**Entity Framework Core 8.0**

LAB 1-5

Model.cs

using System.Collections.Generic;

public class Category

{

    public int Id { get; set; }

    public string Name { get; set; }

    public List<Product> Products { get; set; }

}

public class Product

{

    public int Id { get; set; }

    public string Name { get; set; }

    public decimal Price { get; set; }

    public int CategoryId { get; set; }

    public Category Category { get; set; }

}

AppDbContext.cs

using Microsoft.EntityFrameworkCore;

public class AppDbContext : DbContext

{

    public DbSet<Product> Products { get; set; }

    public DbSet<Category> Categories { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

    {

        optionsBuilder.UseSqlServer("Server=localhost\\SQLEXPRESS;Database=RetailDb;Trusted\_Connection=True;Encrypt=False;");

    }

}

Program.cs

using System.Threading.Tasks;

using System;

using Microsoft.EntityFrameworkCore;

class Program

{

    static async Task Main(string[] args)

    {

        using var context = new AppDbContext();

        var electronics = new Category { Name = "Electronics" };

        var groceries = new Category { Name = "Groceries" };

        await context.Categories.AddRangeAsync(electronics, groceries);

        var product1 = new Product { Name = "Laptop", Price = 75000, Category = electronics };

        var product2 = new Product { Name = "Rice Bag", Price = 1200, Category = groceries };

        await context.Products.AddRangeAsync(product1, product2);

        await context.SaveChangesAsync();

        var products = await context.Products.ToListAsync();

        foreach (var p in products)

            Console.WriteLine($"{p.Name} - ₹{p.Price}");

        var productById = await context.Products.FindAsync(1);

        Console.WriteLine($"Found: {productById?.Name}");

        var expensive = await context.Products.FirstOrDefaultAsync(p => p.Price > 50000);

        Console.WriteLine($"Expensive: {expensive?.Name}");

    }

}

Output

