

Exercício 2

1. Definição de Banco de Dados:

Um banco de dados é uma coleção organizada de informações relacionadas entre si, que são armazenadas eletronicamente em um sistema de computador. Essas informações podem ser estruturadas, como tabelas em um banco de dados relacional, ou não estruturadas, como documentos em um banco de dados de texto completo. Um banco de dados permite o armazenamento eficiente, a recuperação, a manipulação e a análise dos dados, facilitando o gerenciamento de grandes volumes de informações.

2. Definição de SGBD:

SGBD é a sigla para Sistema de Gerenciamento de Banco de Dados. É um software responsável por gerenciar e controlar o acesso aos dados armazenados em um banco de dados. Um SGBD fornece uma interface para que os usuários possam interagir com o banco de dados, executar consultas, atualizar os dados e garantir a integridade e a segurança dos mesmos. Existem diversos sistemas de gerenciamento de banco de dados disponíveis, como Oracle, MySQL, SQL Server, PostgreSQL, entre outros.

Exemplos de SGBD:

- Oracle: É um sistema de gerenciamento de banco de dados relacional desenvolvido pela Oracle Corporation. Ele oferece uma ampla gama de recursos e é frequentemente utilizado em grandes empresas para aplicações críticas que requerem alta disponibilidade e desempenho.
- MySQL: É um SGBD relacional de código aberto amplamente utilizado. Ele é conhecido por sua facilidade de uso, escalabilidade e desempenho, sendo muito utilizado em aplicações web.
- SQL Server: Desenvolvido pela Microsoft, o SQL Server é um SGBD relacional com suporte a recursos avançados de segurança, análise de dados e integração com outras tecnologias Microsoft.
- PostgreSQL: É um SGBD relacional de código aberto conhecido por sua confiabilidade, conformidade com os padrões do setor e suporte a recursos avançados, como extensibilidade e replicação.

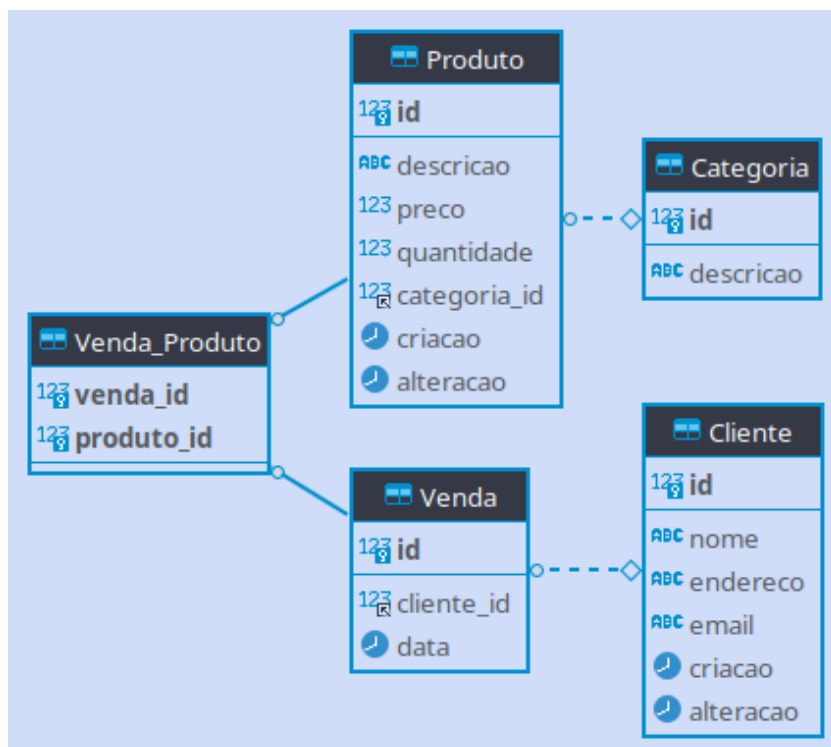
3. Definição de DDL e DML:

3.1 - DDL (Data Definition Language): É uma linguagem utilizada para definir a estrutura e o esquema de um banco de dados. Com o DDL, é possível criar tabelas, definir relacionamentos entre elas, especificar as restrições de integridade e realizar outras operações de definição de dados. Alguns exemplos de comandos DDL são CREATE, ALTER e DROP. Esses comandos permitem criar, modificar ou excluir objetos do banco de dados, como tabelas, índices, visões, entre outros.

3.2 - DML (Data Manipulation Language): É uma linguagem utilizada para manipular os dados armazenados em um banco de dados. Com o DML, é possível realizar operações como inserção, atualização, exclusão e recuperação de dados. Os comandos DML mais comuns são SELECT, INSERT, UPDATE e DELETE. Esses comandos permitem consultar dados, adicionar novos registros, atualizar valores existentes e excluir dados de uma tabela, por exemplo.

4. Faça o Diagrama Entidade-Relacionamentos do estudo de caso a seguir:

Uma empresa pretende vender produtos pela internet através de uma loja virtual. Portanto, está precisando de um sistema para gerenciar os produtos, as vendas e seus futuros clientes. Cada produto tem uma categoria, que possui: o id e a descrição. Deve-se fazer um cadastro de clientes contendo: id, nome, endereço, email e datas de criação e alteração das suas informações. Os produtos devem ser gerenciados pelo sistema com os dados: id, descrição, preço, quantidade, categoria e datas de criação e alteração das suas informações. As vendas são geradas entre clientes e produtos. Onde um cliente compra vários produtos e um tipo de produto pode ser comprado por inúmeros clientes.



5. Use os comandos DDL disponíveis no SQL para criação das tabelas da questão 5.

-- Criação da tabela Categoria

```
CREATE TABLE Categoria (  
    id INT PRIMARY KEY,  
    descricao VARCHAR(255)  
);
```

-- Criação da tabela Cliente

```
CREATE TABLE Cliente (  
    id INT PRIMARY KEY,  
    nome VARCHAR(255),  
    endereco VARCHAR(255),  
    email VARCHAR(255),  
    criacao DATETIME,  
    alteracao DATETIME  
);
```

-- Criação da tabela Produto

```
CREATE TABLE Produto (  
    id INT PRIMARY KEY,  
    descricao VARCHAR(255),  
    preco DECIMAL(10,2),  
    quantidade INT,  
    categoria_id INT,  
    criacao DATETIME,  
    alteracao DATETIME,  
    FOREIGN KEY (categoria_id) REFERENCES Categoria(id)  
);
```

-- Criação da tabela Venda

```
CREATE TABLE Venda (  
    id INT PRIMARY KEY,  
    cliente_id INT,  
    data DATETIME,  
    FOREIGN KEY (cliente_id) REFERENCES Cliente(id)  
);
```

-- Criação da tabela de associação entre Venda e Produto

```
CREATE TABLE Venda_Produto (  
    venda_id INT,  
    produto_id INT,  
    PRIMARY KEY (venda_id, produto_id),  
    FOREIGN KEY (venda_id) REFERENCES Venda(id),  
    FOREIGN KEY (produto_id) REFERENCES Produto(id)  
);
```

6. Use os comandos DML para “popular” e “explorar” as informações da base resultante da questão 6.

-- Populando a tabela Categoria:

```
INSERT INTO Categoria (id, descricao) VALUES
(1, 'Eletrônicos'),
(2, 'Brinquedos'),
(3, 'Móveis'),
(4, 'Livros'),
(5, 'Cosméticos'),
(6, 'Esportes'),
(7, 'Alimentos'),
```

-- Populando a tabela Cliente:

```
INSERT INTO Cliente (id, nome, endereco, email, criacao, alteracao) VALUES
(1, 'João Gabriel', 'Rua A, 123', 'joao@email.com', '2023-06-06', '2023-06-06'),
(2, 'Ana Maria', 'Av. B, 456', 'maria@email.com', '2023-06-06', '2023-06-06'),
(3, 'Pedro Oliveira', 'Rua C, 789', 'pedro@email.com', '2023-06-06', '2023-06-06'),
(4, 'Ana Souza', 'Av. D, 987', 'ana@email.com', '2023-06-06', '2023-06-06'),
(5, 'Carlos Ferreira', 'Rua E, 654', 'carlos@email.com', '2023-06-06', '2023-06-06'),
(6, 'Fernanda Oliveira', 'Rua F, 321', 'fernanda@email.com', '2023-06-07', '2023-06-07'),
(7, 'Rafael Costa', 'Av. G, 654', 'rafael@email.com', '2023-06-07', '2023-06-07');
```

-- Populando a tabela Produto:

```
INSERT INTO Produto (id, descricao, preco, quantidade, categoria_id, criacao, alteracao) VALUES
(1, 'Smartphone', 1500.00, 10, 1, '2023-06-06', '2023-06-06'),
(2, 'Camiseta', 29.90, 20, 6, '2023-06-06', '2023-06-06'),
(3, 'Relógio', 250.00, 5, 7, '2023-06-06', '2023-06-06'),
(4, 'Livro de Ficção', 49.90, 8, 4, '2023-06-06', '2023-06-06'),
(5, 'Perfume', 99.90, 15, 5, '2023-06-06', '2023-06-06'),
(6, 'Quebra-Cabeça', 39.90, 10, 2, '2023-06-07', '2023-06-07'),
(7, 'Sofá', 1200.00, 5, 3, '2023-06-07', '2023-06-07'),
(8, 'Bola de Futebol', 59.90, 15, 6, '2023-06-07', '2023-06-07');
```

-- Populando a tabela Venda:

```
INSERT INTO Venda (id, cliente_id, data) VALUES
(1, 1, '2023-06-06'),
(2, 2, '2023-06-06'),
(3, 3, '2023-06-06'),
(4, 4, '2023-06-06'),
(5, 5, '2023-06-06'),
(6, 6, '2023-06-07'),
(7, 7, '2023-06-07'),
(8, 1, '2023-06-07');
```

-- Populando a tabela Venda_Produto:

```
INSERT INTO Venda_Produto (venda_id, produto_id) VALUES
(1, 1),
(1, 2),
(2, 3),
(3, 4),
(4, 5),
(5, 6),
(6, 7),
(7, 8);
```