**Week\_3 -Entity Framework**

The OUTPUTS of Labs 1 -5 are displayed below followed by the final codes.

*Project Structure:-*

RetailInventory

│

├── AppDbContext.cs

├── Program.cs

├── appsettings.json

├── RetailInventory.csproj

│

├── Models

│ ├── Category.cs

│ └── Product.cs

│

├── Helpers

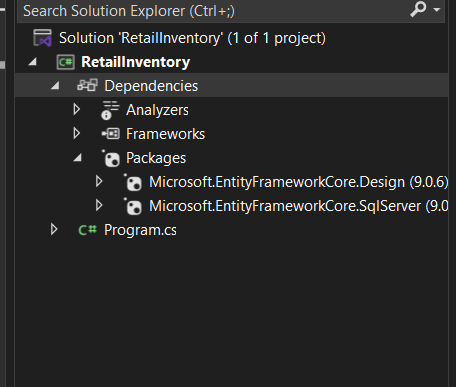
│ └── ConfigurationHelper.cs

│

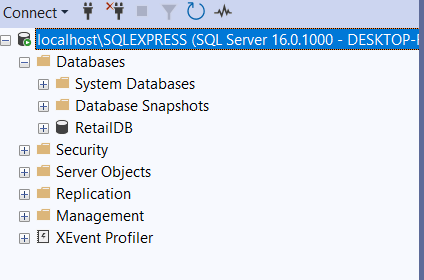
└── Migrations

└── (Auto-generated EF migration files )

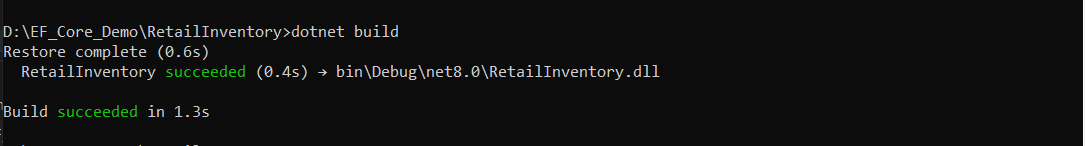
**Lab 1 Output :-**

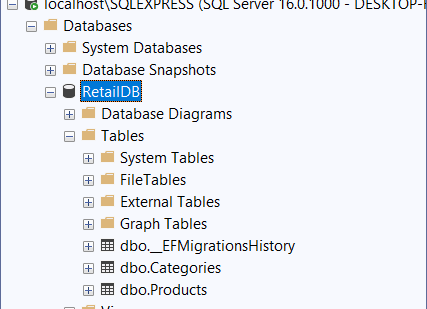
****

**Lab 2 – Output:-**

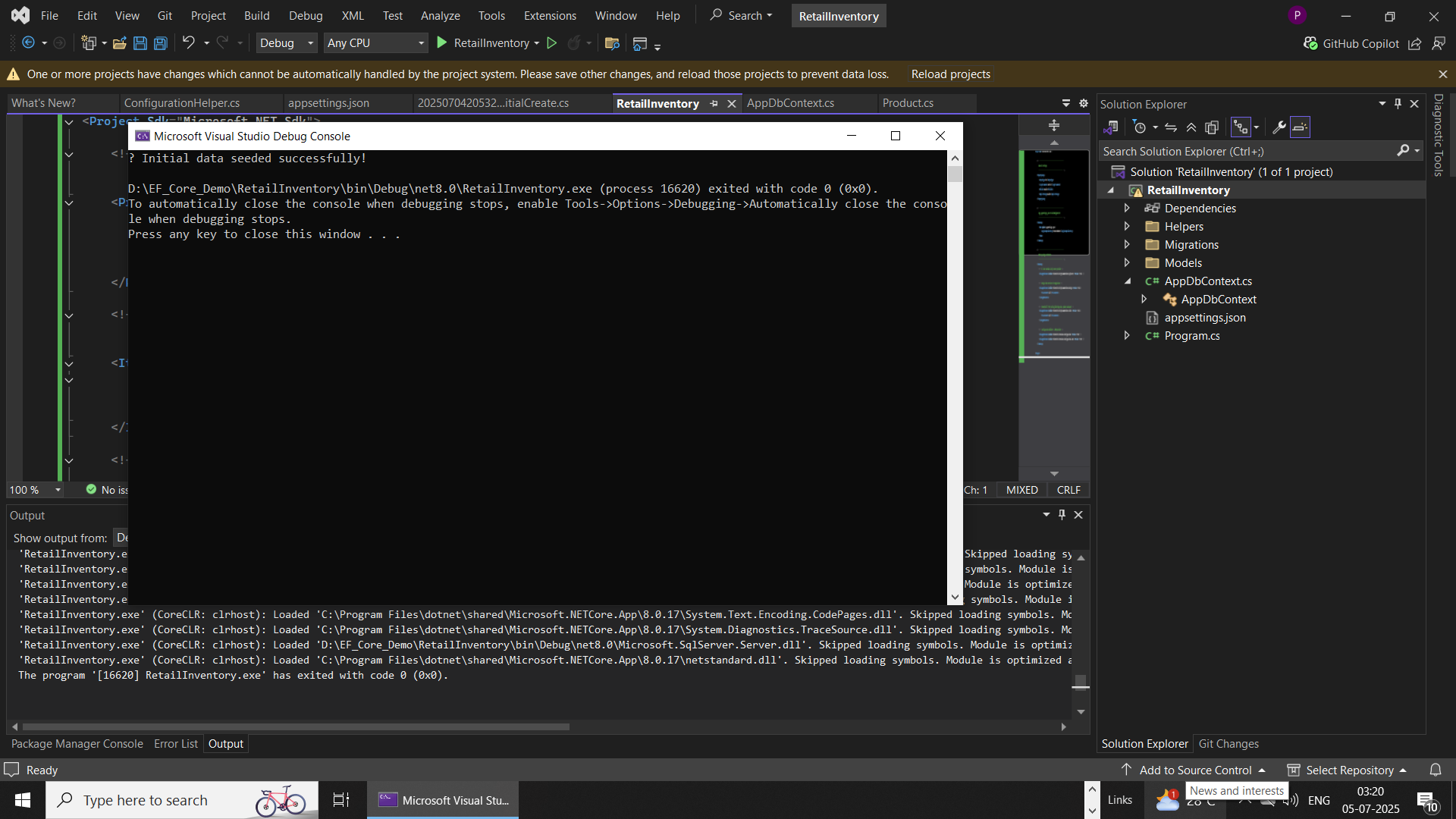
****

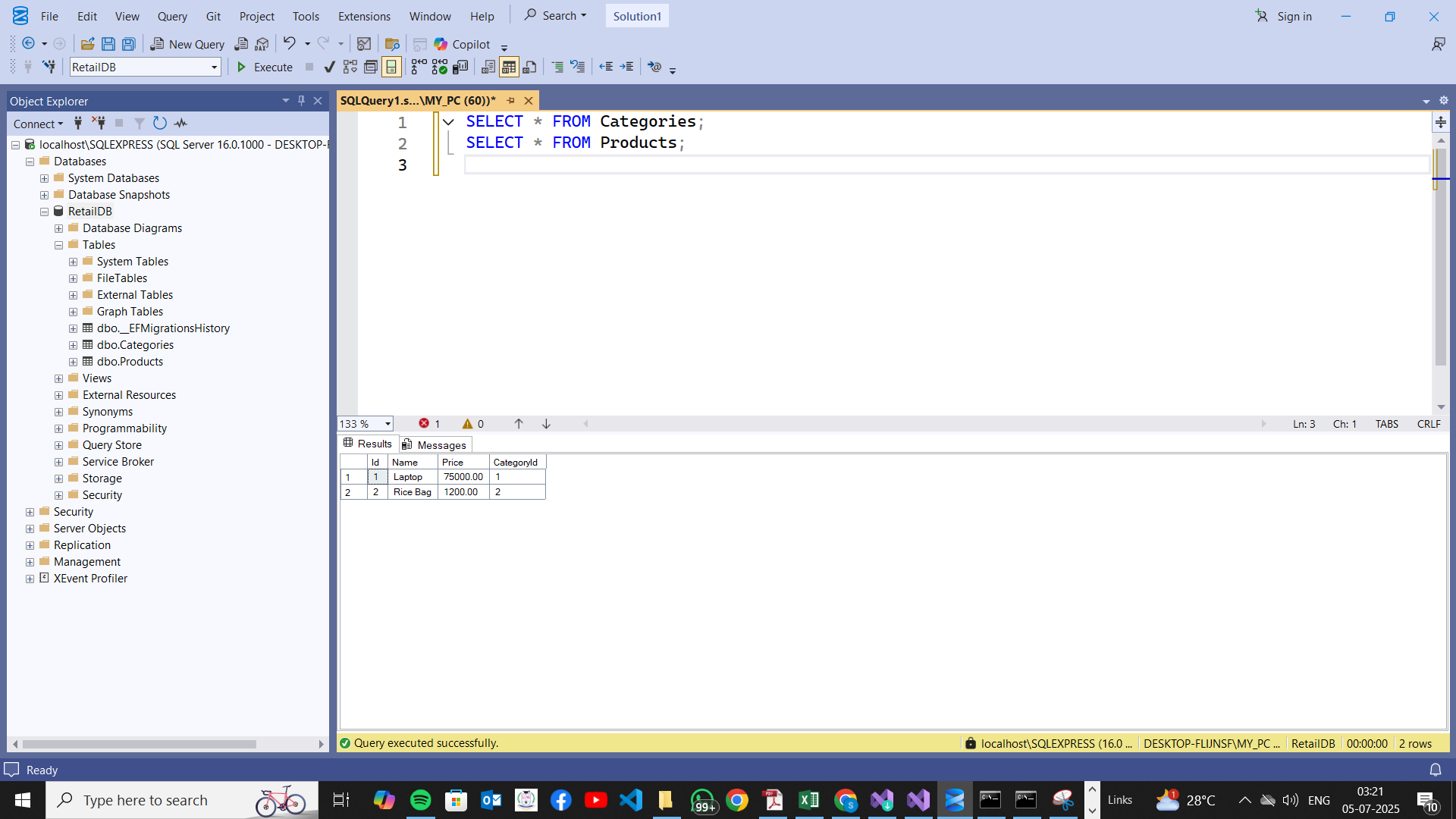
**Lab 3 – Output:-**

****

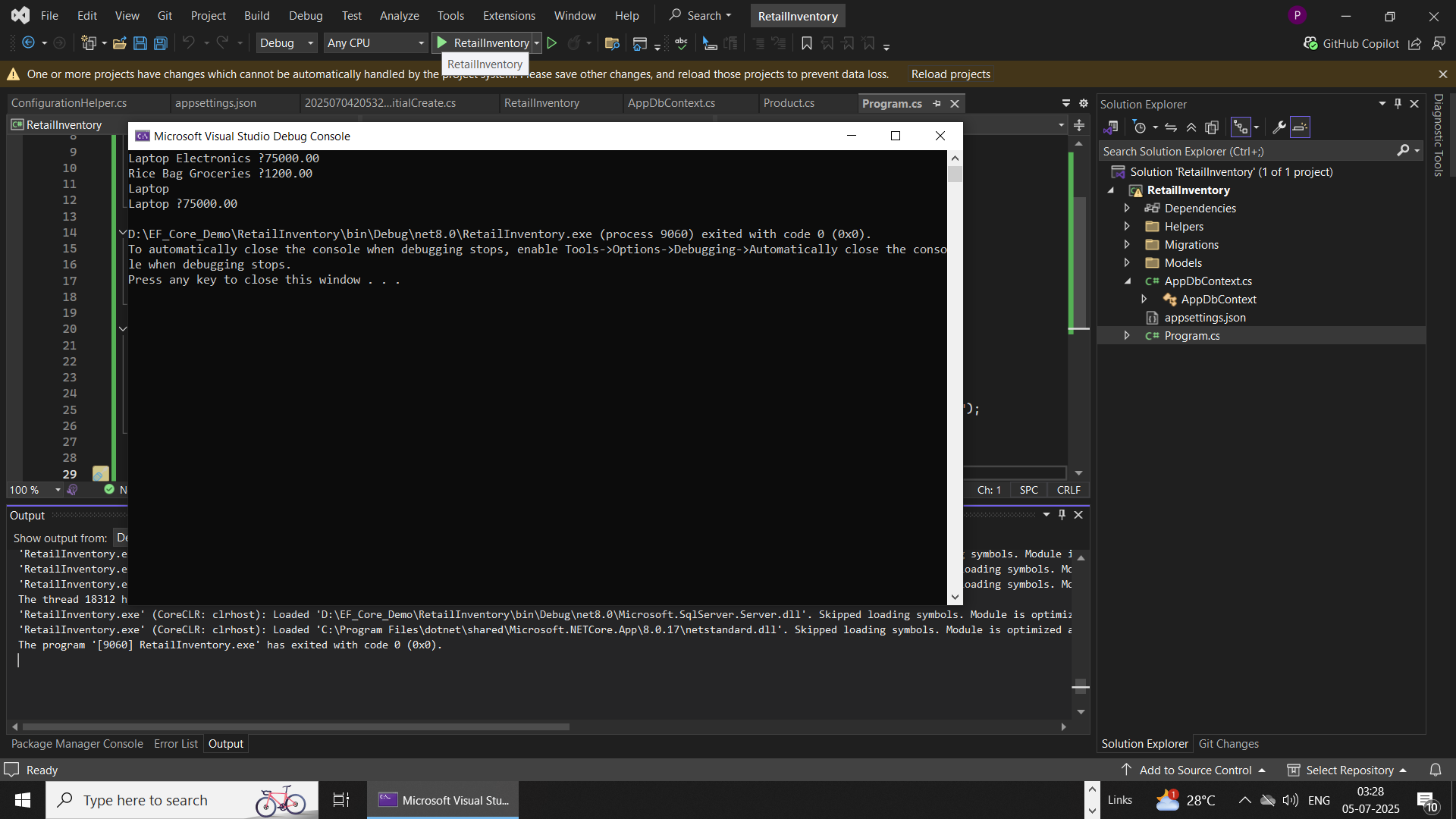
****

**Lab – 4 Output:-**

****

****

**Lab 5 Output :-**

****

***The different codes involved are as follows:-***

***(refer the project structure for better understanding)***

*Product.cs :-*

namespace RetailInventory.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 RetailInventory.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

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

}

}

*ConfigurationHelper.cs:-*

using System;

using System.IO;

using Microsoft.Extensions.Configuration;

namespace RetailInventory.Helpers

{

public static class ConfigurationHelper

{

public static string GetConnectionString()

{

var config = new ConfigurationBuilder()

.SetBasePath(Directory.GetCurrentDirectory())

.AddJsonFile("appsettings.json", optional: false, reloadOnChange: true)

.Build();

return config.GetConnectionString("DefaultConnection");

}

}

}

*AppDbContext.cs:-*

using Microsoft.EntityFrameworkCore;

using RetailInventory.Models;

using RetailInventory.Helpers;

namespace RetailInventory

{

public class AppDbContext : DbContext

{

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

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

if (!optionsBuilder.IsConfigured)

{

string conn = ConfigurationHelper.GetConnectionString();

optionsBuilder.UseSqlServer(conn);

}

}

protected override void OnModelCreating(ModelBuilder modelBuilder)

{

modelBuilder.Entity<Category>()

.HasMany(c => c.Products)

.WithOne(p => p.Category)

.HasForeignKey(p => p.CategoryId)

.OnDelete(DeleteBehavior.Cascade);

}

}

}

*Appsettings.json:-*

{

"ConnectionStrings": {

"DefaultConnection": "Server=localhost\\SQLEXPRESS;Database=RetailDB;Trusted\_Connection=True;Encrypt=False;"

}

}

*Migrations -> AppDbContextModelSnapshot.cs*

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Infrastructure;

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

#nullable disable

namespace RetailInventory.Migrations

{

[DbContext(typeof(AppDbContext))]

partial class AppDbContextModelSnapshot : ModelSnapshot

{

protected override void BuildModel(ModelBuilder modelBuilder)

{

#pragma warning disable 612, 618

modelBuilder

.HasAnnotation("ProductVersion", "9.0.6")

.HasAnnotation("Relational:MaxIdentifierLength", 128);

SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

modelBuilder.Entity("RetailInventory.Models.Category", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<string>("Name")

.IsRequired()

.HasColumnType("nvarchar(max)");

b.HasKey("Id");

b.ToTable("Categories");

});

modelBuilder.Entity("RetailInventory.Models.Product", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<int>("CategoryId")

.HasColumnType("int");

b.Property<string>("Name")

.IsRequired()

.HasColumnType("nvarchar(max)");

b.Property<decimal>("Price")

.HasColumnType("decimal(18,2)");

b.HasKey("Id");

b.HasIndex("CategoryId");

b.ToTable("Products");

});

modelBuilder.Entity("RetailInventory.Models.Product", b =>

{

b.HasOne("RetailInventory.Models.Category", "Category")

.WithMany("Products")

.HasForeignKey("CategoryId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

b.Navigation("Category");

});

modelBuilder.Entity("RetailInventory.Models.Category", b =>

{

b.Navigation("Products");

});

#pragma warning restore 612, 618

}

}

}

*Migrations :- 20250704205324\_InitialCreate.cs*

using Microsoft.EntityFrameworkCore.Migrations;

#nullable disable

namespace RetailInventory.Migrations

{

/// <inheritdoc />

public partial class InitialCreate : Migration

{

/// <inheritdoc />

protected override void Up(MigrationBuilder migrationBuilder)

{

migrationBuilder.CreateTable(

name: "Categories",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

Name = table.Column<string>(type: "nvarchar(max)", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_Categories", x => x.Id);

});

migrationBuilder.CreateTable(

name: "Products",

columns: table => new

{

Id = table.Column<int>(type: "int", nullable: false)

.Annotation("SqlServer:Identity", "1, 1"),

Name = table.Column<string>(type: "nvarchar(max)", nullable: false),

Price = table.Column<decimal>(type: "decimal(18,2)", nullable: false),

CategoryId = table.Column<int>(type: "int", nullable: false)

},

constraints: table =>

{

table.PrimaryKey("PK\_Products", x => x.Id);

table.ForeignKey(

name: "FK\_Products\_Categories\_CategoryId",

column: x => x.CategoryId,

principalTable: "Categories",

principalColumn: "Id",

onDelete: ReferentialAction.Cascade);

});

migrationBuilder.CreateIndex(

name: "IX\_Products\_CategoryId",

table: "Products",

column: "CategoryId");

}

/// <inheritdoc />

protected override void Down(MigrationBuilder migrationBuilder)

{

migrationBuilder.DropTable(

name: "Products");

migrationBuilder.DropTable(

name: "Categories");

}

}

}

*Program.cs :-*

using Microsoft.EntityFrameworkCore;

using RetailInventory;

await using (var context = new AppDbContext())

{

var products = await context.Products

.Include(p => p.Category)

.ToListAsync();

foreach (var p in products)

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

}

await using (var context = new AppDbContext())

{

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

Console.WriteLine(product is not null ? $"{product.Name}" : "Product not found");

}

await using (var context = new AppDbContext())

{

var expensive = await context.Products

.FirstOrDefaultAsync(p => p.Price > 50000);

Console.WriteLine(expensive is not null ? $"{expensive.Name} ₹{expensive.Price}" : "No expensive product found");

}

Console.ReadKey();