta demonstração. Para obter explicações detalhadas sobre cada sintaxe SQL, consulte a secção individual conforme palavra-chave.

O objetivo é obter uma referência rápida acerca da sintaxe SQL. A sintaxe para utilizar funções é:

**SELECT "tipo de função" ("nome\_coluna")  
FROM "nome\_tabela";**

1. Criar Banco de Dados: Create Database

CREATE DATABASE Supermercado;

USE Supermercado;

1. Criar Tabelas: Create Table

CREATE TABLE Fornecedores (

Cod\_Fornec SMALLINT PRIMARY KEY,

Nome\_Fornec VARCHAR(50)

);

CREATE TABLE Produtos (

ID\_Prod SMALLINT PRIMARY KEY,

Nome\_Prod VARCHAR(50) NOT NULL,

Quant\_Prod VARCHAR(50) NOT NULL,

Preco\_Prod DECIMAL(10,2),

Cod\_Fornec SMALLINT,

CONSTRAINT fk\_cod\_fornec FOREIGN KEY (Cod\_Fornec)

REFERENCES Fornecedores(Cod\_Fornec)

);

1. Inserir Dados nas Tabelas : Insert Into

INSERT INTO Fornecedores (Cod\_Fornec, Nome\_Fornec)

VALUES (1,´Acme´),(2,´ggs´);

INSERT INTO Produtos

VALUES (100, ´Monitor LCD´, 12, 550.00, 1),

(101, ´Roteador Banda Larga´, 8, 130.00, 1),

(102, ´Teclado´, 15, 40.00, 1),

(103, ´PenDrive 64GB´, 22, 48.00, 1),

(104, ´Webcam´, 6, 170.00, 1);

1. Efetuar consulta trazendo colunas especificas

SELECT Nome\_Fornec FROM Fornecedores;

SELECT Nome\_Prod, Preco\_Prod FROM Produtos;

SELECT \* FROM Produtos;

1. Consultando uma tabela com filtro de dados especifico

SELECT Nome\_Prod FROM Produtos

WHERE ID\_Prod = 101;

1. Alterar um registro na Tabela :Update

UPDATE Produtos

SET Preco\_Prod = 630.00

WHERE ID\_Prod = 100;

SELECT Preco\_Prod FROM Produtos

WHERE ID\_Prod = 100;

1. Excluir registros de uma tabela : Delete From

DELETE FROM Produtos

WHERE ID\_Prod = 101;

Pode se consultar antes se tem o registro

1. Trazer dados de duas ou mais tabelas relacionados

SELECT Nome\_Prod, Nome\_Fornec

FROM Fornecedores

INNER Join Produtos

ON Fornecedores.Cod\_Fornec = Produtos.Cod\_Fornec;

1. Retornar resultados ordenados em uma consulta

SELECT Nome\_Prod, Preco\_Prod

FROM Produtos

ORDER BY Nome\_Prod;

1. Apagar todos os dados de um tabela de uma

TRUNCATE TABLE Produtos;

SELECT \* FROM Produtos;

CREATE TABLE Medicamentos

(ID\_Prod SMALLINT PRIMARY KEY,[Nome\_Prod] CHAR(50),

Quant\_Prod CHAR(50),

Preco\_Prod DECIMAL(10,2),

Cod\_Fornec SMALLINT);

INSERT INTO Medicamentos VALUES(1,'ass',12,550.00,1),

(101,'melhoral',8,130.00,1),

(102,'cebion',15,40.00,1),

(103,'xarope',22,48.00,1),

(104,'laxante',6,170.00,1);

SELECT \* FROM Medicamentos;