Abaixo está a criação detalhada do banco de dados SQL Server, incluindo tabelas e suas respectivas relações. Foi incluída a lógica necessária para implementar as regras especificadas, como a obrigatoriedade de preenchimento dos campos e o vínculo entre os dados.

**1. Estrutura do Banco de Dados**

sql

Copiar código

-- Criação do banco de dados

CREATE DATABASE Empresa1;

GO

USE Empresa1;

GO

-- Tabela de Cursos

CREATE TABLE Cursos (

Id INT IDENTITY(1,1) PRIMARY KEY,

Descricao NVARCHAR(255) NOT NULL

);

-- Tabela de Turmas

CREATE TABLE Turmas (

Id INT IDENTITY(1,1) PRIMARY KEY,

Descricao NVARCHAR(255) NOT NULL,

CursoId INT NOT NULL,

CONSTRAINT FK\_Turmas\_Cursos FOREIGN KEY (CursoId) REFERENCES Cursos(Id) ON DELETE CASCADE

);

-- Tabela de Leads

CREATE TABLE Leads (

Id INT IDENTITY(1,1) PRIMARY KEY,

Nome NVARCHAR(255) NOT NULL,

Telefone NVARCHAR(20) NOT NULL,

Email NVARCHAR(255) NOT NULL,

CursoInteresse INT NOT NULL,

CONSTRAINT FK\_Leads\_Cursos FOREIGN KEY (CursoInteresse) REFERENCES Cursos(Id) ON DELETE CASCADE

);

-- Tabela de Alunos

CREATE TABLE Alunos (

Id INT IDENTITY(1,1) PRIMARY KEY,

CodigoMatricula INT NOT NULL UNIQUE,

Nome NVARCHAR(255) NOT NULL,

Telefone NVARCHAR(20) NOT NULL,

Email NVARCHAR(255) NOT NULL,

CursoId INT NOT NULL,

TurmaId INT NOT NULL,

DataCadastro DATETIME NOT NULL DEFAULT GETDATE(),

CONSTRAINT FK\_Alunos\_Cursos FOREIGN KEY (CursoId) REFERENCES Cursos(Id) ON DELETE CASCADE,

CONSTRAINT FK\_Alunos\_Turmas FOREIGN KEY (TurmaId) REFERENCES Turmas(Id) ON DELETE CASCADE

);

GO

**2. Regras de Negócio Implementadas**

1. **Obrigatoriedade de preenchimento**:
   * Todos os campos são definidos como NOT NULL para garantir que nenhum dado seja armazenado de forma incompleta.
2. **Relação entre tabelas**:
   * A tabela Leads possui uma chave estrangeira CursoInteresse, vinculada à tabela Cursos.
   * A tabela Turmas possui uma chave estrangeira CursoId, vinculando turmas ao curso correspondente.
   * A tabela Alunos possui chaves estrangeiras CursoId e TurmaId, garantindo que os alunos estejam vinculados a um curso e a uma turma válida.
3. **Código de matrícula**:
   * Será gerado automaticamente como um número sequencial único, que pode ser gerado por meio de uma função ou trigger.

**3. Exemplo de Função para Geração de Código de Matrícula**

sql

Copiar código

-- Função para gerar o próximo número de matrícula

CREATE FUNCTION GenerateCodigoMatricula()

RETURNS INT

AS

BEGIN

DECLARE @Codigo INT;

SELECT @Codigo = ISNULL(MAX(CodigoMatricula), 0) + 1 FROM Alunos;

RETURN @Codigo;

END;

GO

**4. Procedure para Matricular um Lead**

sql

Copiar código

-- Procedure para matricular um lead e vinculá-lo como aluno

CREATE PROCEDURE MatricularAluno

@LeadId INT,

@TurmaId INT

AS

BEGIN

SET NOCOUNT ON;

-- Verifica se o Lead existe

IF NOT EXISTS (SELECT 1 FROM Leads WHERE Id = @LeadId)

BEGIN

RAISERROR ('Lead não encontrado.', 16, 1);

RETURN;

END

-- Verifica se a turma está vinculada ao curso do Lead

DECLARE @CursoId INT;

SELECT @CursoId = CursoInteresse FROM Leads WHERE Id = @LeadId;

IF NOT EXISTS (SELECT 1 FROM Turmas WHERE Id = @TurmaId AND CursoId = @CursoId)

BEGIN

RAISERROR ('Turma inválida para o curso de interesse do lead.', 16, 1);

RETURN;

END

-- Insere o Lead como aluno

INSERT INTO Alunos (CodigoMatricula, Nome, Telefone, Email, CursoId, TurmaId)

SELECT

dbo.GenerateCodigoMatricula(),

Nome, Telefone, Email, CursoInteresse, @TurmaId

FROM Leads WHERE Id = @LeadId;

-- Exclui o Lead da tabela de Leads após matrícula

DELETE FROM Leads WHERE Id = @LeadId;

PRINT 'Aluno matriculado com sucesso!';

END;

GO

**5. Consultas para Filtros**

**Filtro por Nome**

sql

Copiar código

SELECT \* FROM Leads WHERE Nome LIKE '%<nome>%';

**Filtro por Email**

sql

Copiar código

SELECT \* FROM Leads WHERE Email LIKE '%<email>%';

**Filtro por Curso de Interesse**

sql

Copiar código

SELECT L.\*, C.Descricao AS CursoDescricao

FROM Leads L

JOIN Cursos C ON L.CursoInteresse = C.Id

WHERE C.Descricao LIKE '%<curso>%';

**6. Interface para Consulta e Matrícula**

Na aplicação, implemente uma interface gráfica que:

1. **Exibe os Leads capturados** com filtros por nome, email ou curso de interesse.
2. **Possui um botão "Matricular"** ao lado de cada Lead, que ao ser clicado:
   * Mostra uma lista de turmas disponíveis com base no curso de interesse.
   * Permite a matrícula do Lead como Aluno.

**7. Relatórios**

**Listagem de Alunos Matriculados**

sql

Copiar código

SELECT

A.Id,

A.CodigoMatricula,

A.Nome,

A.Email,

C.Descricao AS Curso,

T.Descricao AS Turma,

A.DataCadastro

FROM Alunos A

JOIN Cursos C ON A.CursoId = C.Id

JOIN Turmas T ON A.TurmaId = T.Id;

Essa estrutura cobre os requisitos funcionais e de negócio mencionados. Além disso, permite a evolução futura do sistema com novas funcionalidades, como gerenciamento de leads e turmas.