

--create database test5;

--use test5;

-- Criação das Tabelas

-- Utilizando a AUTONUMERAÇÃO para a tabela Funcionario

CREATE TABLE Funcionario

(

Id_Funcionario INT IDENTITY(1000, 10) PRIMARY KEY,

Nome_Funcionario VARCHAR(50) NOT NULL,

CPF_Funcionario CHAR(11) NOT NULL UNIQUE,

Salario_Funcionario Money NOT NULL,

Cargo_Funcionario VARCHAR(30) NOT NULL

);

-- Utilizando FOREIGN KEY com exclusão em cascata

CREATE TABLE Ponto

(

```
    NumRegistro INT IDENTITY(1000, 10) PRIMARY KEY,  
    DataRegistro DATE NOT NULL,  
    IdFuncionario INT NOT NULL REFERENCES Funcionario(Id_Funcionario) ON  
DELETE CASCADE  
);
```

```
CREATE TABLE Cliente
```

```
(  
  
    CPFCliente CHAR(11) PRIMARY KEY,  
    NomeCliente VARCHAR(50) NOT NULL,  
    CEP_Cliente CHAR(8) NOT NULL,  
    Cidade_Cliente VARCHAR(30) NOT NULL,  
    Estado_Cliente VARCHAR(20) NOT NULL,  
    Numero_Cliente CHAR(9) NOT NULL  
  
);
```

```
-- Utilizando a restrição CHECK
```

```
CREATE TABLE Produto
```

```
(
```

```
CodProduto INT IDENTITY (1000, 01) PRIMARY KEY,  
Nome_Produto VARCHAR(30) NOT NULL,  
Preco Money,  
QtdEstoque_Produto INT CHECK (QtdEstoque_Produto >= 0)  
);
```

CREATE TABLE Fornecedor

```
(  
  
    CNPJ CHAR(14) PRIMARY KEY,  
    Nome VARCHAR(50) NOT NULL,  
    Telefone CHAR(9) NOT NULL,  
    Cep CHAR(8) NOT NULL,  
    Logradouro VARCHAR(50) NOT NULL,  
    Num_Endereco INT NOT NULL,  
    Rua VARCHAR(50) NOT NULL  
);
```

CREATE TABLE NotaFiscal (

```
    Id_Nota INT PRIMARY KEY NOT NULL,  
    Data_Compra DATE NOT NULL,
```

CPF_NotaFiscal CHAR(11) NOT NULL,
Id_Funcionario INT NOT NULL,
FOREIGN KEY (CPF_NotaFiscal) REFERENCES Cliente(CPFCliente),
FOREIGN KEY (Id_Funcionario) REFERENCES Funcionario(Id_Funcionario)
);

CREATE TABLE Nf_Produto (

Id_nota INT NOT NULL,
Cod_Produto INT NOT NULL ,
Qtd INT NOT NULL,
Preco MONEY NOT NULL,
PRIMARY KEY (CodProduto, Id_nota),
FOREIGN KEY (Cod_Produto) REFERENCES Produto(CodProduto),
FOREIGN KEY (Id_nota) REFERENCES NotaFiscal(Id_Nota)
);

CREATE TABLE Fn_Produto (

Cod_Produto INT PRIMARY KEY NOT NULL,
CNPJ CHAR(14) NOT NULL,
FOREIGN KEY (Cod_Produto) REFERENCES Produto(CodProduto)
);

go

-- Alimentação das tabelas

INSERT INTO Cliente (NomeCliente, CPFCliente, CEP_Cliente, Cidade_Cliente, Estado_Cliente, Numero_Cliente)

VALUES

('Flavio Silva', '70491779038', '51270691', 'Recife', 'Pernambuco', 27583346),

('Francisco Lima', '25612829027', '50900480', 'Recife', 'Pernambuco', 25742811),

('Jeferson Lucena', '86256459091', '51110972', 'Recife', 'Pernambuco', 32793763),

('Lohane Vêkanandre de Raio Laser bala de Icekiss', '39208876039', '51300120', 'Recife', 'Pernambuco', 23874751);

INSERT INTO Fornecedor (CNPJ, Nome, Telefone, Cep, Logradouro, Num_Endereco, Rua)

VALUES

('18274836', 'Thiago Ribeiro', '87997527878', '18274836', 'Rua Nova 14', 118, '50'),

('19064275', 'Gustavo Domingues', '81987588889', '19064275', 'Rua Velha 13', 300, '49'),

('12783786', 'Pedro Gabriel', '61987884527', '12783786', 'Rua semi nova 15', 450, '60');

**INSERT INTO Funcionario (Nome_Funcionario, CPF_Funcionario,
Salario_Funcionario, Cargo_Funcionario)**

VALUES

**('Thiago Ribeiro', '12345678980', 2527, 'Vendedor'),
('Caio Vinicius', '98765432165', 2343, 'Repositor'),
('Pedro Gabriel', '45678912387', 2200, 'Caixa'),
('Gustavo Domingues', '27554864778', 3330, 'Gerente');**

SET IDENTITY_INSERT Produto ON;

INSERT INTO Produto (CodProduto, Nome_Produto, QtdEstoque_Produto)

VALUES

(1001, 'camisa_social', 30),

(1002, 'camisa_polo', 40),

(1003, 'regata', 10),

(1004, 'camisa_estampada', 40),

(1005, 'camisa_feminina', 30),

(1006, 'camisa_agostinho_carrara', 30),

(1007, 'bermuda_jeans', 20),

(1008, 'bermuda_treino', 30),

(1009, 'shorts_praia', 15),

(1010, 'calca_jeans', 30),

(1011, 'calca_moletom', 20),

(1012, 'calca_social', 10),

(1013, 'cueca_box', 20),

(1014, 'cueca_copinho', 20),

(1015, 'cueca_C&K', 20),

(1016, 'saptos_social', 15),

(1017, 'sapa_tenis', 10),

(1018, 'sapato_cano_longo', 13),

(1019, 'bota', 9),

(1020, 'havaianas', 18),

(1021, 'kenner', 8),

(1022, 'vestidos', 15),

(1023, 'camisa_jeans_feminina', 20),

(1024, 'top', 25),

(1025, 'calca_feminina', 30),

(1026, 'saia', 20),

(1027, 'minissaia', 15),

(1028, 'biquini_parte_de-cima', 10),

(1029, 'biquini_parte_de_baixo', 10),

(1030, 'biquini_conjunto', 9),

(1031, 'calcinha', 19),

(1032, 'calcinha_fio_dental', 15),

(1033, 'salto', 14),

(1034, 'rasteirinha', 12),

(1035, 'bota_feminina', 9),

(1036, 'sapato_feminino', 12);

**INSERT INTO NotaFiscal (Id_Nota, Data_Compra, CPF_NotaFiscal,
Id_Funcionario)**

VALUES

**(345346353, '2023-09-12', '70491779038', 1000),
(565753544, '2023-02-23', '86256459091', 1000),
(838372912, '2023-08-13', '25612829027', 1010);**

INSERT INTO Ponto (DataRegistro, IdFuncionario)

VALUES

**('2023-10-15', 1000),
('2023-09-12', 1010),
('2023-08-20', 1020),
('2023-09-12', 1030);**

INSERT INTO Nf_Produto (Id_nota, Qtd, Cod_Produto, Preco)

VALUES

**('345346353', '2', '1001', '30.4'),
('565753544', '9', '1002', '45.4'),**

```
('838372912', '7', '1003', '67.4');
```

```
INSERT INTO Fn_Produto (Cod_Produto, CNPJ)
```

```
VALUES
```

```
(1001, '18274836'),
```

```
(1007, '19064275'),
```

```
(1023, '12783786');
```

```
-- Consulta
```

```
-- 2 inner join
```

```
SELECT
```

```
    c.NomeCliente,
```

```
    f.Nome_Funcionario,
```

```
    nf.Id_Nota
```

```
FROM
```

```
    Cliente c
```

```
JOIN
```

```
    NotaFiscal nf ON c.CPFCliente = nf.CPF_NotaFiscal
```

JOIN

Funcionario f ON nf.Id_Funcionario = f.Id_Funcionario;

-- Utilizando as funções (SUM, COUNT, AVG, MAX, MIN)

SELECT

**SUM(Salario_Funcionario) AS SomaSalario,
COUNT(Id_Funcionario) AS QuantidadeFuncionarios,
AVG(CAST(Salario_Funcionario AS DECIMAL(10, 2))) AS MediaSalario,
MAX(Salario_Funcionario) AS MaxSalario,
MIN(Salario_Funcionario) AS MinSalario**

FROM Funcionario;

SELECT

**SUM(QtdEstoque_Produto) AS SomaEstoque,
COUNT(CodProduto) AS QuantidadeProdutos,
AVG(QtdEstoque_Produto * 1.0) AS MediaEstoque,
MAX(QtdEstoque_Produto) AS MaxEstoque,
MIN(QtdEstoque_Produto) AS MinEstoque**

FROM Produto;

SELECT

**COUNT(1) AS TotalRegistros,
AVG(1.0) AS MediaRegistros,
MAX(DataRegistro) AS MaxData,
MIN(DataRegistro) AS MinData**

FROM Ponto;

-- Automação

-- TRIGGER

CREATE TRIGGER TriggerProdutos

ON Produto

AFTER INSERT, DELETE

AS

BEGIN

-- Verificação após a inserção

IF EXISTS (SELECT 1 FROM inserted)

BEGIN

PRINT 'Novos produtos foram adicionados.';

END

-- Verificação após a exclusão

IF EXISTS (SELECT 1 FROM deleted)

BEGIN

PRINT 'Produtos foram removidos.';

END

-- Lógica comum para visualização (seleção)

SELECT

CodProduto,

Nome_Produto,

QtdEstoque_Produto

FROM

Produto;

END;

go

-- FUNCTION

CREATE FUNCTION GerenciarFuncionario

(

@Opcao VARCHAR(10),

@Id_Funcionario INT = NULL,

@Nome_Funcionario VARCHAR(50) = NULL,

@CPF_Funcionario CHAR(11) = NULL,

@Salario_Funcionario DECIMAL(10, 2) = NULL,

@Cargo_Funcionario VARCHAR(30) = NULL

)

RETURNS TABLE

AS

RETURN

(

-- Seleção de Funcionários

SELECT

Id_Funcionario,

Nome_Funcionario,

CPF_Funcionario,

Salario_Funcionario,

Cargo_Funcionario

FROM

Funcionario

WHERE

(@Opcao = 'Mostrar' OR @Opcao IS NULL) AND

(Id_Funcionario = @Id_Funcionario OR @Id_Funcionario IS NULL) AND

(Nome_Funcionario = @Nome_Funcionario OR @Nome_Funcionario IS NULL) AND

(CPF_Funcionario = @CPF_Funcionario OR @CPF_Funcionario IS NULL) AND

(Salario_Funcionario = @Salario_Funcionario OR @Salario_Funcionario IS NULL) AND

(Cargo_Funcionario = @Cargo_Funcionario OR @Cargo_Funcionario IS NULL)

-- Inserção de Funcionários

UNION ALL

SELECT

Id_Funcionario,

Nome_Funcionario,

CPF_Funcionario,

Salario_Funcionario,

Cargo_Funcionario

FROM

Funcionario

WHERE

@Opcao = 'Adicionar' AND

@Nome_Funcionario IS NOT NULL AND

@CPF_Funcionario IS NOT NULL AND

@Salario_Funcionario IS NOT NULL AND

@Cargo_Funcionario IS NOT NULL

-- Remoção de Funcionários

UNION ALL

SELECT

Id_Funcionario,

Nome_Funcionario,

CPF_Funcionario,

Salario_Funcionario,

Cargo_Funcionario

FROM

Funcionario

WHERE

@Opcao = 'Remover' AND

@Id_Funcionario IS NOT NULL

);

-- Exemplo de Uso

-- Mostrar todos os funcionários

-- SELECT * FROM GerenciarFuncionario('Mostrar');

-- Adicionar um novo funcionário

**-- Nota: Certifique-se de fornecer valores válidos e únicos para
CPF_Funcionario**

**-- SELECT * FROM GerenciarFuncionario('Adicionar', NULL, 'Novo Funcionário',
'12345678901', 2500, 'Atendente');**

-- Remover um funcionário pelo ID

-- Nota: Substitua 1001 pelo ID do funcionário que você deseja remover

-- SELECT * FROM GerenciarFuncionario('Remover', 1001);

-- PROCEDURE

-- Exemplo de procedure para atualizar o salário de um funcionário

CREATE PROCEDURE AtualizarSalarioFuncionario6

@CPF_Funcionario char(11),

@NovoSalario int

AS

BEGIN

UPDATE Funcionario

SET Salario_Funcionario = @NovoSalario

WHERE CPF_Funcionario = @CPF_Funcionario;

END;

-- Exemplo de chamada da procedure

**-- EXEC dbo.AtualizarSalarioFuncionario6 @CPF_Funcionario = 12345678980,
@NovoSalario = 2527;**

-- INDEX

-- Criando um INDEX na tabela de maior volume de consulta alfa

CREATE INDEX idx_NomeCliente

ON Cliente (NomeCliente);