**Exercício 1: ADO.NET - Manipulação Básica de Dados**

1. Crie um banco de dados simples contendo uma tabela "Clientes" com campos como "ID", "Nome", "Email" e "Telefone".
2. Use ADO.NET para criar uma conexão com o banco de dados.
3. Escreva um método que insira um novo cliente na tabela.
4. Implemente um método para atualizar as informações de um cliente existente.
5. Crie um método que liste todos os clientes da tabela.

**Exercício 2: Entity Framework - ORM Básico**

1. Crie um projeto com Entity Framework.
2. Defina uma classe de modelo chamada "Produto" com propriedades como "ID", "Nome", "Preço" e "Estoque".
3. Configure a conexão com o banco de dados usando o Entity Framework.
4. Escreva um método para adicionar um novo produto ao banco de dados.
5. Implemente um método para atualizar as informações de um produto existente.
6. Crie um método que liste todos os produtos.

**Exercício 3: ADO.NET - Consultas Avançadas**

1. Adicione uma tabela "Pedidos" ao banco de dados com campos como "ID", "Data", "ClienteID", "Total", etc.
2. Crie uma consulta ADO.NET que retorne todos os pedidos de um cliente específico.
3. Escreva uma consulta ADO.NET que retorne o total de vendas por mês.
4. Implemente uma consulta que liste os clientes que fizeram mais de um pedido nos últimos 3 meses.

**Exercício 4: Entity Framework - Relacionamentos e Consultas**

1. Crie uma classe de modelo "Categoria" e estabeleça um relacionamento de um-para-muitos com a classe "Produto".
2. Escreva uma consulta do Entity Framework que retorne todos os produtos de uma determinada categoria.
3. Implemente uma consulta que liste todas as categorias junto com a contagem de produtos em cada uma.
4. Crie uma consulta que retorne os produtos mais caros de cada categoria.

**Exercício 5: ADO.NET vs. Entity Framework - Performance**

1. Crie um cenário de teste para comparar a performance de consultas usando ADO.NET puro e Entity Framework.
2. Execute consultas complexas em ambas as abordagens e meça o tempo de execução.
3. Analise os resultados e identifique qual abordagem tem melhor performance em diferentes cenários.

**Exercício 6: Entity Framework - Migrations e Seed Data**

1. Crie um projeto com Entity Framework e habilita as migrações.
2. Configure o banco de dados para permitir que ele seja atualizado automaticamente com base nas alterações de modelo.
3. Escreva um script de seed data que insira alguns registros iniciais no banco de dados.

**Exercício 7: Entity Framework - Consultas Assíncronas**

1. Refatore suas consultas do Entity Framework para operações assíncronas.
2. Compare o desempenho das consultas síncronas com as consultas assíncronas em termos de tempo de resposta.

**Exercício 8: Entity Framework - Relacionamentos Complexos**

1. Implemente um cenário onde um pedido pode ter vários produtos e cada produto pode ter vários fornecedores.
2. Configure os relacionamentos e escreva consultas para recuperar informações completas de pedidos, produtos e fornecedores.

**Exercício 9: ADO.NET - Stored Procedures**

1. Crie uma stored procedure no banco de dados que retorne todos os pedidos de um cliente específico.
2. Use ADO.NET para chamar essa stored procedure a partir do código C#.

**Exercício 10: Entity Framework - Versionamento de Dados**

1. Introduza a noção de versionamento em sua aplicação.
2. Implemente lógica para registrar a versão de cada registro de dados em uma tabela.