Lab 2: Setting Up the Database Context for a Retail Store

Code:

// File: Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using Microsoft.EntityFrameworkCore;

namespace RetailStoreApp

{

// Model: Product

public class Product

{

public int ProductID { get; set; }

public string Name { get; set; }

public decimal Price { get; set; }

public int Stock { get; set; }

}

// Model: Customer

public class Customer

{

public int CustomerID { get; set; }

public string FullName { get; set; }

public string Email { get; set; }

}

// Model: Order

public class Order

{

public int OrderID { get; set; }

public DateTime OrderDate { get; set; }

// Foreign Key

public int CustomerID { get; set; }

public Customer Customer { get; set; }

}

// DbContext: Retail Store Database

public class RetailDbContext : DbContext

{

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

public DbSet<Customer> Customers { get; set; }

public DbSet<Order> Orders { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlite("Data Source=retailstore.db");

}

}

class Program

{

static void Main(string[] args)

{

using (var db = new RetailDbContext())

{

// Create the database and tables if they don't exist

db.Database.EnsureCreated();

// Add a sample product

var product = new Product

{

Name = "Notebook",

Price = 49.99m,

Stock = 100

};

db.Products.Add(product);

db.SaveChanges();

// Show all products

Console.WriteLine("All Products:");

foreach (var p in db.Products.ToList())

{

Console.WriteLine($"ID: {p.ProductID}, Name: {p.Name}, Price: {p.Price}, Stock: {p.Stock}");

}

}

}

}

}

Output:  
