Lab 3:Using EF Core CLI to Create and Apply Migrations

**Product.cs:**

namespace RetailInventorySimple.Models

{

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; }

}

}

**Category.cs:**

using System.Collections.Generic;

namespace RetailInventorySimple.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

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

}

}

**AppDbContext.cs:**

using Microsoft.EntityFrameworkCore;

using RetailInventorySimple.Models;

namespace RetailInventorySimple.Data

{

public class AppDbContext : DbContext

{

public DbSet<Product> Products { get; set; } = null!;

public DbSet<Category> Categories { get; set; } = null!;

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

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

}

}

}

**Program.cs:**

using System;

using System.Linq;

using Microsoft.EntityFrameworkCore;

using RetailInventorySimple.Data;

using RetailInventorySimple.Models;

class Program

{

static void Main()

{

using var context = new AppDbContext();

if (!context.Products.Any())

{

var category = new Category { Name = "Books" };

category.Products.Add(new Product { Name = "C# Guide", Price = 25m });

category.Products.Add(new Product { Name = "EF Core Tutorial", Price = 30m });

context.Categories.Add(category);

context.SaveChanges();

}

var products = context.Products.Include(p => p.Category).ToList();

foreach (var product in products)

{

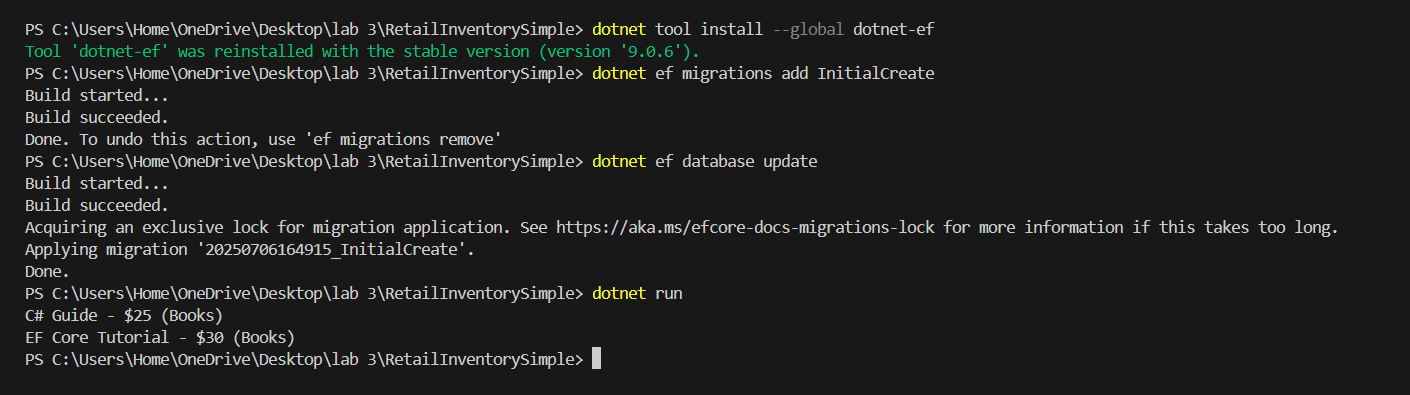
Console.WriteLine($"{product.Name} - ${product.Price} ({product.Category?.Name})");

}

}

}

**Output:**

****