LAB 2:EF CORE 8.0 HOL

**AIM:** Setting Up the Database Context for a Retail Store

**Scenario:** The retail store wants to store product and category data in SQL Server.

**Models\Product.cs:**

using System.Collections.Generic;

namespace RetailInventory.Models

{

public class Product

{

public int ProductId { get; set; }

public string ProductName { get; set; } = string.Empty;

public decimal Price { get; set; }

public int Stock { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; } = null!;

public int SupplierId { get; set; }

public Supplier Supplier { get; set; } = null!;

}

}

**Models\Category.cs:**

namespace RetailInventory.Models

{

public class Category

{

public int CategoryId { get; set; }

public string CategoryName { get; set; } = string.Empty;

public List<Product> Products { get; set; } = new List<Product>();

}

}

**Models\Supplier.cs:**

using System.Collections.Generic;

namespace RetailInventory.Models

{

public class Supplier

{

public int SupplierId { get; set; }

public string SupplierName { get; set; } = string.Empty;

public string ContactEmail { get; set; } = string.Empty;

public List<Product> Products { get; set; } = new List<Product>();

}

}

**Data\RetailDbContext.cs:**

using Microsoft.EntityFrameworkCore;

using RetailInventory.Models;

namespace RetailInventory.Data

{

public class RetailDbContext : DbContext

{

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

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

public DbSet<Supplier> Suppliers { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

optionsBuilder.UseSqlite("Data Source=retail\_inventory.db");

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Category>()

.Property(c => c.CategoryName)

.IsRequired()

.HasMaxLength(100);

modelBuilder.Entity<Supplier>()

.Property(s => s.SupplierName)

.IsRequired()

.HasMaxLength(100);

modelBuilder.Entity<Product>()

.Property(p => p.ProductName)

.IsRequired()

.HasMaxLength(100);

modelBuilder.Entity<Product>()

.HasOne(p => p.Category)

.WithMany(c => c.Products)

.HasForeignKey(p => p.CategoryId);

modelBuilder.Entity<Product>()

.HasOne(p => p.Supplier)

.WithMany(s => s.Products)

.HasForeignKey(p => p.SupplierId);

}

}

}

**Program.cs:**

using RetailInventory.Data;

using RetailInventory.Models;

using Microsoft.EntityFrameworkCore;

using System;

using System.Collections.Generic;

using System.Linq;

namespace RetailInventory

{

class Program

{

static void Main()

{

using var context = new RetailDbContext();

context.Database.EnsureCreated();

if (!context.Products.Any())

{

SeedData(context);

}

var inventory = context.Products

.Include(p => p.Category)

.Include(p => p.Supplier)

.ToList();

Console.WriteLine("\nRetail Inventory:\n");

foreach (var item in inventory)

{

Console.WriteLine($"{item.ProductName,-20} | ₹{item.Price,8} | Stock: {item.Stock,3} | Category: {item.Category.CategoryName} | Supplier: {item.Supplier.SupplierName}");

}

}

static void SeedData(RetailDbContext context)

{

var cat1 = new Category { CategoryName = "Electronics" };

var cat2 = new Category { CategoryName = "Groceries" };

var sup1 = new Supplier { SupplierName = "ElectroMax", ContactEmail = "contact@electromax.com" };

var sup2 = new Supplier { SupplierName = "DailyNeeds", ContactEmail = "hello@dailyneeds.com" };

var products = new List<Product>

{

new Product { ProductName = "Bluetooth Speaker", Price = 2500, Stock = 10, Category = cat1, Supplier = sup1 },

new Product { ProductName = "LED TV", Price = 40000, Stock = 5, Category = cat1, Supplier = sup1 },

new Product { ProductName = "Rice Bag 10kg", Price = 800, Stock = 20, Category = cat2, Supplier = sup2 }

};

context.Categories.AddRange(cat1, cat2);

context.Suppliers.AddRange(sup1, sup2);

context.Products.AddRange(products);

context.SaveChanges();

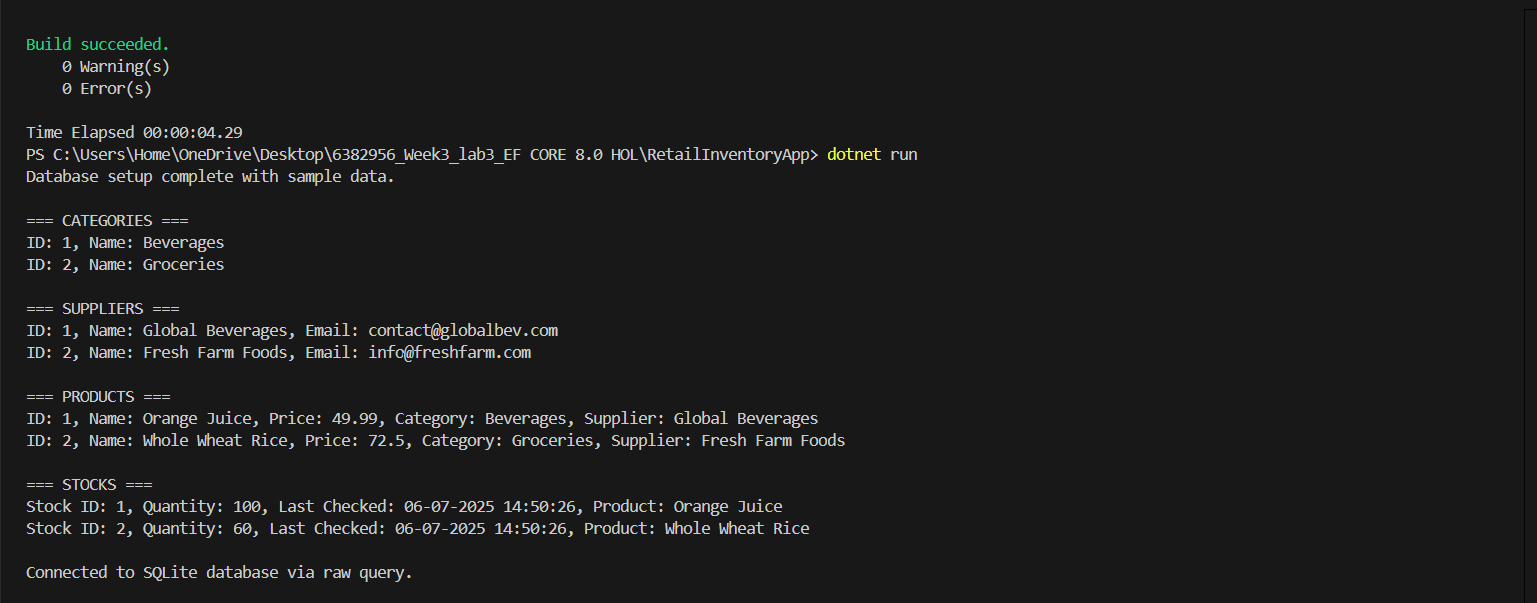
Console.WriteLine("\n Sample data inserted.\n");

}

}

}

**Output:**

****