

Exercício Prático: Sistema Completo de Consultas Médicas

Abordagem Database-First

Desenvolver um sistema básico para o gerenciamento de médicos, pacientes e consultas médicas, utilizando SQL para criação e inserção de dados, e Entity Framework Core para gerar automaticamente as entidades e o contexto de acesso ao banco de dados.

1 Instruções do Exercício

1.1 Parte 1: Configuração do Banco de Dados com SQL

1. Criação do Banco de Dados e Tabelas

- Crie um novo banco de dados chamado **ConsultasMedicasDB**.
- Crie três tabelas: **Medico**, **Paciente** e **Consulta** com as seguintes características:
 - **Medico**: colunas **Id** (chave primária, auto-incremento), **Nome** (até 100 caracteres), **Especialidade** (até 50 caracteres) e **CRM** (até 20 caracteres).
 - **Paciente**: colunas **Id** (chave primária, auto-incremento), **Nome** (até 100 caracteres), **CPF** (11 caracteres) e **DataNascimento** (tipo DATE).
 - **Consulta**: colunas **Id** (chave primária, auto-incremento), **DataHora** (tipo DATETIME), **MedicoId** e **PacienteId**, que referenciam as chaves primárias das tabelas **Medico** e **Paciente**, respectivamente.

2. População Inicial de Dados

- Insira pelo menos dois médicos e dois pacientes na base de dados.
- Agende pelo menos três consultas, assegurando que uma das consultas seja marcada com um dos médicos inseridos e outra com o segundo médico.

3. Consulta de Dados para Verificação

- Realize consultas SQL para exibir todos os dados de cada tabela (**Medico**, **Paciente**, **Consulta**) e verifique se foram inseridos corretamente.

1.2 Parte 2: Criação do Projeto C# e Scaffolding com Entity Framework Core

1. Configuração do Projeto e Instalação do Entity Framework Core

- Crie um novo projeto de Console em C#.
- Instale os pacotes necessários para o Entity Framework Core e configure a conexão com o banco de dados **ConsultasMedicasDB** criado na Parte 1.

2. Scaffolding do Banco de Dados

- Utilize o comando de scaffolding do Entity Framework Core para gerar automaticamente as classes de modelo (**Medico**, **Paciente**, **Consulta**) e o DbContext (nome sugerido: **ConsultasMedicasDbContext**) no projeto C#.

3. Implementação de Consultas Usando LINQ

- No método **Main** do programa, realize as seguintes consultas com LINQ usando o DbContext gerado:

- Exiba todos os médicos com seus respectivos dados (Nome, Especialidade e CRM).
- Liste todos os pacientes, mostrando Nome, CPF e Data de Nascimento.
- Liste todas as consultas agendadas, incluindo a data e hora da consulta, nome do médico e nome do paciente.

1.3 Parte 3: Exercícios de Manipulação e Consultas Adicionais

1. Inserção de Dados com o DbContext

- Adicione mais um médico e mais um paciente ao banco de dados diretamente pelo código C#.
- Salve essas novas inserções no banco e exiba a lista atualizada de médicos e pacientes.

2. Consultas com Filtros e Ordenação

- Implemente uma consulta que liste todas as consultas futuras, ordenadas pela data.
- Filtre as consultas de um médico específico (usando o CRM como parâmetro) e exiba apenas as consultas desse médico.

3. Exclusão de Dados

- Realize a exclusão de uma consulta com uma data anterior ao dia de hoje.
- Mostre a lista atualizada de consultas para confirmar a exclusão.

2 Entrega do Exercício

- **Código Completo do Projeto**

- Inclua o projeto C# completo com todas as consultas realizadas.

- **Relatório com Capturas de Tela**

- Inclua capturas de tela mostrando as tabelas e registros criados no SQL, as consultas LINQ realizadas no projeto e os resultados obtidos.