

Trabalho da Disciplina de Projeto de Bancos de Dados

Prof. Fred Lucena

## **PROJETO DE BANCO DE DADOS RELACIONAL**

Equipe:

Matheus Aquino dos Santos

---

Felipe Sabino de Oliveira

---

Yuri Jesus Vasconcelos

---

## **Objetivo**

O objetivo do projeto é desenvolver um banco de dados que seja capaz de aprimorar e organizar a logística de uma lanchonete, fornecendo dados de diferentes setores e áreas da loja, contribuindo para uma maior produtividade de seus funcionários e organização de demandas da lanchonete. Isso inclui o cadastro de clientes e funcionários, gestão de produtos no estoque, gerenciamento de pedidos, entre outras funcionalidades.

Em resumo, o projeto visa transformar a lanchonete em um negócio mais eficiente e organizado, por meio da criação do banco de dados, sendo capaz de fornecer dados de maneira simples aos usuários, adaptando-se às demandas dos clientes.

## **Modelo de Negócio**

O modelo de negócio da empresa é uma lanchonete que vende salgados. No qual produzimos e vendemos uma variedade de salgados, como pastéis, coxinhas, hambúrguer, entre outros lanches, para clientes que desejam lanches rápidos e saborosos. Nosso objetivo é lucrar vendendo esses produtos, proporcionando aos clientes uma opção rápida de alimentação. Nossa principal fonte de receita é a venda de salgados e bebidas.

Estamos no ramo alimentício, mais especificamente no setor de lanchonetes. Nosso mercado é o de alimentos prontos para consumo, que inclui lanches rápidos e salgados. Nossos clientes são estudantes, pessoas que fazem uma pausa para o lanche, famílias que desejam uma refeição rápida, entre outros tipos de clientes que temos na nossa lanchonete.

## Levantamento dos requisitos

- Produtos:

O sistema permite o cadastro de produtos(nome, descrição, categoria, preço).

Deve ser possível definir se um produto está disponível para venda(Situação: 'Em estoque' OR 'Esgotado').

- Pedidos:

Os clientes podem fazer pedidos, selecionando produtos e quantidades desejadas.

O sistema deve calcular automaticamente o valor total do pedido.

É necessário registrar a data, hora do pedido e endereço de entrega.

Os funcionários da lanchonete devem poder visualizar os pedidos em tempo real.

Devem atualizar o status dos pedidos, como "preparando", "saiu para entrega" e "entregue".

Possibilidade de cancelamento de pedidos. O sistema deve permitir o rastreamento de entregas em tempo real.

- Clientes:

Deve haver um cadastro de clientes com informações(nome, cpf, endereço, número de telefone).

Os clientes podem criar contas para visualizar o histórico de seus pedidos anteriores.

O cliente não é previamente cadastrado no sistema.

- Estoque:

O sistema deve controlar o estoque de produtos e ajustar o estoque automaticamente quando um pedido for feito.

Alertas devem ser enviados quando um produto estiver com estoque baixo.

- Fornecedor:

O sistema deve permitir o cadastro de informações sobre os fornecedores da lanchonete, incluindo nome da empresa, CNPJ, endereço, número de telefone, e-mail e site (se aplicável).

Deve ser possível categorizar os fornecedores com base em tipos de produtos fornecidos, como alimentos, bebidas.

O fornecedor é previamente cadastrado no sistema.

- Pagamentos:

Os clientes devem escolher entre várias opções de pagamento (cartão de crédito e débito, dinheiro, pagamento online, etc).

O sistema deve registrar informações de pagamento, como número de cartão (se aplicável) e valor.

- Relatórios:

O sistema deve gerar relatórios de vendas, mostrando informações como vendas diárias, semanais e mensais.

Relatórios de produtos mais vendidos e menos vendidos e relatórios de clientes com maior frequência de pedidos.

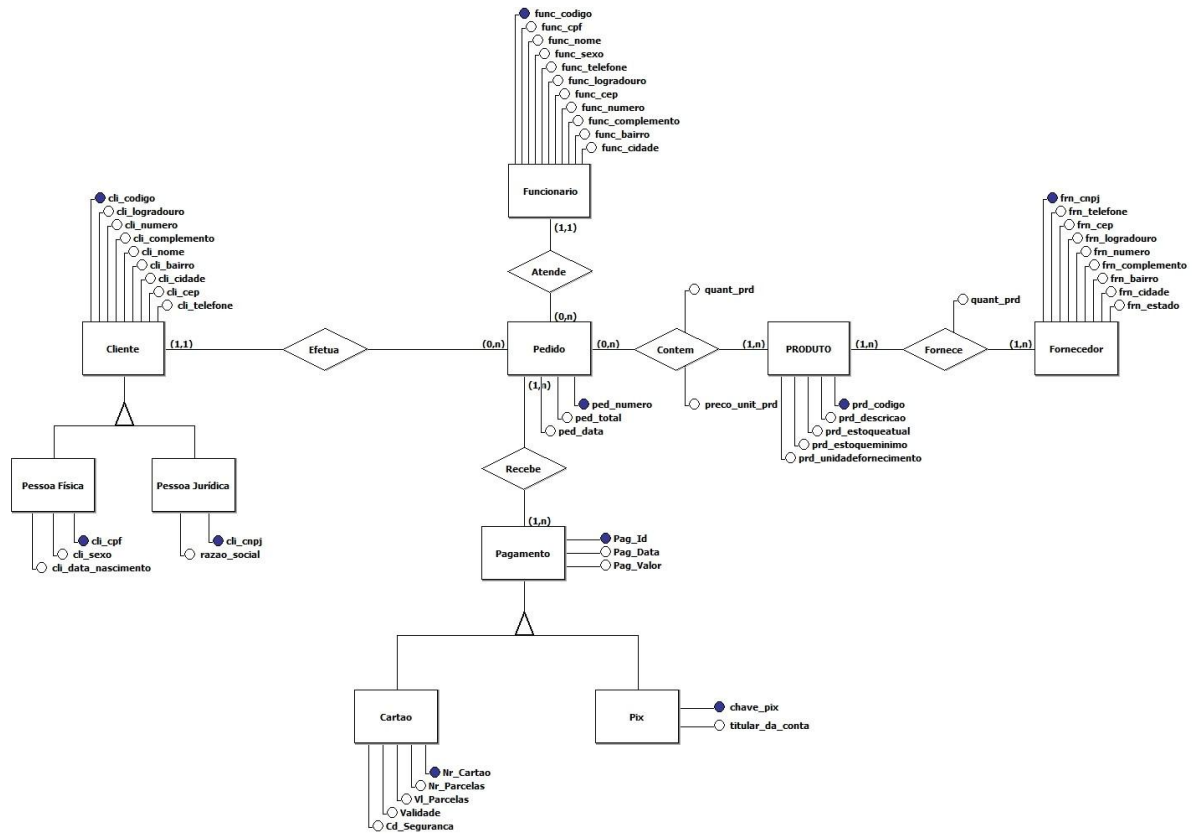
- Promoções:

O sistema deve suportar a criação e gestão de promoções e descontos temporários em produtos. A aplicação de descontos deve ser automática no momento do pedido.

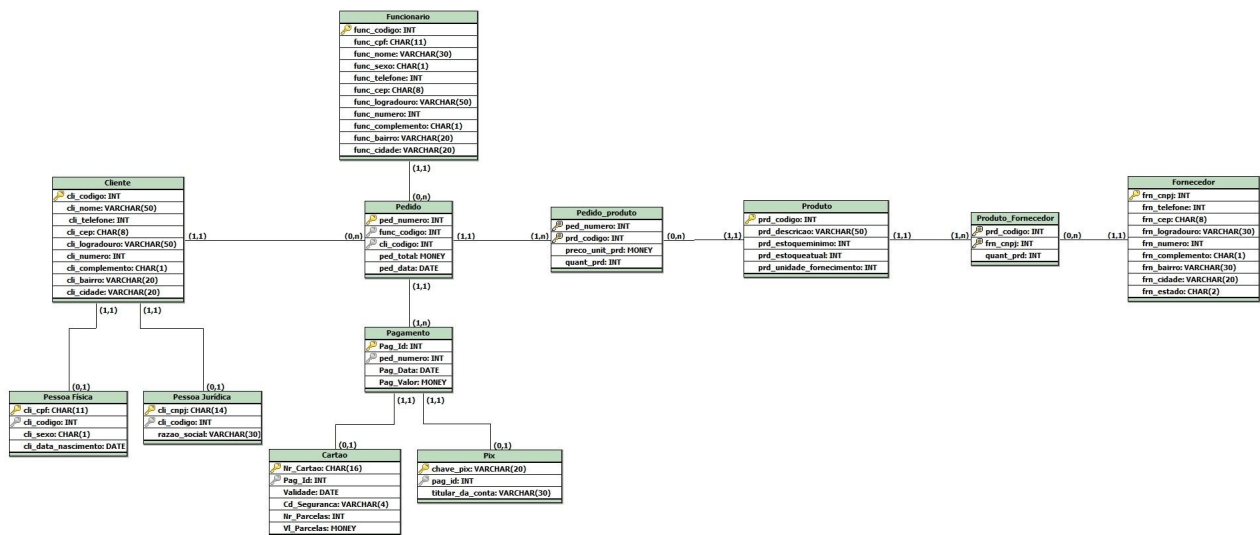
## **Escolha do SGBD**

Um dos motivos da escolha do SSMS(SQL Server Management Studio) foi a experiência possuída pelos integrantes do projeto. Além dessa afinidade para com esse SGBD, ele possui muitas qualidades, como: consultas SQL de maneira eficiente, a presença de uma interface gráfica intuitiva que facilita a criação, edição e gerenciamento de bancos de dados, a possibilidade de depuração de consultas SQL, o oferecimento de uma interface visual para criar e modificar tabelas e a permissão de gerenciamento da segurança do banco de dados, incluindo a configuração de permissões de usuário.

## Geração do Modelo Conceitual



## Geração do Modelo Lógico





```
CREATE TABLE Produto (prd_codigo          INTEGER PRIMARY KEY IDENTITY(1,1),
                        prd_nome            VARCHAR(50),
                        prd_unidadeFornecida INTEGER NOT NULL,
                        prd_estoqueminimo    INTEGER DEFAULT(0),
                        prd_estoqueatual     INTEGER DEFAULT(0));
```

```
CREATE TABLE Fornecedor (frn_cnpj         CHAR   (14) PRIMARY KEY,
                           frn_razao_social VARCHAR(80) NOT NULL,
                           frn_logradouro   VARCHAR(50),
                           frn_numero      INTEGER   ,
                           frn_complemento  CHAR    (1) ,
                           frn_bairro       VARCHAR(30),
                           frn_cidade       VARCHAR(30),
                           frn_estado       CHAR    (2) ,
                           frn_cep          CHAR    (8) ,
                           frn_telefone     CHAR    (11) NOT NULL);
```

```
CREATE TABLE Pedido_Produto (ped_numero    INTEGER,
                               prd_codigo    INTEGER,
                               preco_unit_prd MONEY NOT NULL,
                               quant_prd     INTEGER DEFAULT(0),
                               PRIMARY KEY    (ped_numero, prd_codigo),
                               FOREIGN KEY    (ped_numero) REFERENCES Pedido(ped_numero),
                               FOREIGN KEY    (prd_codigo) REFERENCES Produto(prd_codigo));
```

```
CREATE TABLE Produto_Fornecedor (prd_codigo    INTEGER,
                                    frn_cnpj       CHAR(14),
                                    frn_quantidade INTEGER,
                                    PRIMARY KEY     (prd_codigo, frn_cnpj),
                                    FOREIGN KEY     (prd_codigo) REFERENCES Produto(prd_codigo),
                                    FOREIGN KEY     (frn_cnpj)  REFERENCES Fornecedor(frn_cnpj));
```

```
CREATE TABLE Pagamento (Pag_Id          INTEGER PRIMARY KEY IDENTITY(1,1),
                           ped_Numero     INTEGER REFERENCES Pedido(ped_numero) NOT NULL,
                           pag_Data       DATE    NOT NULL,
                           pag_Valor      MONEY   NOT NULL);
```

```
CREATE TABLE Cartao (Nr_Cartao    CHAR(16) PRIMARY KEY NOT NULL,
                       Validade     DATE    NOT NULL,
                       Cd_Seguranca CHAR(4)  NOT NULL,
                       Nr_Parcelsas INTEGER  NOT NULL,
                       Vl_Parcelsas MONEY   NOT NULL,
                       Pag_Id        INTEGER REFERENCES Pagamento(Pag_Id) NOT NULL);
```



```
CREATE TABLE Pix (Chave_Pix VARCHAR(30) PRIMARY KEY NOT NULL,
Pag_Id INTEGER REFERENCES Pagamento(Pag_Id) NOT NULL);
```

/\* Alimentação de tabelas \*/

```
INSERT INTO Cliente ( cli_nome , cli_cidade , cli_bairro , cli_complemento, cli_logradouro , cli_numero, cli_cep , cli_telefone)
VALUES ('Felipe Sabino' , 'Recife' , 'Ibura' , 'A' , 'Rua Brejo da Cruz' , 74 , '32222222' , '81977774444'),
('Matheus Aquino' , 'Olinda' , 'Rio Doce' , 'D' , 'Rua Santa Ana' , 16 , '87498517' , '81955552222'),
('Yuri Jesus' , 'Recife' , 'Nova descoberta' , 'C' , 'Rua Alto do Jardim' , 810 , '45612479' , '81944441111'),
('ESPOSENDE CALCADOS' , 'Recife' , 'Casa Amarela' , 'A' , 'Rua Padre de Lemos' , 160 , '45679819' , '81933330000'),
('MAGAZINE LUIZA' , 'Recife' , 'Boa vista' , 'B' , 'Rua imperatriz Teresa Cristina' , 264 , '87679819' , '81966660000'),
('O Boticário' , 'Recife' , 'Santo Antônio' , 'A' , 'Rua Nova' , 10 , '45687719' , '81933331111'),
('Ana Clara' , 'Paulista' , 'Maranguape II' , 'A' , 'Rua das Estrelas' , 744 , '89722222' , '81977770000'),
('Maria Eliza' , 'Abreu e Lima' , 'Caetés II' , 'C' , 'Rua Sete' , 24 , '32222888' , '81988884444'),
('Pedro Nunes' , 'Recife' , 'Macaxeira' , 'E' , 'Rua Maria Amália' , 14 , '32222111' , '81955554444'),
('Carlos Oliveira' , 'Recife' , 'Corrégo Jenipapo' , 'D' , 'Rua Bento de Abreu' , 432 , '32222000' , '81944444444'),
('Camila Soares' , 'Recife' , 'Guabiraba' , 'E' , 'Rua Cassiterita' , 48 , '32222478' , '81933334444'),
('KINITOS FESTAS' , 'Recife' , 'Casa Amarela' , 'A' , 'Rua Padre de Lemos' , 160 , '04569819' , '81933332222'),
('JUNIOR MAGAZINE' , 'Recife' , 'Casa Amarela' , 'B' , 'Rua Padre de Lemos' , 264 , '87449819' , '81911110000');
```

```
INSERT INTO Pessoa_Fisica (cli_cpf , cli_codigo, cli_sexo, cli_data_nascimento)
VALUES (15497618311, 1000 , 'M' , '10-02-1980' ),
(89756324517, 1010 , 'M' , '01-10-1988' ),
(12315217497, 1020 , 'M' , '15-06-1982' ),
(1549765789, 1060 , 'F' , '10-12-2000' ),
(89756312346, 1070 , 'F' , '09-07-2003' ),
(78233217497, 1080 , 'M' , '07-06-2004' ),
(15541618311, 1090 , 'M' , '03-03-1989' ),
(89756874517, 1100 , 'F' , '02-05-2002' );
```

```
INSERT INTO Pessoa_Juridica (cli_cnpj , cli_codigo, cli_razao_social )
VALUES (89745419000112, 1030 , 'Esposende Calcados LTDA' ),
(79483512000129, 1040 , 'MAGAZINE LUIZA S/A' ),
(68784913222212, 1050 , 'O Boticário Franchising S/a' ),
(81111119000112, 1110 , 'KINITOS FESTAS LTDA' ),
(22223512000129, 1120 , 'JUNIOR MAGAZINE LTDA' );
```

```
INSERT INTO Funcionarios (func_nome , func_cpf , func_sexo, func_telefone, func_cidade, func_bairro , func_complemento, func_logradouro , func_numero, func_cep)
VALUES ('Maria Clara' , 84795483105, 'F' , 81978945613 , 'Olinda' , 'Ouro Preto' , 'A' , 'Rua Dracena' , '65' , 53370560),
('Rafael Alves' , 98756483105, 'M' , 81978948451 , 'Olinda' , 'Jardim Fragoso' , 'B' , 'Rua Santa Rita' , '685' , 58710060),
('Jéssica Silva' , 98756488102, 'F' , 81978948451 , 'Recife' , 'Iputinga' , 'A' , 'Rua São Mateus' , '175' , 87873060),
('Vanessa Lima' , 14785483105, 'F' , 81978949874 , 'Paulista' , 'Jardim Paulista' , 'C' , 'Rua Trinta e Seis' , '5' , 51110560),
('Rodrigo Barbosa' , 98745483105, 'M' , 81912348451 , 'Recife' , 'Passarinho' , 'B' , 'Rua Jandáia' , '15' , 58719999),
('Juliana Costa' , 36986488102, 'F' , 81978567451 , 'Recife' , 'Dois unidos' , 'F' , 'Rua do Campo' , '95' , 88883060);
```

```
INSERT INTO Pedido (ped_Data , func_codigo, cli_codigo, cli_descricao )
VALUES ('10-11-2023' , 1000 , 1000 , '2 Batata Frita' ),
('15-01-2023' , 1010 , 1010 , '2 Refrigerante Coca-Cola' ),
('08-09-2023' , 1020 , 1020 , '3 Sanduíche' ),
('29-11-23' , 1030 , 1030 , '50 Cachorro Quente' ),
('25-01-2023' , 1040 , 1040 , '45 Sorvete' ),
('08-10-2023' , 1050 , 1050 , '25 Hambúrguer com Cheddar' ),
('08-09-2023' , 1060 , 1060 , '3 Suco' ),
('30-11-2023' , 1010 , 1070 , '4 Bolo' ),
('25-01-2023' , 1020 , 1080 , '2 Pastel' ),
('08-10-2023' , 1030 , 1090 , '3 Coxinha' );
```

```
INSERT INTO Produto ( prd_nome , prd_unidadeFornecida, prd_estoqueminimo, prd_estoqueatual)
VALUES ('Batata Frita' , 130 , 0 , 60 ),
('Refrigerante Coca-Cola' , 200 , 0 , 60 ),
('Hambúrguer' , 110 , 0 , 60 ),
('Sanduíche' , 130 , 0 , 60 ),
('Cachorro Quente' , 150 , 0 , 60 ),
('Sorvete' , 100 , 0 , 60 ),
('Coxinha' , 200 , 0 , 60 ),
('Pastel' , 100 , 0 , 60 ),
('Suco' , 150 , 0 , 60 ),
('Bolo' , 50 , 0 , 60 );
```

```
INSERT INTO Fornecedor ( frn_cnpj , frn_razao_social , frn_logradouro , frn_numero , frn_complemento, frn_bairro , frn_cidade , frn_estado, frn_cep , frn_telefone)
VALUES ('89745419000150' , 'Suprema Paes e Massas Congeladas LTDA' , 'Av. Barreto de Menezes' , 90 , 'C' , 'Prazeres' , 'Jaboatão dos Guararapes' , 'PE' , '87987086' , 81978984535 ),
('81115419000112' , 'Coca Cola Indústrias Ltda' , 'Av. Clarice Marroquim' , 87 , 'A' , 'Boa Esperança' , 'Abreu e Lima' , 'PE' , '87951233' , 81921457895 ),
('78945419000112' , 'MASTERBOI LTDA' , 'Av. da Recuperação' , 98 , 'A' , 'Dois Irmãos' , 'Recife' , 'PE' , '87156423' , 81978645127 ),
('29997626000104' , 'CASA DE SORVETE INDUSTRIA E COMERCIO LTDA' , 'Rua Engenheiro Brandao Cavalcante' , 3287 , 'B' , 'Boa Viagem' , 'Recife' , 'PE' , '50751090' , 81999999895 ),
('78945419088888' , 'Qualyclean Distribuidora LTDA' , 'Av. Pinheiros' , 98 , 'C' , 'Imbiribeira' , 'Recife' , 'PE' , '14786423' , 81911145127 );
```

```
INSERT INTO Pedido_Produto (ped_numero, prd_codigo, preco_unit_prd, quant_prd)
VALUES ( 1 , 1 , 5.00 , 2 ),
( 2 , 2 , 4.00 , 1 ),
( 3 , 4 , 4.50 , 2 ),
( 4 , 5 , 3.00 , 50 ),
( 5 , 6 , 3.50 , 45 ),
( 6 , 3 , 15.00 , 25 ),
( 7 , 9 , 2.50 , 3 ),
( 8 , 10 , 3.50 , 4 ),
( 9 , 8 , 4.50 , 2 ),
( 10 , 6 , 2.50 , 3 );
```

```

INSERT INTO Produto_Fornecedor (prd_codigo,      frn_cnpj      , frn_quantidade)
VALUES ( 1 , '89745419000150' , 130 ),
( 2 , '81115419000112' , 200 ),
( 3 , '78945419000112' , 110 ),
( 4 , '89745419000150' , 130 ),
( 5 , '89745419000150' , 150 ),
( 6 , '29997626000104' , 100 ),
( 7 , '78945419088888' , 200 ),
( 8 , '78945419088888' , 100 ),
( 9 , '29997626000104' , 150 ),
( 10 , '89745419000150' , 50 );

```

```

INSERT INTO Pagamento (ped_Numero,      pag_Data      , pag_Valor)
VALUES ( 1 , '01-12-2023' , 10.00 ),
( 2 , '10-11-2023' , 4.00 ),
( 3 , '08-11-2023' , 9.00 ),
( 4 , '02-12-2023' , 150.00 ),
( 5 , '30-11-2023' , 157.50 ),
( 6 , '20-11-2023' , 45.00 ),
( 7 , '18-11-2023' , 7.50 ),
( 8 , '02-10-2023' , 14.00 ),
( 9 , '06-11-2023' , 7.00 ),
( 10 , '06-11-2023' , 7.50 );

```

```

INSERT INTO Cartao (Pag_Id,      Nr_Cartao      , Validade      , Cd_Seguranca, Nr_Parcels, Vl_Parcels)
VALUES ( 4 , 1057846879400741, '03-10-2027' , 9784 , 2 , 75.00 ),
( 5 , 1098745679456125, '02-11-2029' , 9154 , 3 , 52.50 ),
( 6 , 1057846879456125, '07-12-2025' , 4587 , 2 , 22.50 ),
( 8 , 7894531779456125, '01-11-2025' , 8147 , 2 , 12.00 );

```

```

INSERT INTO Pix (Pag_Id,      Chave_Pix      )
VALUES ( 1 , '12345678900 ' ),
( 2 , 'matheus@gmail.com' ),
( 3 , '81987562015' ),
( 7 , '12345789457' ),
( 9 , 'kinitos@gmail.com' ),
( 10 , '81914782015' );

```

```
/* Consultas de tabelas */
```

```
/* Alias */
```

```
SELECT cli_nome AS 'Nome do Cliente', cli_cidade AS Cidade  
FROM Cliente;
```

```
SELECT cli_cpf AS 'CPF do Cliente', cli_data_nascimento AS 'Data de Nascimento'  
FROM Pessoa_Fisica;
```

```
SELECT cli_cnpj AS 'CNPJ do Cliente', cli_codigo AS 'Codigo do Cliente', cli_razao_social AS 'Razao Social'  
FROM Pessoa_Juridica;
```

```
SELECT func_nome AS 'Nome do Funcionario', func_cidade AS 'Cidade'  
FROM Funcionarios;
```

```
SELECT ped_numero AS 'Numero do Pedido', ped_Data AS 'Data do Pedido'  
FROM Pedido;
```

```
SELECT prd_codigo AS 'Codigo do Produto', prd_nome AS 'Nome do Produto'  
FROM Produto;
```

```
SELECT frn_cnpj AS 'CNPJ do Fornecedor', frn_razao_social AS 'Razao Social'  
FROM Fornecedor;
```

```
SELECT ped_numero AS 'Numero do Pedido', prd_codigo AS 'Codigo do Produto', preco_unit_prd AS 'Preco Unitario'  
FROM Pedido_Produto;
```

```
SELECT prd_codigo AS 'Codigo do Produto', frn_cnpj AS 'CNPJ do Fornecedor', frn_quantidade AS 'Quantidade do pedido'  
FROM Produto_Fornecedor;
```

```
SELECT Pag_Id AS 'ID de Pagamento', ped_Numero AS 'Numero do Pedido', pag_Data AS 'Data do Pagamento', pag_Valor AS 'Valor do Pagamento'  
FROM Pagamento;
```

```
SELECT Nr_Cartao AS 'Numero do Cartao', Validade AS 'Validade do Cartao', Cd_Seguranca AS 'Codigo de Seguranca', Nr_Parcelsas AS 'Numero de Parcelas', Vl_Parcelsas AS 'Valor das Parcelas'  
FROM Cartao;
```

```
SELECT Chave_Pix AS 'Chave Pix', Pag_Id AS 'ID de Pagamento'  
FROM Pix;
```

---

```
/* Joins */
```

```
SELECT Cliente.cli_codigo,
       Cliente.cli_nome,
       Cliente.cli_cidade,
       Pessoa_Fisica.cli_cpf,
       Pessoa_Fisica.cli_sexo,
       Pessoa_Fisica.cli_data_nascimento
```

```
FROM Cliente
JOIN Pessoa_Fisica ON Cliente.cli_codigo = Pessoa_Fisica.cli_codigo;
```

```
SELECT Cliente.cli_codigo,
       Cliente.cli_nome,
       Cliente.cli_cidade,
       Cliente.cli_bairro,
       Cliente.cli_complemento,
       Cliente.cli_logradouro,
       Cliente.cli_numero,
       Cliente.cli_cep,
       Cliente.cli_telefone,
       Pessoa_Juridica.cli_cnpj,
       Pessoa_Juridica.cli_razao_social
```

```
FROM Cliente
JOIN Pessoa_Juridica ON Cliente.cli_codigo = Pessoa_Juridica.cli_codigo;
```

```
SELECT Produto.prđ_nome AS 'Nome do Produto', Fornecedor.frn_razao_social AS 'Razao Social', Produto_Fornecedor.frn_quantidade AS 'Quantidade Fornecida'
FROM Produto
JOIN Produto_Fornecedor ON Produto.prđ_codigo = Produto_Fornecedor.prđ_codigo
JOIN Fornecedor ON Produto_Fornecedor.frn_cnpj = Fornecedor.frn_cnpj;
```

```
/* Funções de Agregação */
```

```
SELECT SUM(pag_Valor) AS 'Valor Total de Vendas'
FROM Pagamento;
```

```
SELECT MIN(ped_Data) AS 'Data do Pedido Mais Antigo', MAX(ped_Data) AS 'Data do Pedido Mais Recente'
FROM Pedido;
```

```
SELECT AVG(preco_unit_prđ) AS 'Preco Medio dos Produtos'
FROM Pedido_Produto;
```

```
SELECT COUNT(cli_codigo) AS 'Total de Clientes'
FROM Cliente;
```

```
/* Trigger */
```

```
CREATE TRIGGER AtualizarEstoque
ON Pedido_Produto
FOR INSERT
AS
BEGIN
    UPDATE Produto
    SET prd_estoqueatual = prd_estoqueatual - inserir.quant_prd
    FROM Produto
    INNER JOIN inserted inserir ON Produto.prd_codigo = inserir.prd_codigo;
END;
```

```
select * from Produto;
```

```
/* Function */
```

```
CREATE FUNCTION CalcularTotalPagoCliente (@cliCodigo INTEGER)
RETURNS MONEY
AS
BEGIN
    DECLARE @totalPago MONEY;

    SELECT @totalPago = ISNULL(SUM(pag_Valor), 0)
    FROM Pagamento
    INNER JOIN Pedido ON Pagamento.ped_Numero = Pedido.ped_numero
    WHERE Pedido.cli_codigo = @cliCodigo;

    RETURN @totalPago;
END;
```

```
-- Exemplo de uso da function
```

```
DECLARE @codigoClienteExemplo INTEGER = 1030;
DECLARE @totalPagoCliente MONEY;

SET @totalPagoCliente = dbo.CalcularTotalPagoCliente(@codigoClienteExemplo);
SELECT @totalPagoCliente AS 'Total Pago pelo Cliente';
```

```
/* Stored Procedure */
```

```
CREATE PROCEDURE ListarPedidosPorCliente  
    @cliCodigo INTEGER
```

```
AS
```

```
BEGIN
```

```
    SELECT Pedido.ped_numero AS 'Numero do Pedido', Pedido.ped_Data AS 'Data do Pedido', Pedido.cli_descricao AS 'Descricao do Pedido'  
    FROM Pedido  
    WHERE Pedido.cli_codigo = @cliCodigo;
```

```
END;
```

```
EXEC ListarPedidosPorCliente @cliCodigo = 1010;
```

```
/* Index */
```

```
CREATE INDEX indice_cliente_nome ON Cliente(cli_nome)
```

```
select * from Cliente
```

Código Sql:

CREATE DATABASE SalgadosECia;

USE SalgadosECia;

/\* Criação de tabelas \*/

CREATE TABLE Cliente (cli\_codigo INTEGER PRIMARY KEY IDENTITY(1000,10),

cli\_nome VARCHAR(50) NOT NULL,

cli\_cidade VARCHAR(30) NOT NULL,

cli\_bairro VARCHAR(30) NOT NULL,

cli\_complemento CHAR(1),

cli\_logradouro VARCHAR(50) NOT NULL,

cli\_numero INTEGER NOT NULL,

cli\_cep CHAR(8) NOT NULL,

cli\_telefone CHAR(11) NOT NULL

);

CREATE TABLE Pessoa\_Fisica(cli\_cpf CHAR(11) PRIMARY KEY NOT NULL,

cli\_codigo INTEGER FOREIGN KEY  
REFERENCES Cliente(cli\_codigo) ON DELETE CASCADE,

cli\_sexo CHAR(1) CHECK(cli\_sexo =  
'M' OR cli\_sexo = 'F'),

cli\_data\_nascimento DATE NOT NULL

);

CREATE TABLE Pessoa\_Juridica (cli\_cnpj CHAR(14) PRIMARY KEY NOT NULL,

```
cli_codigo    INTEGER FOREIGN KEY REFERENCES Cliente(cli_codigo) ON DELETE
CASCADE,

cli_razao_social VARCHAR(80) NOT NULL

);
```

```
CREATE TABLE Funcionarios (func_codigo    INTEGER PRIMARY KEY IDENTITY(1000,10),

    func_nome    VARCHAR(30) NOT NULL,

    func_cpf      CHAR(11) NOT NULL UNIQUE,

    func_sexo    CHAR(1) CHECK(func_sexo = 'M' OR func_sexo = 'F'),

    func_telefone VARCHAR(11) NOT NULL,

    func_cidade  VARCHAR(30) NOT NULL,

    func_bairro  VARCHAR(20) NOT NULL,

    func_complemento CHAR(1),

    func_logradouro VARCHAR(50) NOT NULL,

    func_numero  INTEGER NOT NULL,

    func_cep      CHAR(8) NOT NULL

);
```

```
CREATE TABLE Pedido (ped_numero  INTEGER PRIMARY KEY IDENTITY(1,1),

    ped_Data    DATE NOT NULL,

    func_codigo INTEGER FOREIGN KEY REFERENCES Funcionarios(func_codigo),

    cli_codigo  INTEGER FOREIGN KEY REFERENCES Cliente(cli_codigo),

    cli_descricao VARCHAR(100) NOT NULL

);
```





```
PRIMARY KEY (ped_numero, prd_codigo),  
FOREIGN KEY (ped_numero) REFERENCES Pedido(ped_numero),  
FOREIGN KEY (prd_codigo) REFERENCES Produto(prd_codigo)  
);
```

```
CREATE TABLE Produto_Fornecedor (prd_codigo INTEGER,  
                                   frn_cnpj CHAR(14),  
                                   frn_quantidade INTEGER,  
                                   PRIMARY KEY (prd_codigo, frn_cnpj),  
                                   FOREIGN KEY (prd_codigo) REFERENCES Produto(prd_codigo),  
                                   FOREIGN KEY (frn_cnpj) REFERENCES Fornecedor(frncnpj)  
);
```

```
CREATE TABLE Pagamento (Pag_Id INTEGER PRIMARY KEY IDENTITY(1,1),  
                           ped_Numero INTEGER REFERENCES Pedido(ped_numero) NOT NULL,  
                           pag_Data DATE NOT NULL,  
                           pag_Valor MONEY NOT NULL  
);
```

```
CREATE TABLE Cartao (Nr_Cartao CHAR(16) PRIMARY KEY NOT NULL,  
                      Validade DATE NOT NULL,  
                      Cd_Seguranca CHAR(4) NOT NULL,  
                      Nr_Parcels INTEGER NOT NULL,
```

```
VI_Parcels MONEY NOT NULL,  
Pag_Id INTEGER REFERENCES Pagamento(Pag_Id) NOT NULL  
);
```

```
CREATE TABLE Pix (Chave_Pix VARCHAR(30) PRIMARY KEY NOT NULL,  
Pag_Id INTEGER REFERENCES Pagamento(Pag_Id) NOT NULL  
);
```

```
/* Alimentação de tabelas */
```

```
INSERT INTO Cliente (cli_nome, cli_cidade, cli_bairro, cli_complemento, cli_logradouro, cli_numero, cli_cep,  
cli_telefone)  
VALUES ('Felipe Sabino', 'Recife', 'Ibura', 'A', 'Rua Brejo da Cruz', 74, '32222222',  
'81977774444'),  
( 'Matheus Aquino', 'Olinda', 'Rio Doce', 'D', 'Rua Santa Ana', 16, '87498517',  
'81955552222'),  
( 'Yuri Jesus', 'Recife', 'Nova descoberta', 'C', 'Rua Alto do Jardim', 810, '45612479', '81944441111'),  
( 'ESPOSENDE CALCADOS', 'Recife', 'Casa Amarela', 'A', 'Rua Padre de Lemos', 160, '45679819',  
'81933330000'),  
( 'MAGAZINE LUIZA', 'Recife', 'Boa vista', 'B', 'Rua imperatriz Teresa Cristina', 264, '87679819',  
'81966660000'),  
( 'O Boticário', 'Recife', 'Santo Antônio', 'A', 'Rua Nova', 10, '45687719', '81933331111'),  
( 'Ana Clara', 'Paulista', 'Maranguape II', 'A', 'Rua das Estrelas', 744, '89722222',  
'81977770000'),
```

```

('Maria Eliza', 'Abreu e Lima', 'Caetés II', 'C', 'Rua Sete', 24, '32222888',
'81988884444'),

('Pedro Nunes', 'Recife', 'Macaxeira', 'E', 'Rua Maria Amália', 14, '32222111',
'81955554444'),

('Carlos Oliveira', 'Recife', 'Corrégo Jenipapo', 'D', 'Rua Bento de Abreu', 432,
'32222000', '81944444444'),

('Camila Soares', 'Recife', 'Guabiraba', 'E', 'Rua Cassiterita', 48, '32222478',
'81933334444'),

('KINITOS FESTAS', 'Recife', 'Casa Amarela', 'A', 'Rua Padre de Lemos', 160,
'04569819', '81933332222'),

('JUNIOR MAGAZINE', 'Recife', 'Casa Amarela', 'B', 'Rua Padre de Lemos', 264, '87449819',
'81911110000');

```

```

INSERT INTO Pessoa_Fisica (cli_cpf, cli_codigo, cli_sexo, cli_data_nascimento)

VALUES (15497618311, 1000, 'M', '10-02-1980'),

(89756324517, 1010, 'M', '01-10-1988'),

(12315217497, 1020, 'M', '15-06-1982'),

(15497655789, 1060, 'F', '10-12-2000'),

(89756312346, 1070, 'F', '09-07-2003'),

(78233217497, 1080, 'M', '07-06-2004'),

(15541618311, 1090, 'M', '03-03-1989'),

(89756874517, 1100, 'F', '02-05-2002');

```

```

INSERT INTO Pessoa_Juridica (cli_cnpj, cli_codigo, cli_razao_social)

VALUES (89745419000112, 1030, 'Esposende Calçados LTDA'),

(79483512000129, 1040, 'MAGAZINE LUIZA S/A'),

(68784913222212, 1050, 'o Boticário Franchising S/a'),

```

(81111119000112, 1110, 'KINITOS FESTAS LTDA'),

(22223512000129, 1120, 'JUNIOR MAGAZINE LTDA');

INSERT INTO Funcionarios (func\_nome, func\_cpf, func\_sexo, func\_telefone, func\_cidade, func\_bairro, func\_complemento, func\_logradouro, func\_numero, func\_cep)

VALUES ('Maria Clara', 84795483105, 'F', 81978945613, 'Olinda', 'Ouro Preto', 'A', 'Rua Dracena', '65', 53370560),

('Rafael Alves', 98756483105, 'M', 81978948451, 'Olinda', 'Jardim Fragoso', 'B', 'Rua Santa Rita', '685', 58710060),

('Jéssica Silva', 98756488102, 'F', 81978948451, 'Recife', 'Iputinga', 'A', 'Rua São Mateus', '175', 87873060),

('Vanessa Lima', 14785483105, 'F', 81978949874, 'Paulista', 'Jardim Paulista', 'C', 'Rua Trinta e Seis', '5', 51110560),

('Rodrigo Barbosa', 98745483105, 'M', 81912348451, 'Recife', 'Passarinho', 'B', 'Rua Jandáia', '15', 58719999),

('Juliana Costa', 36986488102, 'F', 81978567451, 'Recife', 'Dois unidos', 'F', 'Rua do Campo', '95', 88883060);

INSERT INTO Pedido (ped\_Data, func\_codigo, cli\_codigo, cli\_descricao)

VALUES ('10-11-2023', 1000, 1000, '2 Batata Frita'),

('15-01-2023', 1010, 1010, '2 Refrigerante Coca-Cola'),

('08-09-2023', 1020, 1020, '3 Sanduíche'),

('29-11-23', 1030, 1030, '50 Cachorro Quente'),

('25-01-2023', 1040, 1040, '45 Sorvete'),

('08-10-2023', 1050, 1050, '25 Hambúrguer com Cheddar'),

('08-09-2023', 1000, 1060, '3 Suco'),

('30-11-2023', 1010, 1070, '4 Bolo'),

('25-01-2023', 1020, 1080, '2 Pastel' ),  
('08-10-2023', 1030, 1090, '3 Coxinha');

INSERT INTO Produto (prd\_nome, prd\_unidadeFornecida, prd\_estoqueminimo, prd\_estoqueatual)

VALUES ('Batata Frita', 130, 0, 60),  
('Refrigerante Coca-Cola', 200, 0, 60),  
('Hambúrguer', 110, 0, 60),  
('Sanduíche', 130, 0, 60),  
('Cachorro Quente', 150, 0, 60),  
('Sorvete', 100, 0, 60),  
('Coxinha', 200, 0, 60),  
('Pastel', 100, 0, 60),  
('Suco', 150, 0, 60),  
('Bolo', 50, 0, 60);

INSERT INTO Fornecedor (frn\_cnpj, frn\_razao\_social, frn\_logradouro, frn\_numero, frn\_complemento, frn\_bairro, frn\_cidade, frn\_estado, frn\_cep, frn\_telefone)

VALUES ('89745419000150', 'Suprema Paes e Massas Congeladas LTDA', 'Av. Barreto de Menezes', 90, 'C', 'Prazeres', 'Jaboatão dos Guararapes', 'PE', '87987086', 81978984535),

('81115419000112', 'Coca Cola Indústrias Ltda', 'Av. Clarice Marroquim', 87, 'A', 'Boa Esperança', 'Abreu e Lima', 'PE', '87951233', 81921457895),

('78945419000112', 'MASTERBOI LTDA', 'Av. da Recuperação', 98, 'A', 'Dois Irmãos', 'Recife', 'PE', '87156423', 81978645127),

('29997626000104', 'CASA DE SORVETE INDUSTRIA E COMERCIO LTDA', 'Rua Engenheiro Brandao Cavalcante', 3287, 'B', 'Boa Viagem', 'Recife', 'PE', '50751090', 81999999895),

```
('78945419088888', 'Qualyclean Distribuidora LTDA', 'Av. Pinheiros', 98, 'C', 'Imbiribeira',  
'Recife', 'PE', '14786423', 81911145127);
```

```
INSERT INTO Pedido_Produto (ped_numero, prd_codigo, preco_unit_prd, quant_prd)
```

```
VALUES (1, 1, 5.00, 2),  
  
(2, 2, 4.00, 1),  
  
(3, 4, 4.50, 2),  
  
(4, 5, 3.00, 50),  
  
(5, 6, 3.50, 45),  
  
(6, 3, 15.00, 25),  
  
(7, 9, 2.50, 3),  
  
(8, 10, 3.50, 4),  
  
(9, 8, 4.50, 2),  
  
(10, 6, 2.50, 3);
```

```
INSERT INTO Produto_Fornecedor (prd_codigo, frn_cnpj, frn_quantidade)
```

```
VALUES (1, '89745419000150', 130),  
  
(2, '81115419000112', 200),  
  
(3, '78945419000112', 110),  
  
(4, '89745419000150', 130),  
  
(5, '89745419000150', 150),  
  
(6, '29997626000104', 100),  
  
(7, '78945419088888', 200),  
  
(8, '78945419088888', 100),  
  
(9, '29997626000104', 150),
```

(10, '89745419000150', 50);

INSERT INTO Pagamento (ped\_Numero, pag\_Data, pag\_Valor)

VALUES (1, '01-12-2023', 10.00),  
(2, '10-11-2023', 4.00),  
(3, '08-11-2023', 9.00),  
(4, '02-12-2023', 150.00),  
(5, '30-11-2023', 157.50),  
(6, '20-11-2023', 45.00),  
(7, '18-11-2023', 7.50),  
(8, '02-10-2023', 14.00),  
(9, '06-11-2023', 7.00),  
(10, '06-11-2023', 7.50);

INSERT INTO Cartao (Pag\_Id, Nr\_Cartao, Validade, Cd\_Seguranca, Nr\_Parcels, Vl\_Parcels)

VALUES (4, 1057846879400741, '03-10-2027', 9784, 2, 75.00),  
(5, 1098745679456125, '02-11-2029', 9154, 3, 52.50),  
(6, 1057846879456125, '07-12-2025', 4587, 2, 22.50),  
(8, 7894531779456125, '01-11-2025', 8147, 2, 12.00);

INSERT INTO Pix (Pag\_Id, Chave\_Pix)

VALUES (1, '12345678900'),  
(2, 'matheus@gmail.com'),



```
(3, '81987562015'),  
  
      (7, '12345789457'),  
  
      (9, 'kinitos@gmail.com'),  
  
      (10, '81914782015');
```

```
/* Consultas de tabelas */
```

```
/* Alias */
```

```
SELECT cli_nome AS 'Nome do Cliente', cli_cidade AS Cidade, cli_bairro AS 'Bairro do Cliente'  
  
FROM Cliente;
```

```
SELECT cli_cpf AS 'CPF do Cliente', cli_data_nascimento AS 'Data de Nascimento'  
  
FROM Pessoa_Fisica;
```

```
SELECT cli_cnpj AS 'CNPJ do Cliente', cli_codigo AS 'Codigo do Cliente', cli_razao_social AS 'Razao Social'  
  
FROM Pessoa_Juridica;
```

```
SELECT func_nome AS 'Nome do Funcionario', func_cidade AS 'Cidade'  
  
FROM Funcionarios;
```

```
SELECT ped_numero AS 'Numero do Pedido', ped_Data AS 'Data do Pedido'

FROM Pedido;
```

```
SELECT prd_codigo AS 'Codigo do Produto', prd_nome AS 'Nome do Produto'

FROM Produto;
```

```
SELECT frn_cnpj AS 'CNPJ do Fornecedor', frn_razao_social AS 'Razao Social'

FROM Fornecedor;
```

```
SELECT ped_numero AS 'Numero do Pedido', prd_codigo AS 'Codigo do Produto', preco_unit_prd AS 'Preco
Unitario'

FROM Pedido_Produto;
```

```
SELECT prd_codigo AS 'Codigo do Produto', frn_cnpj AS 'CNPJ do Fornecedor', frn_quantidade AS 'Quantidade
do pedido'

FROM Produto_Fornecedor;
```

```
SELECT Pag_Id AS 'ID de Pagamento', ped_Numero AS 'Numero do Pedido', pag_Data AS 'Data do Pagamento',
pag_Valor AS 'Valor do Pagamento'

FROM Pagamento;
```

```
SELECT Nr_Cartao AS 'Numero do Cartao', Validade AS 'Validade do Cartao', Cd_Seguranca AS 'Codigo de Seguranca', Nr_Parcels AS 'Numero de Parcelas', Vl_Parcels AS 'Valor das Parcelas'

FROM Cartao;
```

```
SELECT Chave_Pix AS 'Chave Pix', Pag_Id AS 'ID de Pagamento'

FROM Pix;
```

```
/* Joins */
```

```
SELECT Cliente.cli_codigo,

       Cliente.cli_nome,

       Cliente.cli_cidade,

       Pessoa_Fisica.cli_cpf,

       Pessoa_Fisica.cli_sexo,

       Pessoa_Fisica.cli_data_nascimento

FROM Cliente

JOIN Pessoa_Fisica ON Cliente.cli_codigo = Pessoa_Fisica.cli_codigo;
```

```
SELECT Cliente.cli_codigo,

       Cliente.cli_nome,

       Cliente.cli_cidade,

       Cliente.cli_bairro,

       Cliente.cli_complemento,

       Cliente.cli_logradouro,
```

Cliente.cli\_numero,

Cliente.cli\_cep,

Cliente.cli\_telefone,

Pessoa\_Juridica.cli\_cnpj,

Pessoa\_Juridica.cli\_razao\_social

FROM Cliente

JOIN Pessoa\_Juridica ON Cliente.cli\_codigo = Pessoa\_Juridica.cli\_codigo;

/\* Funções de Agregação \*/

SELECT SUM(pag\_Valor) AS 'Valor Total de Vendas'

FROM Pagamento;

SELECT MIN(ped\_Data) AS 'Data do Pedido Mais Antigo', MAX(ped\_Data) AS 'Data do Pedido Mais Recente'

FROM Pedido;

SELECT AVG(preco\_unit\_prd) AS 'Preco Medio dos Produtos'

FROM Pedido\_Produto;

SELECT COUNT(cli\_codigo) AS 'Total de Clientes'

FROM Cliente;

/\* Trigger \*/

CREATE TRIGGER AtualizarEstoque

ON Pedido\_Produto

AFTER INSERT

AS

BEGIN

    UPDATE Produto

    SET prd\_estoqueatual = prd\_estoqueatual - inserir.quant\_prd

    FROM Produto

    INNER JOIN inserted inserir ON Produto.prd\_codigo = inserir.prd\_codigo;

END;

select \* from Produto;

/\* Function \*/

CREATE FUNCTION CalcularTotalPagoCliente (@cliCodigo INTEGER)

RETURNS MONEY

AS

BEGIN

    DECLARE @totalPago MONEY;

```

SELECT @totalPago = ISNULL(SUM(pag_Valor), 0)

FROM Pagamento

INNER JOIN Pedido ON Pagamento.ped_Numero = Pedido.ped_numero

WHERE Pedido.cli_codigo = @cliCodigo;

RETURN @totalPago;

END;

```

-- Exemplo de uso da function

```

DECLARE @codigoClienteExemplo INTEGER = 1030;

DECLARE @totalPagoCliente MONEY;

SET @totalPagoCliente = dbo.CalcularTotalPagoCliente(@codigoClienteExemplo);

SELECT @totalPagoCliente AS 'Total Pago pelo Cliente';

```

/\* Stored Procedure \*/

```

CREATE PROCEDURE ListarPedidosPorCliente

```

```

    @cliCodigo INTEGER

```

```

AS

```

```

BEGIN

```

```

    SELECT Pedido.ped_numero AS 'Numero do Pedido', Pedido.ped_Data AS 'Data do Pedido',
    Pedido.cli_descricao AS 'Descricao do Pedido'

```

FROM Pedido

WHERE Pedido.cli\_codigo = @cliCodigo;

END;

EXEC ListarPedidosPorCliente @cliCodigo = 1010;

/\* Index \*/

CREATE INDEX indice\_cliente\_nome ON Cliente(cli\_nome)

select \* from Cliente