

Prova 1234

```
-- Verifica se o banco de dados já existe, se sim, remove-o
IF DB_ID('ProvaVendasDB') IS NOT NULL -- Renomeando o DB para indicar a evolução
BEGIN
    ALTER DATABASE ProvaVendasDB SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
    DROP DATABASE ProvaVendasDB
END
GO
```

```
-- Cria o banco de dados evoluído
CREATE DATABASE ProvaVendasDB;
GO
```

```
-- Usa o banco de dados criado
USE ProvaVendasDB;
GO
```

```
-- Cria a tabela Clientes (SEM ALTERAÇÕES ESTRUTURAIS)
CREATE TABLE Clientes (
    ClienteID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,
    Nome VARCHAR(100) NOT NULL,
    Sobrenome VARCHAR(100) NOT NULL,
    Email VARCHAR(255) UNIQUE
);
GO
```

```
-- Cria a tabela EnderecosCliente (NOVA TABELA)
```

```
CREATE TABLE EnderecosCliente (  
    EnderecoID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    ClienteID INT NOT NULL,  
    TipoEndereco VARCHAR(50) NOT NULL, -- Ex: 'Entrega', 'Cobrança', 'Residencial'  
    Logradouro VARCHAR(255) NOT NULL,  
    Numero VARCHAR(50),  
    Complemento VARCHAR(100),  
    Bairro VARCHAR(100),  
    Cidade VARCHAR(100) NOT NULL,  
    Estado VARCHAR(2) NOT NULL,  
    CEP VARCHAR(10),  
    FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID)  
);  
GO
```

-- Cria a tabela Fornecedores (NOVA TABELA)

```
CREATE TABLE Fornecedores (  
    FornecedorID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    NomeFornecedor VARCHAR(255) NOT NULL UNIQUE,  
    Contato VARCHAR(100),  
    Telefone VARCHAR(20),  
    Email VARCHAR(255) UNIQUE,  
    Cidade VARCHAR(100),  
    Estado VARCHAR(2)  
);  
GO
```

-- Cria a tabela Produtos (MODIFICADA para incluir FornecedorID)

```

CREATE TABLE Produtos (

    ProdutoID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,

    NomeProduto VARCHAR(255) NOT NULL,

    Categoria VARCHAR(100),

    Preco DECIMAL(10, 2) NOT NULL,

    Estoque INT NOT NULL,

    FornecedorID INT, -- Adicionando referência ao fornecedor

    FOREIGN KEY (FornecedorID) REFERENCES Fornecedores(FornecedorID) -- Chave
    estrangeira para Fornecedores

);

GO

```

-- Cria a tabela Pedidos (MODIFICADA para incluir EnderecoEntregaID e EnderecoCobrancaID)

```

CREATE TABLE Pedidos (

    PedidoID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,

    ClienteID INT NOT NULL,

    DataPedido DATE NOT NULL,

    ValorTotal DECIMAL(10, 2) NOT NULL,

    StatusPedido VARCHAR(50) DEFAULT 'Pendente', -- Ex: 'Pendente', 'Processando',
    'Enviado', 'Entregue', 'Cancelado'

    EnderecoEntregaID INT, -- Endereço para onde o pedido será enviado

    EnderecoCobrancaID INT, -- Endereço para cobrança do pedido

    FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),

    FOREIGN KEY (EnderecoEntregaID) REFERENCES EnderecosCliente(EnderecoID),

    FOREIGN KEY (EnderecoCobrancaID) REFERENCES EnderecosCliente(EnderecoID)

);

GO

```

-- Cria a tabela ItensPedido (SEM ALTERAÇÕES ESTRUTURAIS)

```
CREATE TABLE ItensPedido (  
    ItemID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    PedidoID INT NOT NULL,  
    ProdutoID INT NOT NULL,  
    Quantidade INT NOT NULL,  
    PrecoUnitario DECIMAL(10, 2) NOT NULL,  
    SubtotalItem DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (PedidoID) REFERENCES Pedidos(PedidoID),  
    FOREIGN KEY (ProdutoID) REFERENCES Produtos(ProdutoID)  
);  
GO
```

-- Cria a tabela Pagamentos (NOVA TABELA)

```
CREATE TABLE Pagamentos (  
    PagamentoID INT PRIMARY KEY IDENTITY(1,1) NOT NULL,  
    PedidoID INT NOT NULL,  
    DataPagamento DATE NOT NULL,  
    MetodoPagamento VARCHAR(50) NOT NULL, -- Ex: 'Cartão de Crédito', 'Boleto', 'PIX'  
    ValorPago DECIMAL(10, 2) NOT NULL,  
    StatusPagamento VARCHAR(50) DEFAULT 'Aprovado', -- Ex: 'Aprovado', 'Pendente',  
    'Recusado', 'Estornado'  
    FOREIGN KEY (PedidoID) REFERENCES Pedidos(PedidoID)  
);  
GO
```

-- Usa o banco de dados criado

```
USE ProvaVendasDB;  
GO
```

-- Insere dados na tabela Clientes (15 clientes)

INSERT INTO Clientes (Nome, Sobrenome, Email) VALUES

('Ana', 'Silva', 'ana.silva.ev@email.com'),
('Bruno', 'Martins', 'bruno.m.ev@email.com'),
('Carla', 'Souza', 'carla.s.ev@email.com'),
('Daniel', 'Oliveira', 'daniel.o.ev@email.com'),
('Eduarda', 'Santos', 'eduarda.s.ev@email.com'),
('Fernando', 'Lima', 'fernando.l.ev@email.com'),
('Gabriela', 'Costa', 'gabriela.c.ev@email.com'),
('Hugo', 'Pereira', 'hugo.p.ev@email.com'),
('Isabela', 'Almeida', 'isabela.a.ev@email.com'),
('João', 'Rodrigues', 'joao.r.ev@email.com'),
('Luiza', 'Fernandes', 'luiza.f.ev@email.com'),
('Miguel', 'Gomes', 'miguel.g.ev@email.com'),
('Natália', 'Carvalho', 'natalia.c.ev@email.com'),
('Otávio', 'Rocha', 'otavio.r.ev@email.com'),
('Paula', 'Lima', 'paula.l.ev@email.com');

GO

-- Insere dados na tabela Fornecedores (5 fornecedores)

INSERT INTO Fornecedores (NomeFornecedor, Contato, Telefone, Email, Cidade, Estado) VALUES

('Tech Solutions Ltda.', 'João Fornecedor', '(11) 9876-5432', 'contato@techsol.com',
'São Paulo', 'SP'),
('Eletronica Brasil S/A', 'Maria Contato', '(21) 1234-5678', 'vendas@ebrasil.com', 'Rio de
Janeiro', 'RJ'),

```
('Moveis Modernos', 'Pedro Representante', '(31) 5555-1111',  
'comercial@moveis.com', 'Belo Horizonte', 'MG'),  
  
('Audio Prime', 'Ana Atendimento', '(41) 8888-2222', 'suporte@audioprime.com',  
'Curitiba', 'PR'),  
  
('Acessórios Top', 'Rafael Vendas', '(51) 7777-3333', 'contato@acessorios.com', 'Porto  
Alegre', 'RS');  
  
GO
```

-- Insere dados na tabela Produtos (15 produtos, relacionados a fornecedores)

```
INSERT INTO Produtos (NomeProduto, Categoria, Preco, Estoque, FornecedorID)  
VALUES
```

```
('Laptop Gamer X', 'Eletrônicos', 4500.00, 15, 1), -- Tech Solutions  
('Smartphone Z', 'Eletrônicos', 2200.00, 50, 1), -- Tech Solutions  
('Teclado Mecânico', 'Acessórios', 350.00, 30, 5), -- Acessórios Top  
('Mouse Óptico', 'Acessórios', 80.00, 100, 5), -- Acessórios Top  
('Monitor UltraWide', 'Eletrônicos', 1800.00, 10, 2), -- Eletronica Brasil  
('Cadeira Gamer', 'Móveis', 1200.00, 5, 3), -- Moveis Modernos  
('Webcam HD', 'Acessórios', 150.00, 40, 5), -- Acessórios Top  
('Fones de Ouvido BT', 'Áudio', 250.00, 60, 4), -- Audio Prime  
('Microfone USB', 'Áudio', 400.00, 20, 4), -- Audio Prime  
('Mesa Ergonômica', 'Móveis', 750.00, 8, 3), -- Moveis Modernos  
('Smart TV 50"', 'Eletrônicos', 3000.00, 25, 2), -- Eletronica Brasil  
('Soundbar', 'Áudio', 600.00, 35, 4), -- Audio Prime  
('Mochila para Laptop', 'Acessórios', 120.00, 80, 5), -- Acessórios Top  
('Gabinete Gamer', 'Eletrônicos', 550.00, 18, 1), -- Tech Solutions  
('Impressora Multifuncional', 'Eletrônicos', 800.00, 22, 2); -- Eletronica Brasil  
  
GO
```

-- Insere dados na tabela EnderecosCliente (múltiplos endereços para alguns clientes)

INSERT INTO EnderecosCliente (ClienteID, TipoEndereco, Logradouro, Numero, Complemento, Bairro, Cidade, Estado, CEP) VALUES

(1, 'Residencial', 'Rua das Flores', '100', 'Apto 101', 'Centro', 'São Paulo', 'SP', '01000-000'), -- Ana

(1, 'Entrega', 'Av. Principal', '500', NULL, 'Bairro Novo', 'São Paulo', 'SP', '01001-000'), -
- Ana (Entrega diferente)

(2, 'Residencial', 'Rua do Sol', '250', NULL, 'Copacabana', 'Rio de Janeiro', 'RJ', '20000-000'), -- Bruno

(3, 'Residencial', 'Av. Liberdade', '30', 'Casa', 'Savassi', 'Belo Horizonte', 'MG', '30000-000'), -- Carla

(3, 'Cobrança', 'Rua da Moeda', '10', 'Sala 5', 'Centro', 'Belo Horizonte', 'MG', '30001-000'), -- Carla (Cobrança diferente)

(4, 'Residencial', 'Rua das Pedras', '123', NULL, 'Jardins', 'São Paulo', 'SP', '01002-000'),
-- Daniel

(5, 'Residencial', 'Av. Atlântica', '1000', 'Apto 502', 'Centro', 'Porto Alegre', 'RS', '90000-000'), -- Eduarda

(6, 'Residencial', 'Rua da Praia', '50', NULL, 'Ipanema', 'Rio de Janeiro', 'RJ', '20001-000'), -- Fernando

(7, 'Residencial', 'Rua Sete de Setembro', '77', NULL, 'Centro', 'São Paulo', 'SP', '01003-000'), -- Gabriela

(8, 'Residencial', 'Rua Verde', '88', 'Fundos', 'Água Verde', 'Curitiba', 'PR', '80000-000'), -
- Hugo

(9, 'Residencial', 'Rua Amarela', '99', NULL, 'Centro', 'Belo Horizonte', 'MG', '30002-000'), -- Isabela

(10, 'Residencial', 'Rua Azul', '111', 'Bloco A', 'Morumbi', 'São Paulo', 'SP', '01004-000'),
-- João

(11, 'Residencial', 'Rua Rosa', '222', NULL, 'Centro', 'Porto Alegre', 'RS', '90001-000'), --
Luiza

(12, 'Residencial', 'Rua Roxa', '333', NULL, 'Savassi', 'Belo Horizonte', 'MG', '30003-000'),
-- Miguel

(13, 'Residencial', 'Rua Laranja', '444', 'Casa 2', 'Jardins', 'São Paulo', 'SP', '01005-000'), -
- Natália

(14, 'Residencial', 'Rua Marrom', '555', NULL, 'Copacabana', 'Rio de Janeiro', 'RJ', '20002-000'), -- Otávio

(15, 'Residencial', 'Rua Cinza', '666', NULL, 'Centro', 'Curitiba', 'PR', '80001-000'); --
Paula

GO

-- Insere dados na tabela Pedidos (20 pedidos, relacionados a clientes e endereços)

-- Certifique-se que os EnderecoEntregaID e EnderecoCobrancaID pertencem ao
ClienteID

INSERT INTO Pedidos (ClienteID, DataPedido, ValorTotal, StatusPedido,
EnderecoEntregaID, EnderecoCobrancaID) VALUES

(1, '2023-01-10', 0.00, 'Entregue', 2, 1), -- Ana, Entrega: Av. Principal, Cobrança: Rua
das Flores

(2, '2023-01-15', 0.00, 'Entregue', 3, 3), -- Bruno, Entrega/Cobrança: Rua do Sol

(3, '2023-02-20', 0.00, 'Entregue', 4, 5), -- Carla, Entrega: Av. Liberdade, Cobrança:
Rua da Moeda

(1, '2023-03-01', 0.00, 'Entregue', 2, 1), -- Ana, Entrega: Av. Principal, Cobrança: Rua
das Flores

(4, '2023-03-10', 0.00, 'Entregue', 6, 6), -- Daniel, Entrega/Cobrança: Rua das Pedras

(5, '2023-04-05', 0.00, 'Entregue', 7, 7), -- Eduarda, Entrega/Cobrança: Av. Atlântica

(6, '2023-04-10', 0.00, 'Entregue', 8, 8), -- Fernando, Entrega/Cobrança: Rua da Praia

(7, '2023-05-01', 0.00, 'Entregue', 9, 9), -- Gabriela, Entrega/Cobrança: Rua Sete de
Setembro

(8, '2023-05-15', 0.00, 'Entregue', 10, 10), -- Hugo, Entrega/Cobrança: Rua Verde

(9, '2023-06-20', 0.00, 'Entregue', 11, 11), -- Isabela, Entrega/Cobrança: Rua Amarela

(10, '2024-01-05', 0.00, 'Processando', 12, 12), -- João, Entrega/Cobrança: Rua Azul

(1, '2024-01-10', 0.00, 'Processando', 2, 1), -- Ana, Entrega: Av. Principal, Cobrança: Rua
das Flores

(2, '2024-02-01', 0.00, 'Pendente', 3, 3), -- Bruno, Entrega/Cobrança: Rua do Sol

(3, '2024-02-10', 0.00, 'Pendente', 4, 5), -- Carla, Entrega: Av. Liberdade, Cobrança:
Rua da Moeda

(4, '2024-03-01', 0.00, 'Enviado', 6, 6), -- Daniel, Entrega/Cobrança: Rua das Pedras
(11, '2024-03-05', 0.00, 'Entregue', 13, 13), -- Luiza
(12, '2024-03-10', 0.00, 'Processando', 14, 14), -- Miguel
(13, '2024-03-15', 0.00, 'Pendente', 15, 15), -- Natália
(14, '2024-03-20', 0.00, 'Enviado', 16, 16), -- Otávio
(15, '2024-03-25', 0.00, 'Entregue', 17, 17); -- Paula
GO

-- Insere dados na tabela ItensPedido (relacionando com os pedidos e produtos)

-- Calcula o SubtotalItem

INSERT INTO ItensPedido (PedidoID, ProdutoID, Quantidade, PrecoUnitario, SubtotalItem) VALUES

(1, 1, 1, 4500.00, 4500.00), -- Pedido 1: Laptop Gamer X
(2, 2, 1, 2200.00, 2200.00), -- Pedido 2: Smartphone Z
(3, 3, 2, 350.00, 700.00), -- Pedido 3: Teclado Mecânico x2
(4, 4, 3, 80.00, 240.00), -- Pedido 4: Mouse Óptico x3
(5, 5, 1, 1800.00, 1800.00), -- Pedido 5: Monitor UltraWide
(6, 6, 1, 1200.00, 1200.00), -- Pedido 6: Cadeira Gamer
(7, 7, 5, 150.00, 750.00), -- Pedido 7: Webcam HD x5
(8, 8, 2, 250.00, 500.00), -- Pedido 8: Fones de Ouvido BT x2
(9, 9, 1, 400.00, 400.00), -- Pedido 9: Microfone USB
(10, 10, 1, 750.00, 750.00), -- Pedido 10: Mesa Ergonômica
(11, 1, 1, 4500.00, 4500.00), -- Pedido 11: Laptop Gamer X
(12, 2, 1, 2200.00, 2200.00), -- Pedido 12: Smartphone Z
(12, 4, 1, 80.00, 80.00), -- Pedido 12: Mouse Óptico
(13, 6, 1, 1200.00, 1200.00), -- Pedido 13: Cadeira Gamer
(14, 3, 1, 350.00, 350.00), -- Pedido 14: Teclado Mecânico
(11, 11, 1, 3000.00, 3000.00), -- Pedido 11: Smart TV 50"

(12, 12, 1, 600.00, 600.00), -- Pedido 12: Soundbar
(13, 13, 2, 120.00, 240.00), -- Pedido 13: Mochila para Laptop x2
(14, 14, 1, 550.00, 550.00), -- Pedido 14: Gabinete Gamer
(15, 15, 1, 800.00, 800.00); -- Pedido 15: Impressora Multifuncional
GO

-- Insere dados na tabela Pagamentos (registrando pagamentos para alguns pedidos)

INSERT INTO Pagamentos (PedidoID, DataPagamento, MetodoPagamento, ValorPago, StatusPagamento) VALUES

(1, '2023-01-10', 'Cartão de Crédito', 4500.00, 'Aprovado'),
(2, '2023-01-15', 'Cartão de Crédito', 2200.00, 'Estornado'),
(3, '2023-02-20', 'Boleto', 700.00, 'Aprovado'),
(4, '2023-03-01', 'PIX', 240.00, 'Aprovado'),
(5, '2023-03-10', 'Cartão de Crédito', 1800.00, 'Aprovado'),
(6, '2023-04-05', 'Cartão de Crédito', 1200.00, 'Aprovado'),
(7, '2023-04-10', 'PIX', 750.00, 'Aprovado'),
(8, '2023-05-15', 'Boleto', 500.00, 'Aprovado'),
(10, '2023-06-20', 'Cartão de Crédito', 750.00, 'Recusado'),
(11, '2024-01-05', 'Cartão de Crédito', 7500.00, 'Aprovado'), -- Pedido 11: Laptop + Smart TV
(12, '2024-01-10', 'PIX', 2280.00, 'Aprovado'), -- Pedido 12: Smartphone + Mouse + Soundbar
(14, '2024-03-01', 'Cartão de Crédito', 900.00, 'Aprovado'), -- Pedido 14: Teclado + Gabinete
(15, '2024-03-25', 'Boleto', 800.00, 'Aprovado'), -- Pedido 15: Impressora
(16, '2024-03-05', 'Cartão de Crédito', 3000.00, 'Aprovado'), -- Pedido 16: Luiza, Smart TV
(17, '2024-03-10', 'PIX', 1200.00, 'Aprovado'), -- Pedido 17: Miguel, Cadeira Gamer
(19, '2024-03-20', 'Cartão de Crédito', 550.00, 'Aprovado'); -- Pedido 19: Otávio, Gabinete Gamer

GO

Parte 2: Consultas SQL (15 Perguntas)

1 - Liste o nome de todos os fornecedores: Selecione a coluna NomeFornecedor da tabela Fornecedores.

2 - Produtos na categoria 'Áudio' com estoque superior a 30: Selecione o NomeProduto e Estoque para todos os produtos que pertencem à categoria 'Áudio' e cujo estoque atual é maior que 30. Use WHERE.

3 - Pedidos realizados no primeiro trimestre de 2024: Selecione o PedidoID, ClienteID e DataPedido de todos os pedidos cuja data esteja entre '2024-01-01' e '2024-03-31' (inclusive). Use WHERE.

4 - Contagem de endereços por cliente: Conte quantos endereços (EnderecoID) estão cadastrados para cada cliente. Apresente o ClienteID e a contagem de endereços. Use GROUP BY.

5 - Valor total pago por método de pagamento: Calcule a soma do ValorPago para cada método de pagamento (MetodoPagamento). Apresente o método de pagamento e o valor total pago. Use GROUP BY.

6 - Pagamentos com status 'Recusado' ou 'Estornado': Liste o PagamentoID, PedidoID, DataPagamento e StatusPagamento para todos os pagamentos cujo status seja 'Recusado' ou 'Estornado'. Use WHERE com OR ou IN.

7 - Fornecedoros que fornecem mais de 2 produtos: Liste o NomeFornecedor para todos os fornecedores que estão associados a mais de 2 produtos diferentes. Use JOIN com Produtos, GROUP BY pelo fornecedor e HAVING para filtrar pela contagem de produtos.

8 - Pedidos e o nome completo do cliente que o realizou: Selecione o PedidoID, DataPedido e o Nome completo do cliente (Nome + Sobrenome) para todos os pedidos. Utilize um JOIN entre as tabelas Pedidos e Clientes.

9 - Itens de pedido com nome do produto e nome do fornecedor: Selecione o PedidoID, o NomeProduto, o NomeFornecedor e a Quantidade para todos os itens de pedido. Utilize JOINS entre ItensPedido, Produtos e Fornecedores. Ordene os resultados por PedidoID e, secundariamente, por NomeProduto (ascendente).

10 - Pedidos entregues em São Paulo (SP): Liste o PedidoID, DataPedido e o Nome completo do cliente para todos os pedidos cujo endereço de entrega (EnderecoEntregaID) esteja em 'São Paulo' no estado de 'SP'. Use JOINS entre Pedidos, Clientes e EnderecosCliente, e WHERE para filtrar pela cidade e estado do endereço de entrega.

11 - Clientes que possuem mais de um tipo de endereço cadastrado: Liste o Nome e Sobrenome dos clientes que possuem pelo menos dois registros na tabela EnderecosCliente com tipos de endereço (TipoEndereco) distintos. Use GROUP BY por cliente e HAVING na contagem distinta de tipos de endereço.

12 - Categorias de produtos com valor total de estoque superior a R\$ 10000.00: Liste as Categorias de produtos onde a soma total do valor do estoque (Estoque * Preço) para todos os produtos nessa categoria é maior que 10000.00. Use GROUP BY pela categoria e HAVING na soma (Estoque * Preço).

13 - Pedidos com valor total diferente da soma dos pagamentos aprovados: Liste o PedidoID e o ValorTotal para os pedidos onde a soma dos ValorPago dos pagamentos com StatusPagamento 'Aprovado' associados a este pedido é diferente do ValorTotal do pedido. Use JOINS com Pagamentos, WHERE para filtrar pagamentos aprovados, GROUP BY pelo pedido e HAVING para comparar a soma dos pagamentos com o ValorTotal do pedido.

14 - Crie uma View chamada PedidosProcessandoView que selecione todas as colunas da tabela Pedidos apenas para os pedidos cujo StatusPedido seja 'Processando'.

Em seguida, consulte esta View (SELECT * FROM PedidosProcessandoView;) para listar estes pedidos.

Clientes que têm pedidos 'Processando' (usando View):

15 - Utilize a View PedidosProcessandoView criada na pergunta anterior.

Consulte esta View e utilize um JOIN com a tabela Clientes para listar o Nome e Sobrenome dos clientes que aparecem nesta View.

Prova 567

```
-- Verifica se o banco de dados já existe, se sim, remove-o
IF DB_ID('ProvaProjetosDB') IS NOT NULL -- Renomeando o DB para indicar a evolução
BEGIN
    ALTER DATABASE ProvaProjetosDB SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
    DROP DATABASE ProvaProjetosDB;
END
GO
```

```
-- Cria o banco de dados evoluído
CREATE DATABASE ProvaProjetosDB;
GO
```

```
-- Usa o banco de dados criado
USE ProvaProjetosDB;
GO
```

```
-- Cria a tabela Departamentos (SEM ALTERAÇÕES ESTRUTURAIS)
CREATE TABLE Departamentos (
    DepartamentoID INT PRIMARY KEY IDENTITY(1,1),
    NomeDepartamento VARCHAR(100) NOT NULL UNIQUE
);
GO
```

```
-- Cria a tabela Funcionarios (SEM ALTERAÇÕES ESTRUTURAIS)
CREATE TABLE Funcionarios (
    FuncionarioID INT PRIMARY KEY IDENTITY(1,1),
```

```
Nome VARCHAR(100) NOT NULL,  
Sobrenome VARCHAR(100) NOT NULL,  
Email VARCHAR(255) UNIQUE,  
Cargo VARCHAR(100),  
Salario DECIMAL(10, 2),  
DepartamentoID INT,  
FOREIGN KEY (DepartamentoID) REFERENCES Departamentos(DepartamentoID)  
);  
GO
```

-- Cria a tabela Projetos (SEM ALTERAÇÕES ESTRUTURAIS)

```
CREATE TABLE Projetos (  
    ProjetoID INT PRIMARY KEY IDENTITY(1,1),  
    NomeProjeto VARCHAR(255) NOT NULL UNIQUE,  
    DataInicio DATE,  
    DataFimPrevista DATE,  
    StatusProjeto VARCHAR(50) DEFAULT 'Ativo' -- Ex: 'Ativo', 'Concluído', 'Cancelado',  
    'Em Pausa'  
);  
GO
```

-- Cria a tabela AtribuicoesProjeto (quem trabalha na equipe do projeto - SEM ALTERAÇÕES ESTRUTURAIS)

```
CREATE TABLE AtribuicoesProjeto (  
    AtribuicaoID INT PRIMARY KEY IDENTITY(1,1),  
    FuncionarioID INT NOT NULL,  
    ProjetoID INT NOT NULL,  
    DataAtribuicao DATE NOT NULL,
```

```
FuncaoNoProjeto VARCHAR(100), -- Ex: 'Desenvolvedor', 'Gerente', 'Analista',  
'Testador'
```

```
HorasDedicadas DECIMAL(10, 2), -- Horas estimadas ou registradas no projeto
```

```
FOREIGN KEY (FuncionarioID) REFERENCES Funcionarios(FuncionarioID),
```

```
FOREIGN KEY (ProjetoID) REFERENCES Projetos(ProjetoID)
```

```
);
```

```
GO
```

```
-- Cria a tabela Tarefas (NOVA TABELA)
```

```
CREATE TABLE Tarefas (
```

```
TarefaID INT PRIMARY KEY IDENTITY(1,1),
```

```
ProjetoID INT NOT NULL,
```

```
NomeTarefa VARCHAR(255) NOT NULL,
```

```
Descricao TEXT,
```

```
StatusTarefa VARCHAR(50) DEFAULT 'Pendente', -- Ex: 'Pendente', 'Em Andamento',  
'Concluída', 'Bloqueada'
```

```
Prioridade VARCHAR(50) DEFAULT 'Média', -- Ex: 'Baixa', 'Média', 'Alta', 'Crítica'
```

```
DataCriacao DATE DEFAULT GETDATE(),
```

```
DataVencimento DATE,
```

```
FuncionarioResponsavelID INT, -- Funcionário primariamente responsável pela tarefa
```

```
FOREIGN KEY (ProjetoID) REFERENCES Projetos(ProjetoID),
```

```
FOREIGN KEY (FuncionarioResponsavelID) REFERENCES Funcionarios(FuncionarioID)
```

```
);
```

```
GO
```

```
-- Cria a tabela Marcos (NOVA TABELA)
```

```
CREATE TABLE Marcos (
```

```
MarcoID INT PRIMARY KEY IDENTITY(1,1),
```

```
ProjetoID INT NOT NULL,
```

```

NomeMarco VARCHAR(255) NOT NULL,

DataMarco DATE NOT NULL,

Descricao TEXT,

FOREIGN KEY (ProjetoID) REFERENCES Projetos(ProjetoID)
);

GO

-- Cria a tabela DependenciasTarefa (NOVA TABELA - Relacionamento N:N entre Tarefas)
CREATE TABLE DependenciasTarefa (
    DependencialID INT PRIMARY KEY IDENTITY(1,1),
    TarefaDependenteID INT NOT NULL, -- A tarefa que depende
    TarefaPredecessoraID INT NOT NULL, -- A tarefa da qual depende
    TipoDependencia VARCHAR(50) DEFAULT 'Fim-Início', -- Ex: 'Fim-Início', 'Início-Início'
    FOREIGN KEY (TarefaDependenteID) REFERENCES Tarefas(TarefaID),
    FOREIGN KEY (TarefaPredecessoraID) REFERENCES Tarefas(TarefaID)
);

GO

-- Usa o banco de dados criado
USE ProvaProjetosDB;

GO

-- Insere dados na tabela Departamentos
INSERT INTO Departamentos (NomeDepartamento) VALUES
('Tecnologia'),
('Recursos Humanos'),
('Financeiro'),
('Marketing'),

```


('Design'),
('Operações'),
('Jurídico');
GO

-- Insere dados na tabela Funcionarios (20 funcionários em diferentes departamentos e cargos)

INSERT INTO Funcionarios (Nome, Sobrenome, Email, Cargo, Salario, DepartamentoID)
VALUES

('Carlos', 'Mendes', 'carlos.m@empresa.com', 'Gerente de TI', 8000.00, 1), -- Tecnologia

('Mariana', 'Ferreira', 'mariana.f@empresa.com', 'Desenvolvedora Senior', 6500.00, 1),
-- Tecnologia

('Pedro', 'Costa', 'pedro.c@empresa.com', 'Analista de RH', 4500.00, 2), -- Recursos
Humanos

('Julia', 'Alves', 'julia.a@empresa.com', 'Contador Senior', 5500.00, 3), -- Financeiro

('Rafael', 'Gomes', 'rafael.g@empresa.com', 'Analista de Marketing', 5000.00, 4), --
Marketing

('Fernanda', 'Carvalho', 'fernanda.c@empresa.com', 'Designer UI/UX', 6000.00, 5), --
Design

('Gustavo', 'Rocha', 'gustavo.r@empresa.com', 'Desenvolvedor Junior', 3800.00, 1), --
Tecnologia

('Larissa', 'Lima', 'larissa.l@empresa.com', 'Recrutador', 4200.00, 2), -- Recursos
Humanos

('Marcelo', 'Dias', 'marcelo.d@empresa.com', 'Analista Financeiro', 5200.00, 3), --
Financeiro

('Carolina', 'Nunes', 'carolina.n@empresa.com', 'Especialista de SEO', 4800.00, 4), --
Marketing

('Roberto', 'Sousa', 'roberto.s@empresa.com', 'Gerente de Projetos', 7500.00, 1), --
Tecnologia

('Amanda', 'Oliveira', 'amanda.o@empresa.com', 'Analista de Operações', 4700.00, 6), -
Operações

('Bruno', 'Santos', 'bruno.s@empresa.com', 'Advogado Corporativo', 7000.00, 7), --
Jurídico

('Cristina', 'Almeida', 'cristina.a@empresa.com', 'Desenvolvedora Pleno', 5500.00, 1), --
Tecnologia

('Diego', 'Rodrigues', 'diego.r@empresa.com', 'Testador QA', 4000.00, 1), -- Tecnologia

('Elaine', 'Costa', 'elaine.c@empresa.com', 'Assistente Administrativo', 3500.00, 6), --
Operações

('Fabio', 'Lima', 'fabio.l@empresa.com', 'Gerente de RH', 7000.00, 2), -- Recursos
Humanos

('Giovana', 'Gomes', 'giovana.g@empresa.com', 'Designer Gráfico', 5800.00, 5), --
Design

('Henrique', 'Martins', 'henrique.m@empresa.com', 'Especialista Financeiro', 6800.00,
3), -- Financeiro

('Ingrid', 'Souza', 'ingrid.s@empresa.com', 'Analista Jurídico', 5300.00, 7); -- Jurídico
GO

-- Insere dados na tabela Projetos (10 projetos com diferentes status e datas)

INSERT INTO Projetos (NomeProjeto, DataInicio, DataFimPrevista, StatusProjeto)
VALUES

('Sistema Interno v1.0', '2023-03-01', '2023-09-30', 'Concluído'),
('Novo Website Institucional', '2023-06-15', '2023-12-20', 'Concluído'),
('App Mobile Clientes', '2024-01-10', '2024-07-31', 'Ativo'),
('Campanha Marketing 2024', '2024-02-01', '2024-11-30', 'Ativo'),
('Otimização Processos Financeiros', '2024-03-10', '2024-09-15', 'Ativo'),
('Implantação Novo RH System', '2024-04-01', '2025-03-31', 'Ativo'),
('Redesign UI/UX', '2024-05-20', '2024-08-31', 'Em Pausa'),
('Auditoria Interna 2024', '2024-06-01', '2024-07-15', 'Ativo'),
('Expansão Mercado Internacional', '2024-07-01', '2025-12-31', 'Ativo'),
('Atualização Infraestrutura TI', '2024-08-15', '2025-02-28', 'Pendente');

GO

-- Insere dados na tabela AtribuicoesProjeto (quem trabalha na equipe do projeto - mais atribuições)

INSERT INTO AtribuicoesProjeto (FuncionarioID, ProjetoID, DataAtribuicao, FuncaoNoProjeto, HorasDedicadas) VALUES

(2, 1, '2023-03-01', 'Desenvolvedor', 300.00), -- Mariana no Sistema Interno
(7, 1, '2023-03-15', 'Desenvolvedor', 250.00), -- Gustavo no Sistema Interno
(1, 1, '2023-03-05', 'Gerente', 100.00), -- Carlos no Sistema Interno
(14, 1, '2023-04-01', 'Desenvolvedor', 280.00), -- Cristina no Sistema Interno
(15, 1, '2023-04-10', 'Testador QA', 150.00), -- Diego no Sistema Interno
(6, 2, '2023-06-15', 'Designer UI/UX', 150.00),-- Fernanda no Website
(5, 2, '2023-06-20', 'Analista de Marketing', 80.00), -- Rafael no Website
(11, 2, '2023-06-10', 'Gerente de Projetos', 50.00), -- Roberto no Website
(18, 2, '2023-07-01', 'Designer Gráfico', 120.00), -- Giovana no Website
(2, 3, '2024-01-10', 'Desenvolvedor', 120.00), -- Mariana no App Mobile
(7, 3, '2024-01-15', 'Desenvolvedor', 100.00), -- Gustavo no App Mobile
(14, 3, '2024-01-20', 'Desenvolvedor', 110.00), -- Cristina no App Mobile
(15, 3, '2024-02-01', 'Testador QA', 80.00), -- Diego no App Mobile
(11, 3, '2024-01-08', 'Gerente de Projetos', 30.00), -- Roberto no App Mobile
(5, 4, '2024-02-01', 'Analista de Marketing', 60.00), -- Rafael na Campanha
(10, 4, '2024-02-05', 'Especialista de SEO', 90.00), -- Carolina na Campanha
(8, 6, '2024-04-01', 'Recrutador', 40.00), -- Larissa no RH System
(3, 6, '2024-04-10', 'Analista de RH', 50.00),-- Pedro no RH System
(17, 6, '2024-04-15', 'Gerente de RH', 30.00), -- Fabio no RH System
(4, 5, '2024-03-10', 'Contador Senior', 70.00), -- Julia nos Processos Financeiros
(9, 5, '2024-03-12', 'Analista Financeiro', 80.00),-- Marcelo nos Processos Financeiros
(19, 5, '2024-03-15', 'Especialista Financeiro', 60.00), -- Henrique nos Processos Financeiros
(6, 7, '2024-05-20', 'Designer UI/UX', 20.00),-- Fernanda no Redesign (Em Pausa)

(18, 7, '2024-05-25', 'Designer Gráfico', 15.00), -- Giovana no Redesign (Em Pausa)
(11, 7, '2024-05-18', 'Gerente de Projetos', 10.00), -- Roberto no Redesign (Em Pausa)
(4, 8, '2024-06-01', 'Contador Senior', 40.00), -- Julia na Auditoria
(9, 8, '2024-06-05', 'Analista Financeiro', 35.00), -- Marcelo na Auditoria
(13, 8, '2024-06-10', 'Advogado Corporativo', 20.00), -- Bruno na Auditoria
(19, 8, '2024-06-12', 'Especialista Financeiro', 30.00); -- Henrique na Auditoria
GO

-- Insere dados na tabela Tarefas (várias tarefas para diferentes projetos, com responsáveis e datas)

-- Projeto 3: App Mobile Clientes

INSERT INTO Tarefas (ProjetoID, NomeTarefa, Descricao, StatusTarefa, Prioridade, DataVencimento, FuncionarioResponsavelID) VALUES

(3, 'Desenvolver Login', 'Implementar funcionalidade de login do usuário', 'Em Andamento', 'Alta', '2024-02-15', 2), -- Mariana

(3, 'Criar Tela Inicial', 'Desenvolver a tela principal do aplicativo', 'Pendente', 'Alta', '2024-02-28', 7), -- Gustavo

(3, 'Configurar Banco de Dados', 'Configurar a base de dados para o app', 'Concluída', 'Crítica', '2024-01-25', 14), -- Cristina

(3, 'Testar Funcionalidades', 'Realizar testes nas funcionalidades implementadas', 'Pendente', 'Média', '2024-03-10', 15), -- Diego

(3, 'Integrar APIs', 'Integrar o app com as APIs de backend', 'Em Andamento', 'Alta', '2024-03-20', 2), -- Mariana

(3, 'Desenvolver Tela de Perfil', 'Criar a tela de perfil do usuário', 'Pendente', 'Média', '2024-03-31', 7), -- Gustavo

(3, 'Realizar Testes de Performance', 'Testar o desempenho do aplicativo', 'Pendente', 'Baixa', '2024-04-15', 15); -- Diego

-- Projeto 4: Campanha Marketing 2024

INSERT INTO Tarefas (ProjetoID, NomeTarefa, Descricao, StatusTarefa, Prioridade, DataVencimento, FuncionarioResponsavelID) VALUES

(4, 'Criar Conteúdo para Redes Sociais', 'Elaborar posts e artes para mídias sociais', 'Em Andamento', 'Alta', '2024-03-10', 5), -- Rafael

(4, 'Planejar Anúncios Pagos', 'Definir estratégia e orçamento para anúncios online', 'Concluída', 'Alta', '2024-02-20', 10), -- Carolina

(4, 'Monitorar Métricas', 'Acompanhar o desempenho da campanha', 'Ativa', 'Média', '2024-11-30', 5), -- Rafael

(4, 'Otimizar SEO', 'Realizar otimizações para motores de busca', 'Em Andamento', 'Média', '2024-05-31', 10); -- Carolina

-- Projeto 5: Otimização Processos Financeiros

INSERT INTO Tarefas (ProjetoID, NomeTarefa, Descricao, StatusTarefa, Prioridade, DataVencimento, FuncionarioResponsavelID) VALUES

(5, 'Analisar Processos Atuais', 'Mapear e documentar os processos financeiros existentes', 'Concluída', 'Crítica', '2024-03-31', 4), -- Julia

(5, 'Identificar Gargalos', 'Encontrar pontos de ineficiência nos processos', 'Em Andamento', 'Alta', '2024-04-15', 9), -- Marcelo

(5, 'Propor Soluções', 'Desenvolver propostas para otimização', 'Pendente', 'Alta', '2024-05-01', 19), -- Henrique

(5, 'Implementar Mudanças', 'Executar as mudanças nos processos', 'Pendente', 'Alta', '2024-06-30', 4); -- Julia

INSERT INTO Tarefas (ProjetoID, NomeTarefa, Descricao, StatusTarefa, Prioridade, DataVencimento, FuncionarioResponsavelID) VALUES

(7, 'Pesquisar Tendências', 'Pesquisar as últimas tendências de design', 'Em Pausa', 'Baixa', '2024-06-15', 6), -- Fernanda

(7, 'Criar Wireframes', 'Desenvolver wireframes para as novas telas', 'Pendente', 'Média', '2024-06-30', 18); -- Giovana

-- Projeto 8: Auditoria Interna 2024

INSERT INTO Tarefas (ProjetoID, NomeTarefa, Descricao, StatusTarefa, Prioridade, DataVencimento, FuncionarioResponsavelID) VALUES

(8, 'Coletar Documentos', 'Reunir a documentação necessária para a auditoria', 'Em Andamento', 'Alta', '2024-06-20', 4), -- Julia

(8, 'Analisar Registros Financeiros', 'Examinar os registros contábeis', 'Pendente', 'Alta', '2024-06-30', 9), -- Marcelo

(8, 'Verificar Conformidade Legal', 'Checar a conformidade com as leis', 'Pendente', 'Alta', '2024-07-05', 13); -- Bruno

GO

-- Insere dados na tabela Marcos (marcos para alguns projetos)

INSERT INTO Marcos (ProjetoID, NomeMarco, DataMarco, Descricao) VALUES

(3, 'Lançamento Beta App', '2024-06-30', 'Versão beta do aplicativo disponível para testes internos'),

(3, 'Lançamento Oficial App', '2024-07-31', 'Lançamento público do aplicativo nas lojas'),

(4, 'Início da Campanha', '2024-03-01', 'Início oficial da campanha de marketing'),

(4, 'Relatório de Meio Termo', '2024-07-31', 'Relatório de desempenho da campanha até o momento'),

(5, 'Propostas de Otimização Aprovadas', '2024-05-15', 'Aprovação das propostas de melhoria de processo'),

(8, 'Relatório de Auditoria Final', '2024-07-15', 'Entrega do relatório final da auditoria');

GO

-- Insere dados na tabela DependenciasTarefa (definindo algumas dependências)

-- Dependências no Projeto 3 (App Mobile Clientes)

-- "Criar Tela Inicial" depende de "Desenvolver Login"

INSERT INTO DependenciasTarefa (TarefaDependenteID, TarefaPredecessorID, TipoDependencia) VALUES

(1, 2, 'Fim-Início'); -- TarefaID 1 depende de TarefaID 2

-- "Integrar APIs" depende de "Configurar Banco de Dados"

INSERT INTO DependenciasTarefa (TarefaDependentelID, TarefaPredecessoraID, TipoDependencia) VALUES

(3, 4, 'Fim-Início'); -- TarefaID 3 depende de TarefaID 4

Parte 2: Consultas SQL (15 Perguntas)

1 - Liste o nome de todas as tarefas do projeto 'App Mobile Clientes': Selecione a coluna NomeTarefa da tabela Tarefas para o projeto cujo NomeProjeto é 'App Mobile Clientes'. Use WHERE e JOIN com a tabela Projetos.

2 - Marcos previstos para o ano de 2024: Selecione o NomeMarco e a DataMarco para todos os marcos cuja data esteja no ano de 2024. Use WHERE.

3 - Tarefas com status 'Em Andamento' e prioridade 'Alta': Selecione o NomeTarefa, StatusTarefa e Prioridade para todas as tarefas que estão com StatusTarefa como 'Em Andamento' e Prioridade como 'Alta'. Use WHERE.

4 - Contagem de tarefas por status para o projeto 'Otimização Processos Financeiros': Conte quantas tarefas existem para cada StatusTarefa dentro do projeto 'Otimização Processos Financeiros'. Apresente o StatusTarefa e a contagem. Use JOIN com Projetos, WHERE para filtrar o projeto e GROUP BY.

5 - Contagem de tarefas atribuídas a cada funcionário responsável: Conte quantas tarefas (TarefaID) estão associadas a cada funcionário responsável (FuncionarioResponsavelID). Apresente o FuncionarioID e a contagem de tarefas. Use GROUP BY.

6 - Projetos que possuem tarefas com prioridade 'Crítica': Liste o NomeProjeto para todos os projetos que contêm pelo menos uma tarefa com Prioridade igual a 'Crítica'. Use JOINS com Tarefas e WHERE. Garanta que cada nome de projeto apareça apenas uma vez.

7 - Funcionários responsáveis por tarefas em mais de um projeto: Liste o Nome e Sobrenome dos funcionários que são responsáveis por tarefas (FuncionarioResponsavelID na tabela Tarefas) em 2 ou mais projetos distintos. Use JOINS com Tarefas e Projetos, GROUP BY pelo funcionário e HAVING na contagem distinta de projetos.

8 - Tarefas e o nome do funcionário responsável: Selecione o NomeTarefa e o Nome completo do funcionário responsável (Nome + Sobrenome) para todas as tarefas que possuem um responsável atribuído. Utilize um JOIN entre as tabelas Tarefas e Funcionarios.

9 - Tarefas e suas tarefas predecessoras: Liste o NomeTarefa da tarefa dependente e o NomeTarefa da tarefa predecessora para todas as dependências registradas. Utilize JOINS na tabela DependenciasTarefa com a tabela Tarefas (duas vezes, usando aliases). Ordene os resultados pelo nome da tarefa dependente.

10 - Projetos e a contagem de tarefas por status: Para cada projeto, conte quantas tarefas existem em cada StatusTarefa. Apresente o NomeProjeto, StatusTarefa e a contagem. Use JOINS e GROUP BY em múltiplos campos.

11 - Projetos com mais de 5 tarefas 'Concluída': Liste o NomeProjeto para os projetos que possuem mais de 5 tarefas com StatusTarefa igual a 'Concluída'. Use JOINS com Tarefas, WHERE para filtrar o status, GROUP BY pelo projeto e HAVING para filtrar pela contagem de tarefas concluídas.

12 - Tarefas do projeto 'App Mobile Clientes' ordenadas por data de vencimento (mais próxima primeiro): Selecione o NomeTarefa, DataVencimento e StatusTarefa para todas as tarefas do projeto 'App Mobile Clientes'. Ordene os resultados pela DataVencimento em ordem ascendente (ORDER BY ASC).

13 - Departamentos com salário médio de funcionário acima de R\$ 5000.00 e que possuem funcionários atribuídos a projetos: Liste o NomeDepartamento para os departamentos onde o salário médio de seus funcionários é superior a 5000.00 E que possuem pelo menos um funcionário atribuído a algum projeto. Use JOINS, WHERE para filtrar salários, GROUP BY pelo departamento, HAVING para filtrar pela média salarial e combine com uma condição que verifique a existência de atribuições em projetos (pode usar EXISTS ou outro JOIN e HAVING COUNT > 0).

14 - Crie uma View chamada TarefasEmAndamentoView que selecione todas as colunas da tabela Tarefas apenas para as tarefas cujo StatusTarefa seja 'Em Andamento'.

- Em seguida, consulte esta View (SELECT * FROM TarefasEmAndamentoView;) para listar estas tarefas.

15 - Utilize a View TarefasEmAndamentoView criada na pergunta anterior.

Consulte esta View e utilize JOINS com as tabelas Projetos e Funcionarios para listar o NomeTarefa, o NomeProjeto ao qual pertence e o Nome completo do funcionário responsável.

Prova 890

-- Verifica se o banco de dados já existe, se sim, remove-o

IF DB_ID('ProvaUniversidadeDB') IS NOT NULL

BEGIN

ALTER DATABASE ProvaUniversidadeDB SET SINGLE_USER WITH ROLLBACK
IMMEDIATE;

DROP DATABASE ProvaUniversidadeDB;

END

GO

-- Cria o banco de dados

CREATE DATABASE ProvaUniversidadeDB;

GO

-- Usa o banco de dados criado

USE ProvaUniversidadeDB;

GO

-- Cria a tabela Departamentos

CREATE TABLE Departamentos (

DepartamentoID INT PRIMARY KEY IDENTITY(1,1),

NomeDepartamento VARCHAR(100) NOT NULL UNIQUE,

ChefeDepartamento VARCHAR(100)

);

GO

-- Cria a tabela Professores (NOVA TABELA)

```

CREATE TABLE Professores (
    ProfessorID INT PRIMARY KEY IDENTITY(1,1),
    Nome VARCHAR(100) NOT NULL,
    Sobrenome VARCHAR(100) NOT NULL,
    Email VARCHAR(255) UNIQUE,
    Contrato VARCHAR(50), -- Ex: 'Integral', 'Parcial'
    Salario DECIMAL(10, 2),
    DepartamentoID INT, -- Departamento do professor
    FOREIGN KEY (DepartamentoID) REFERENCES Departamentos(DepartamentoID)
);
GO

```

-- Cria a tabela Cursos (MODIFICADA para incluir ProfessorID)

```

CREATE TABLE Cursos (
    CursoID INT PRIMARY KEY IDENTITY(1,1),
    NomeCurso VARCHAR(255) NOT NULL UNIQUE,
    Creditos INT NOT NULL,
    DepartamentoID INT NOT NULL,
    ProfessorID INT, -- Professor responsável pelo curso (NOVA COLUNA)
    FOREIGN KEY (DepartamentoID) REFERENCES Departamentos(DepartamentoID),
    FOREIGN KEY (ProfessorID) REFERENCES Professores(ProfessorID) -- NOVA CHAVE
    ESTRANGEIRA
);
GO

```

-- Cria a tabela Alunos (SEM ALTERAÇÕES ESTRUTURAIS)

```

CREATE TABLE Alunos (
    AlunoID INT PRIMARY KEY IDENTITY(1,1),

```

```

Nome VARCHAR(100) NOT NULL,
Sobrenome VARCHAR(100) NOT NULL,
Email VARCHAR(255) UNIQUE,
DataMatricula DATE NOT NULL,
MajorDepartamentoID INT, -- Departamento do curso principal do aluno
FOREIGN KEY (MajorDepartamentoID) REFERENCES
Departamentos(DepartamentoID)
);
GO

-- Cria a tabela Matriculas (SEM ALTERAÇÕES ESTRUTURAIS)
CREATE TABLE Matriculas (
    MatriculaID INT PRIMARY KEY IDENTITY(1,1),
    AlunoID INT NOT NULL,
    CursoID INT NOT NULL,
    AnoSemestre VARCHAR(10) NOT NULL, -- Ex: '2023-1', '2023-2', '2024-1'
    NotaFinal DECIMAL(4, 2), -- Pode ser NULL se o curso não terminou
    StatusMatricula VARCHAR(50) DEFAULT 'Ativa', -- Ex: 'Ativa', 'Concluída', 'Trancada'
    UNIQUE (AlunoID, CursoID, AnoSemestre), -- Um aluno não pode se matricular no
    mesmo curso no mesmo semestre
    FOREIGN KEY (AlunoID) REFERENCES Alunos(AlunoID),
    FOREIGN KEY (CursoID) REFERENCES Cursos(CursoID)
);
GO

USE ProvaUniversidadeDB;
GO

-- Insere dados na tabela Departamentos (mantido)

```

```
INSERT INTO Departamentos (NomeDepartamento, ChefeDepartamento) VALUES
('Ciência da Computação', 'Dr. Alan Turing'),
('Engenharia Elétrica', 'Dra. Ada Lovelace'),
('Física', 'Dr. Albert Einstein'),
('Matemática', 'Dra. Sofia Kovalevskaya');
GO
```

-- Insere dados na tabela Professores (NOVOS INSERTS)

```
INSERT INTO Professores (Nome, Sobrenome, Email, Contrato, Salario,
DepartamentoID) VALUES
('Carlos', 'Silva', 'carlos.silva@univ.com', 'Integral', 9000.00, 1), -- CC
('Mariana', 'Santos', 'mariana.santos@univ.com', 'Integral', 8500.00, 1), -- CC
('Fernando', 'Pereira', 'fernando.p@univ.com', 'Parcial', 5000.00, 1), -- CC
('Ana', 'Costa', 'ana.costa@univ.com', 'Integral', 8800.00, 2), -- EE
('Pedro', 'Oliveira', 'pedro.o@univ.com', 'Parcial', 5200.00, 2), -- EE
('Julia', 'Lima', 'julia.l@univ.com', 'Integral', 9200.00, 3), -- Física
('Rafael', 'Souza', 'rafael.s@univ.com', 'Integral', 8700.00, 4), -- Matemática
('Isabela', 'Martins', 'isabela.m@univ.com', 'Parcial', 4800.00, 4); -- Matemática
GO
```

-- Insere dados na tabela Cursos (MODIFICADO para incluir ProfessorID)

-- É importante que os ProfessorIDs referenciem IDs válidos da tabela Professores

```
INSERT INTO Cursos (NomeCurso, Creditos, DepartamentoID, ProfessorID) VALUES
('Introdução à Programação', 4, 1, 1), -- Carlos Silva (ProfessorID 1)
('Estruturas de Dados', 4, 1, 2), -- Mariana Santos (ProfessorID 2)
('Circuitos Elétricos', 5, 2, 4), -- Ana Costa (ProfessorID 4)
('Termodinâmica', 4, 3, 6), -- Julia Lima (ProfessorID 6)
('Cálculo I', 6, 4, 7), -- Rafael Souza (ProfessorID 7)
```

('Banco de Dados', 4, 1, 2), -- Mariana Santos (ProfessorID 2)
('Algoritmos Avançados', 4, 1, 1), -- Carlos Silva (ProfessorID 1)
('Eletrônica Digital', 5, 2, 5), -- Pedro Oliveira (ProfessorID 5)
('Física Quântica', 5, 3, 6), -- Julia Lima (ProfessorID 6)
('Álgebra Linear', 5, 4, 8); -- Isabela Martins (ProfessorID 8)

GO

-- Insere dados na tabela Alunos (mantido)

INSERT INTO Alunos (Nome, Sobrenome, Email, DataMatricula, MajorDepartamentoID)
VALUES

('João', 'Silva', 'joao.s@univ.com', '2022-08-10', 1), -- CC
('Maria', 'Fernandes', 'maria.f@univ.com', '2022-08-10', 2), -- EE
('Pedro', 'Gomes', 'pedro.g@univ.com', '2023-02-01', 1), -- CC
('Ana', 'Souza', 'ana.s@univ.com', '2023-02-01', 4), -- Matemática
('Lucas', 'Lima', 'lucas.l@univ.com', '2024-02-01', 1), -- CC
('Carolina', 'Pereira', 'carolina.p@univ.com', '2024-02-01', 2), -- EE
('Rafael', 'Costa', 'rafael.c@univ.com', '2023-08-15', 3), -- Física
('Isabela', 'Martins', 'isabela.m@univ.com', '2022-08-10', 4); -- Matemática

GO

-- Insere dados na tabela Matriculas (mantido - os IDs de curso e aluno ainda são válidos)

INSERT INTO Matriculas (AlunoID, CursoID, AnoSemestre, NotaFinal, StatusMatricula)
VALUES

(1, 1, '2022-2', 85.5, 'Concluída'), -- João em Intro Prog (CursoID 1)
(1, 5, '2022-2', 70.0, 'Concluída'), -- João em Cálculo I (CursoID 5)
(2, 3, '2022-2', 78.0, 'Concluída'), -- Maria em Circuitos (CursoID 3)
(2, 5, '2022-2', 65.5, 'Concluída'), -- Maria em Cálculo I (CursoID 5)
(3, 1, '2023-1', 92.0, 'Concluída'), -- Pedro em Intro Prog (CursoID 1)

(3, 2, '2023-2', 88.0, 'Concluída'), -- Pedro em Estruturas de Dados (CursoID 2)
(3, 6, '2024-1', NULL, 'Ativa'), -- Pedro em Banco de Dados (CursoID 6)
(4, 5, '2023-1', 95.0, 'Concluída'), -- Ana em Cálculo I (CursoID 5)
(4, 10, '2023-2', 80.0, 'Concluída'), -- Ana em Álgebra Linear (CursoID 10)
(5, 1, '2024-1', NULL, 'Ativa'), -- Lucas em Intro Prog (CursoID 1)
(5, 6, '2024-1', NULL, 'Ativa'), -- Lucas em Banco de Dados (CursoID 6)
(6, 3, '2024-1', NULL, 'Ativa'), -- Carolina em Circuitos (CursoID 3)
(7, 4, '2023-2', 75.0, 'Concluída'), -- Rafael em Termodinâmica (CursoID 4)
(7, 9, '2024-1', NULL, 'Ativa'), -- Rafael em Física Quântica (CursoID 9)
(8, 5, '2022-2', 60.0, 'Concluída'), -- Isabela em Cálculo I (CursoID 5)
(8, 10, '2023-1', 72.0, 'Concluída'); -- Isabela em Álgebra Linear (CursoID 10)
GO

1 - Liste o nome e sobrenome de todos os professores: Selecione as colunas Nome e Sobrenome da tabela Professores.

2 - Cursos do departamento de 'Engenharia Elétrica' com 5 créditos: Selecione o NomeCurso para todos os cursos que pertencem ao departamento de 'Engenharia Elétrica' e que possuem exatamente 5 créditos. Use WHERE e JOIN com a tabela Departamentos.

3 - Alunos matriculados após 01/01/2023: Selecione o Nome e Sobrenome de todos os alunos cuja data de matrícula (DataMatricula) é posterior a '2023-01-01'. Use WHERE.

4 - Contagem total de alunos por departamento Major: Conte quantos alunos têm seu curso principal (MajorDepartamentoID) em cada departamento. Apresente o NomeDepartamento e a contagem de alunos. Use JOIN e GROUP BY.

5 - Salário médio dos professores por tipo de contrato: Calcule o salário médio (Salario) dos professores para cada tipo de contrato (Contrato). Apresente o tipo de contrato e o salário médio. Use GROUP BY.

6 - Cursos com mais de 2 alunos matriculados no semestre '2022-2': Liste o NomeCurso para todos os cursos que tiveram mais de 2 alunos matriculados no semestre '2022-2'. Use JOINS com Matriculas, WHERE para filtrar o semestre, GROUP BY pelo curso e HAVING para filtrar os grupos pela contagem de matrículas.

7 - Professores que ministram mais de um curso: Liste o Nome e Sobrenome dos professores que são responsáveis por mais de um curso. Use JOINS com Cursos, GROUP BY pelo professor e HAVING para filtrar pela contagem de cursos.

8 - Alunos e o nome do curso em que estão matriculados (apenas matrículas Ativas): Selecione o Nome completo do aluno (Nome + Sobrenome) e o NomeCurso para todas as matrículas cujo StatusMatricula seja 'Ativa'. Use JOINS entre Alunos, Matriculas e Cursos, e WHERE.

9 - Matrículas concluídas com nota final, ordenadas por nota (maior para menor): Selecione o MatriculaID, AlunoID, CursoID, AnoSemestre e NotaFinal para todas as matrículas cujo StatusMatricula seja 'Concluída' e NotaFinal não seja nula. Ordene os resultados pela NotaFinal em ordem decrescente (ORDER BY DESC).

10 - Departamentos com salário médio de professor integral acima de R\$ 8600.00: Liste o NomeDepartamento para os departamentos onde o salário médio dos professores com contrato 'Integral' é superior a 8600.00. Use JOINS, WHERE para filtrar por contrato, GROUP BY pelo departamento e HAVING para filtrar pela média salarial.

11 - Alunos que concluíram pelo menos um curso com nota superior a 90: Liste o Nome e Sobrenome dos alunos que possuem pelo menos uma matrícula Concluída com uma NotaFinal maior que 90. Use JOINS com Matriculas, WHERE para filtrar as matrículas relevantes, GROUP BY pelo aluno e HAVING para garantir que ele tenha tal matrícula.

12 - Cursos e os nomes dos seus respectivos professores, ordenados por departamento e nome do curso: Liste o NomeCurso, o Nome e Sobrenome do professor responsável, e o NomeDepartamento ao qual o curso pertence. Use JOINS entre Cursos, Professores e Departamentos. Ordene os resultados por NomeDepartamento (ascendente) e depois por NomeCurso (ascendente).

13 - Contagem de matrículas concluídas por status e semestre: Conte quantas matrículas foram concluídas (StatusMatricula = 'Concluída') em cada AnoSemestre. Apresente o AnoSemestre e a contagem. Use WHERE e GROUP BY.

14 - View de Professores com Contrato Integral:

Crie uma View chamada ProfessoresIntegralView que selecione todos os professores cujo Contrato seja 'Integral', incluindo todas as colunas da tabela Professores.

Em seguida, consulte esta View (SELECT * FROM ProfessoresIntegralView;) para listar estes professores.

15 - Utilize a View ProfessoresIntegralView criada na pergunta anterior.

Consulte esta View e utilize um JOIN com a tabela Cursos para listar o NomeCurso e o Nome e Sobrenome do professor que o ministra, apenas para cursos ministrados ### por professores com contrato Integral.