**Entity Framework Core 8.0**

Lab 1: Understanding ORM with a Retail Inventory System

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

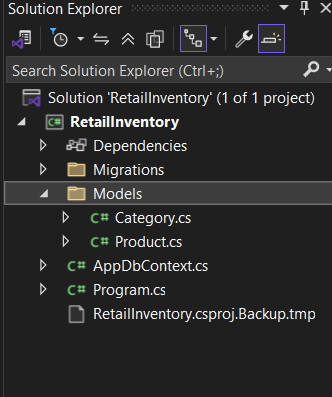
Lab 3: Using EF Core CLI to Create and Apply Migrations

Lab 4: Inserting Initial Data into the Database

Lab 5: Retrieving Data from the Database

***Solution***

***//File Structure:-***



***//Models/Category.cs***

namespace RetailInventory.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

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

}

}

***//Models/Product.cs***

using System.Collections.Generic;

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; }

}

}

***//AppDbContext.cs***

using Microsoft.EntityFrameworkCore;

using RetailInventory.Models;

using System.Collections.Generic;

namespace RetailInventory

{

public class AppDbContext : DbContext

{

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

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

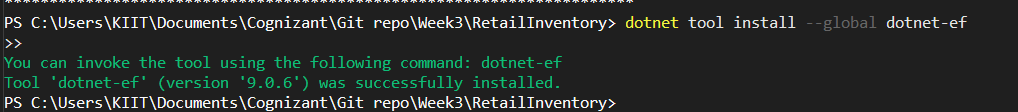
optionsBuilder.UseSqlServer("Server=(localdb)\\mssqllocaldb;Database=RetailDbLab2;Trusted\_Connection=True;");

}

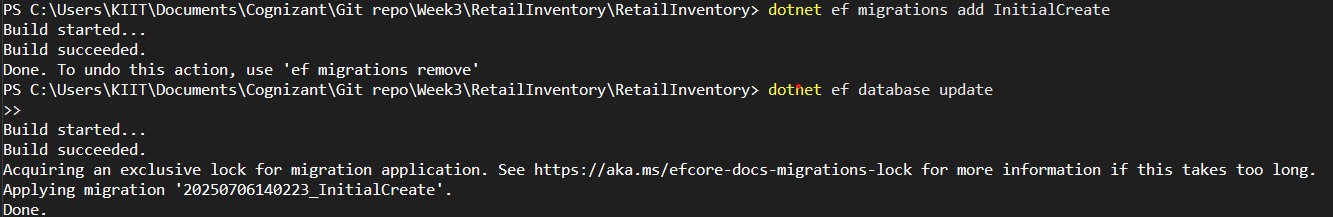
}

}

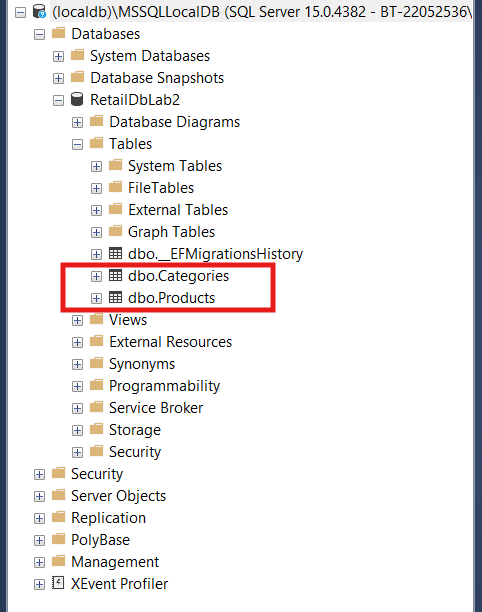
1. ***Install EF Core CLI***



1. ***Create Initial Migration and Apply Migration to Create Database***



1. ***Verify in SQL Server:***



***Insert Data in Program.cs:***

***//Program.cs***

using RetailInventory;

using RetailInventory.Models;

using System.Threading.Tasks;

class Program

{

static async Task Main(string[] args)

{

using var context = new AppDbContext();

if (!context.Categories.Any())

{

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();

Console.WriteLine("Initial data inserted successfully!");

}

else

{

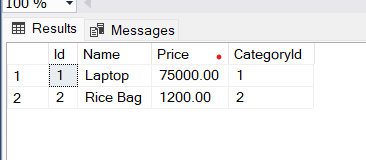
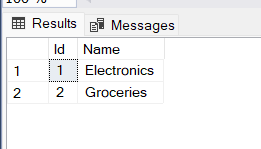
Console.WriteLine("Data already exists. Skipping insert.");

}

}

}

***Verify in SQL Server:***



***Retrieving Data from the Database***

using Microsoft.EntityFrameworkCore;

using RetailInventory;

using RetailInventory.Models;

class Program

{

static async Task Main(string[] args)

{

using var context = new AppDbContext();

// 1. Retrieve All Products

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

Console.WriteLine(" All Products:");

foreach (var p in products)

{

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

}

Console.WriteLine("\n Find Product by ID:");

// 2. Find Product by ID

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

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

Console.WriteLine("\n Find First Expensive Product:");

// 3. FirstOrDefault with Condition

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

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

}

}

