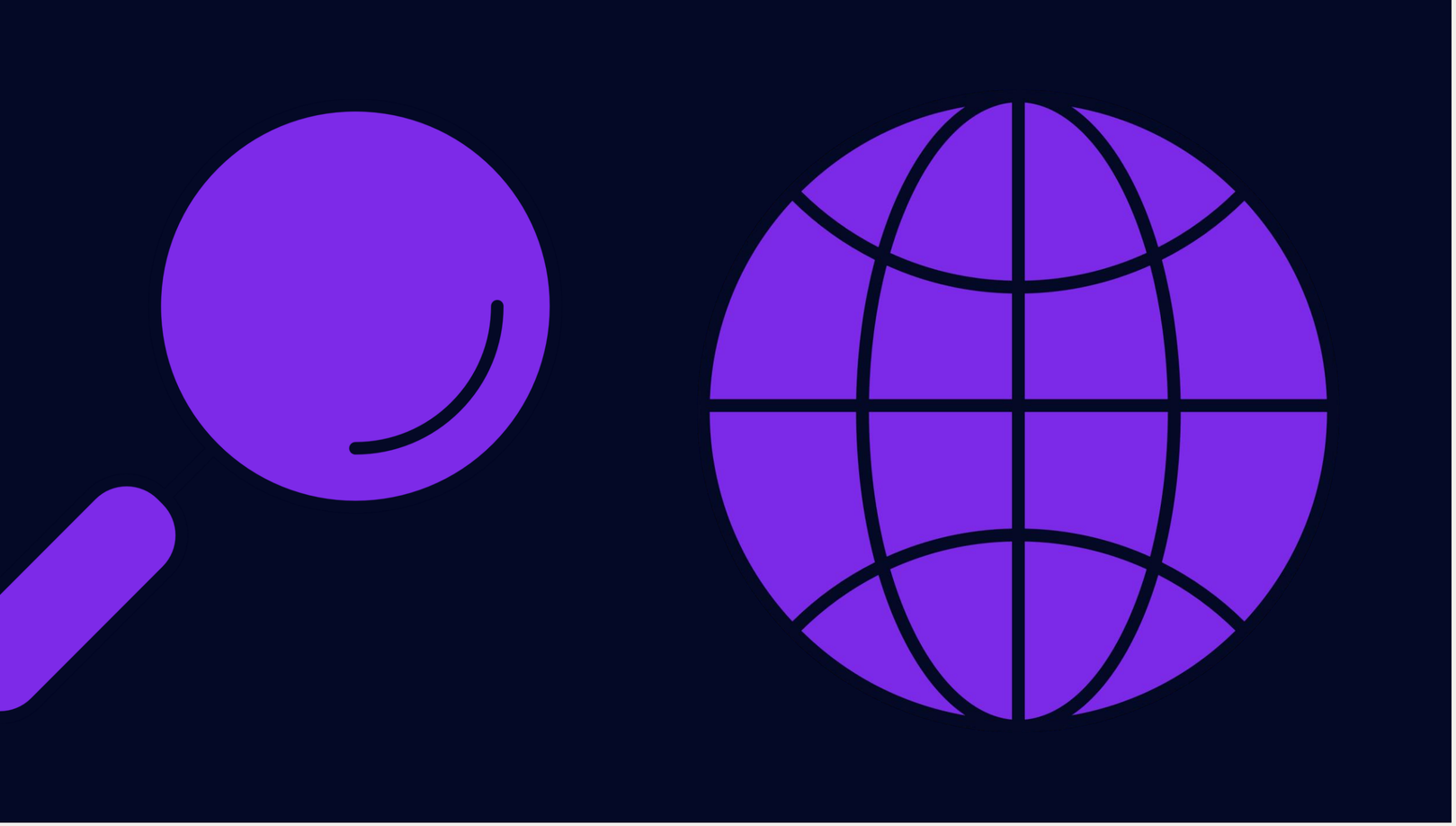


Projeto M.A Soluções

Banco de Dados Modelagem Relacional e Resoluções Práticas

Relembrando Definição de Dados

Linguagem de Definição de Dados (DDL, do inglês Data Definition Language) é um conjunto de comandos utilizados para definir a estrutura e organização de um banco de dados.
Eles permitem:

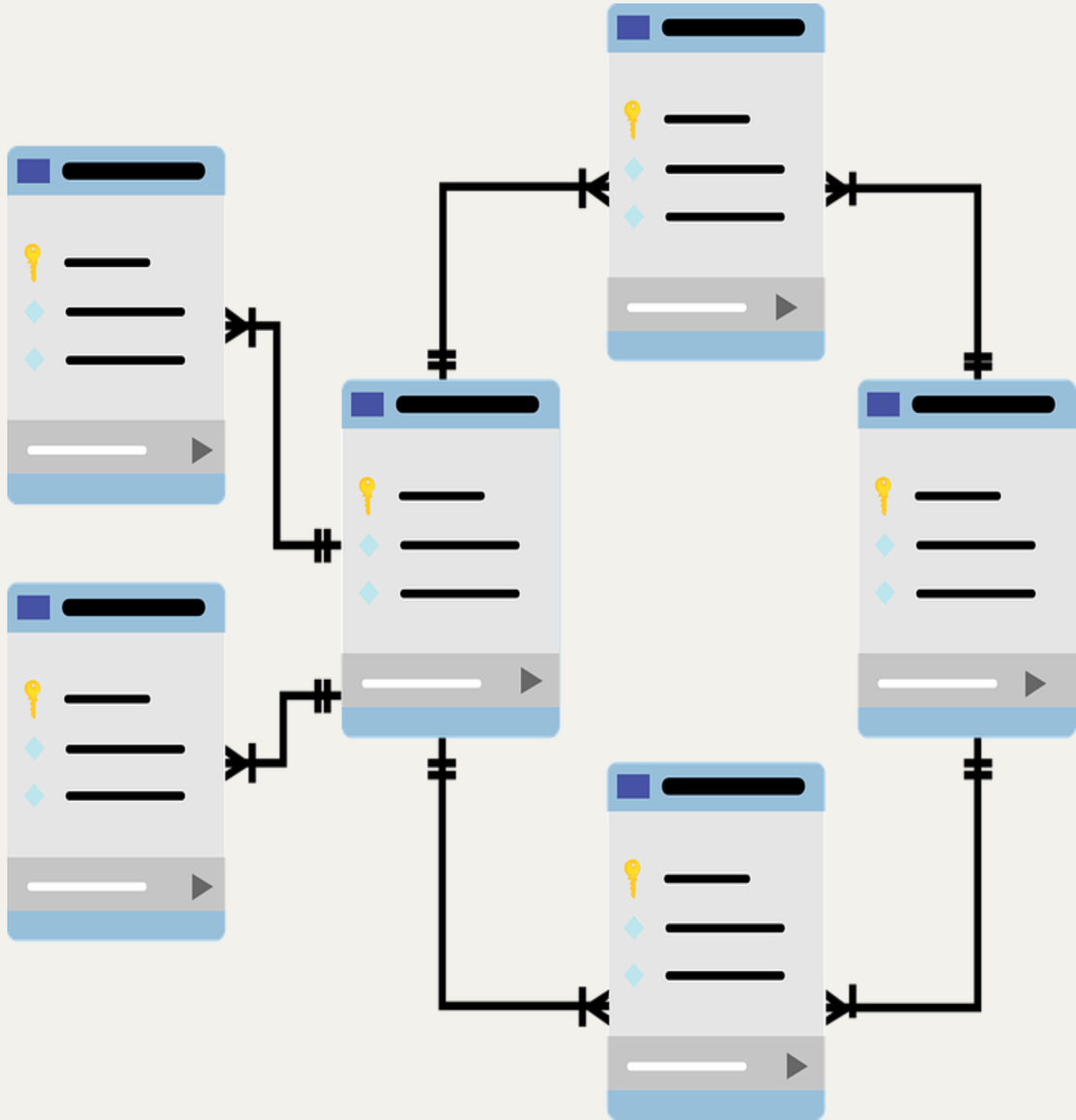
- A realização de consultas – DQL.
- A organização de dados - DDL.
- A manipulação de dados – DML

Exemplos:
DQL – “Select * FROM Clientes” | “Select Nome, Email FROM Clientes”.
DDL – “Create Table Clientes (IDCliente INT, Nome Varchar(50), Email Varchar(100));”
DML – “INSERT INTO Clientes (IDCliente, Nome, Email) VALUES (1, 'João', 'joao@email.com’);”

O que é Modelagem Relacional?

A modelagem relacional é uma maneira de organizar informações em um base de dados de forma parecida com tabelas. Cada linha tem uma informação, e cada coluna é um tipo de informação, como nome, números, datas e etc. Seus principais aspectos são:

- Tabelas: Estrutura
- Chaves Primárias e Estrangeiras: Identificadores únicos e relacionáveis, gerando um relacionamento.
- Normalização: Padrão de organização reduzindo redundâncias e promovendo eficiência.
- Flexibilidade e Escalabilidade: Permite uma prática flexível da manipulação dos dados e suporta expansão caso haja necessidade.



Constraints e suas funcionalidades

O que são? Como funcionam? São necessárias?

- Constraints são regras que você pode definir em uma tabela para controlar o tipo de dados que podem ser inseridos ou atualizados. Garante as integridade dos dados.

Os tipos de constraints:

- Primary Key: Garante que cada linha em uma tabela tenha uma identificação única.
- Foreign Key: Define um relacionamento entre duas tabelas, onde a chave estrangeira na tabela faz referência à chave primária de outra tabela já criada.
- Unique: Garante que os valores em uma coluna sejam únicos, mesmo que haja uma chave primária na tabela.
- Not Null: Garante que um campo não pode conter valores nulos.
- Default: Define um valor padrão para um campo, que será usado se nenhum valor for especificado durante a inserção de dados.

Exemplos de constraints

- Primary Key:

```sql
-- Primary Key (Chave Primária)
CREATE TABLE Clientes (
    IDCliente INT PRIMARY KEY,
    Nome VARCHAR(50)
);
```

- Unique e Not Null:

```sql
-- Unique Constraint - Unicidade
-- Not Null - Não Nulo
CREATE TABLE Produtos (
    IDProduto INT PRIMARY KEY,
    Nome VARCHAR(50)NOT NULL,
    Cpf Varchar(11) UNIQUE,
    CodigoProduto VARCHAR(10) UNIQUE
);
```

- Foreign Key:

```sql
-- Foreign Key (Chave Estrangeira)
CREATE TABLE Pedidos (
    IDPedido INT PRIMARY KEY,
    IDCliente INT,
    Produto VARCHAR(50),
    FOREIGN KEY (IDCliente) REFERENCES Clientes(IDCliente)
);
```

- Foreign Key:

```sql
-- Default Constraint - Padrão
CREATE TABLE Produtos (
    IDProduto INT PRIMARY KEY,
    Nome VARCHAR(50),
    Quantidade INT DEFAULT 0
);
```

Dicionário de Dados SQL

Antes de começarmos a prática, vamos relembrar as variáveis de dados no SQL.

- Cada tipo de dados tem características próprias, como o tamanho que pode ter, os números que pode representar e como é formatado.

- INTEGER/INT: Armazena números inteiros sem vírgula, pode armazenar números negativos.Exemplo: 1,10, -5,-8.
- FLOAT/FLOAT (P): Serve para armazenar números com casas decimais com vírgula, o “P” serve para informar quantos números ficarão depois da vírgula. Exemplo:Float (2) – 2 casas depois da vírgula.
- DECIMAL/NUMERIC (P, S): Semelhante ao Float, mas aqui você pode especificar quantas casas decimais terão antes e depois da vírgula. Exemplo: Decimal (4,2) – 4 números antes da vírgula e 2 depois da vírgula.
- CHAR (n): Serve para armazenar palavras ou frases curtas de tamanho fixo. O "n" diz qual é o tamanho máximo da palavra. Exemplo: “M”, “F”- (Masculino, Feminino).
- VARCHAR (n): Funciona igual a caixinha anterior, mas aqui o tamanho pode variar.
- TEXT: Armazena textos longos, sem limite específico.
- DATE: Armazena datas. Exemplo: 2023-10-10.
- TIME: Armazena horas, Exemplo: 14:30:00.
- DATETIME: Armazena data e hora, Exemplo: 2023-10-10 14:30:00.
- BOOLEAN/BOOL: Armazena valores verdadeiros (TRUE) ou falsos (FALSE).

- Observação: Para incluir comentários em seu código, use “- - “.

Vamos praticar?

- Nosso primeiro exemplo será de uma loja de produtos de informática
Vamos seguir uma lógica, que tabelas serão necessárias para nossa base de dados?

- Primeiro passo: Vamos iniciar criando nosso banco de dados:

```
-- Criar banco de dados
Create database loja_informatica;
```

- Caso você tenha outros bancos, usaremos o comando ***USE***.

```
-- Usar a database selecionada
use loja_informatica;
```

Vamos praticar?

- Segundo passo: Criar a tabela de Produtos:

```sql
--  Criar tabela de produtos
create table produtos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    valor DECIMAL(10,2) NOT NULL,
    filial VARCHAR(50) NOT NULL,
    quantidade_estoque INT NOT NULL,
    marca VARCHAR(20) NOT NULL
);
```

Vamos praticar?

➢Terceiro passo: Criar a tabela de Clientes:

```sql
-- Criar tabela de clientes
create table clientes(
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(50) NOT NULL,
    sobrenome VARCHAR(50) NOT NULL,
    email VARCHAR(200) NOT NULL,
    whatsapp VARCHAR(50) NOT NULL,
    estado VARCHAR(50) NOT NULL,
    telefone varchar(20) NOT NULL,
    sexo ENUM('M','F','I') NOT NULL
);
```

Observação: Em outros bancos de dados o tipo de dados pode mudar, como por exemplo, o ENUM no SQL Server não funciona, podemos usar o tipo de dado CHAR(1), ou seja, dado do tipo character que suporta somente 1 letra.

Vamos praticar?

➢ Quarto passo: Criar a tabela de venda, conectando com as tabelas de produtos e clientes :

```sql
-- Criar tabela de vendas realizadas
create table venda_realizada(
    id INT AUTO_INCREMENT PRIMARY KEY,
    data_venda DATE NOT NULL,
    quantidade INT NOT NULL,
    produto_id INT NOT NULL,
    cliente_id INT NOT NULL,
    FOREIGN KEY(produto_id) REFERENCES produtos(id),
    FOREIGN KEY(cliente_id) REFERENCES clientes(id)
);
```

Observação: Neste exemplo podemos perceber na ***constraint Foreign Key*** *que ela irá fazer uma referência as tabelas criadas anteriormente. Essa referência é um relacionamento entre as tabelas.*

Vamos praticar?

➢ Quinto passo: Inserindo os dados na tabela

Informe a sua tabela no ChatGPT e solicite os Inserts Into

```
create table produtos(
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(255) NOT NULL,
    valor DECIMAL(10,2) NOT NULL,
    filial VARCHAR(50) NOT NULL,
    quantidade_estoque INT NOT NULL,
    marca VARCHAR(20) NOT NULL
  );
```

Preciso de 20 inserts em seguida, desta tabela, que na marca seja da Razer ,que o nome do produto seja condizente a marca e que alterne entre a "Filial A" até "Filial D".

Observação: Para quem não possui um ERP, ou seja, um sistema de empresa que faça o cadastro automático em cada campo desejado, usaremos o ChatGPT para fazer o Insert desejado, de modo que evite trabalho manual excessivo

Se possível, a depender do seu gosto, forneça detalhes ao ChatGPT.
Observe o exemplo a esquerda.

Vamos praticar?

Insert da tabela selecionada.

```sql
-- Query para inserir os dados
INSERT INTO produtos (nome, valor, filial, quantidade_estoque, marca) VALUES
    ('Razer Mouse A', 49.99, 'Filial A', 100, 'Razer'),
    ('Razer Teclado B', 89.99, 'Filial B', 150, 'Razer'),
    ('Razer Fone de Ouvido C', 69.99, 'Filial C', 120, 'Razer'),
    ('Razer Tapete de Mouse D', 19.99, 'Filial D', 80, 'Razer'),
    ('Razer Monitor A', 299.99, 'Filial A', 50, 'Razer'),
    ('Razer Notebook B', 999.99, 'Filial B', 30, 'Razer'),
    ('Razer Mousepad C', 14.99, 'Filial C', 200, 'Razer'),
    ('Razer Headset D', 79.99, 'Filial D', 100, 'Razer'),
    ('Razer Cadeira Gamer A', 199.99, 'Filial A', 20, 'Razer'),
    ('Razer Mochila B', 49.99, 'Filial B', 80, 'Razer'),
    ('Razer Webcam C', 69.99, 'Filial C', 40, 'Razer'),
    ('Razer Caixa de Som D', 59.99, 'Filial D', 60, 'Razer'),
    ('Razer Mouse Gamer A', 69.99, 'Filial A', 70, 'Razer'),
    ('Razer Teclado Mecânico B', 119.99, 'Filial B', 100, 'Razer'),
    ('Razer Fone de Ouvido Sem Fio C', 129.99, 'Filial C', 90, 'Razer'),
    ('Razer Tapete de Mouse XL D', 24.99, 'Filial D', 120, 'Razer'),
    ('Razer Monitor UltraWide A', 399.99, 'Filial A', 30, 'Razer'),
    ('Razer Notebook Avançado B', 1299.99, 'Filial B', 20, 'Razer'),
    ('Razer Mouse Ergonômico C', 79.99, 'Filial C', 60, 'Razer'),
    ('Razer Cadeira Gamer Pro D', 249.99, 'Filial D', 10, 'Razer');
```

É notório que sabendo utilizar uma Inteligência Artificial para este propósito, irá reduzir bastante o trabalho manual e extenso. Agora faça o mesmo com a tabela de clientes e venda realizada.

Experimente usar após este insert gerado, o comando:
"Faça mais 20 Inserts".
Teste este comando, mas peça ao GPT para trocar a marca desejada, como Samsung por exemplo.

Continue fazendo os Inserts nas demais tabelas

```sql
-- Primeiro Insert Into de clientes
INSERT INTO clientes (nome, sobrenome, email, whatsapp, estado, telefone, sexo) VALUES
('Carlos', 'Silva', 'carlos.silva@gmail.com', '+5511987654321', 'SP', '(11)987654321', 'M'),
('Ana', 'Santos', 'ana.santos@gmail.com', '+5511976543210', 'RJ', '(21)976543210', 'F'),
('João', 'Oliveira', 'joao.oliveira@gmail.com', '+5511943210987', 'MG', '(31)943210987', 'M'),
('Maria', 'Pereira', 'maria.pereira@gmail.com', '+5511932109876', 'BA', '(71)932109876', 'F'),
('Luiz', 'Ferreira', 'luiz.ferreira@gmail.com', '+5511965432109', 'RS', '(51)965432109', 'M'),
('Camila', 'Gomes', 'camila.gomes@gmail.com', '+5511954321098', 'PE', '(81)954321098', 'F'),
('Paulo', 'Rocha', 'paulo.rocha@gmail.com', '+5511943210987', 'PR', '(41)943210987', 'M'),
('Laura', 'Santana', 'laura.santana@gmail.com', '+5511932109876', 'CE', '(85)932109876', 'F'),
('Felipe', 'Alves', 'felipe.alves@gmail.com', '+5511965432109', 'PA', '(91)965432109', 'M'),
('Amanda', 'Moraes', 'amanda.moraes@gmail.com', '+5511954321098', 'SC', '(48)954321098', 'F'),
('Gustavo', 'Pinto', 'gustavo.pinto@gmail.com', '+5511943210987', 'GO', '(62)943210987', 'M'),
('Mariana', 'Costa', 'mariana.costa@gmail.com', '+5511932109876', 'AM', '(92)932109876', 'F'),
('Rafael', 'Fernandes', 'rafael.fernandes@gmail.com', '+5511965432109', 'ES', '(27)965432109', 'M'),
('Bianca', 'Rodrigues', 'bianca.rodrigues@gmail.com', '+5511954321098', 'PB', '(83)954321098', 'F');
```

```sql
-- Primeiro Insert Into de Venda relizada
INSERT INTO venda_realizada (data_venda, quantidade, produto_id, cliente_id) VALUES
('2022-09-26', 5, 1, 1),
('2022-09-26', 3, 2, 2),
('2022-09-26', 2, 3, 3),
('2022-09-27', 4, 4, 4),
('2022-09-27', 1, 5, 5),
('2022-09-27', 7, 6, 6),
('2022-09-28', 3, 7, 7),
('2022-09-28', 6, 8, 8);
```

Informe ao GPT a sua tabela criada de clientes e venda realizada. Se possível, seja criterioso quanto ao tipo de informações que você deseja.

Exemplo: Faça 30 inserts da tabela clientes, colocando o número e o whatsapp com o mesmo DDD da UF da região informada.

Comandos Selects para sua tabela – Dicionário de Fórmulas

COUNT: Conta o número de registros em uma tabela ou o número de registros que atendem a uma condição específica.
SUM: Calcula a soma dos valores de uma coluna numérica.
AVG: Calcula a média dos valores de uma coluna numérica.
MAX: Retorna o valor máximo de uma coluna.
MIN: Retorna o valor mínimo de uma coluna.
WHERE: Filtra os registros com base em uma condição.

GROUP BY: Agrupa registros com base nos valores de uma ou mais colunas.
DISTINCT: Retorna valores distintos de uma coluna.
ORDER BY: Ordena os valores na coluna, seja de A até Z ou do maior valor até o menor valor e vice-versa. Usa o ***ASC*** e ***DESC*** para ordenar de modo Crescente e Decrescente respectivamente.

Junção de Tabelas - JOINS

O comando **JOIN** permite fazer consultas de modo relacionado com outras tabelas, permitindo não somente detalhes quanto aos dados, mas também um relacionamento.

Tipos de Joins

- **INNER JOIN:** Retorna registros que têm correspondência em ambas as tabelas.
- **LEFT JOIN:** Retorna todos os registros da tabela à esquerda (tabela A) e os registros correspondentes da tabela à direita (tabela B).
- **RIGHT JOIN:** Retorna todos os registros da tabela à direita (tabela B) e os registros correspondentes da tabela à esquerda (tabela A).

Obrigado!

Gabriel Florêncio

gabrielflorencioti@gmail.com