**Week-3**

**EF Core 8.0 Guided Hands-On Exercises**

**Lab 1:**

**1. What is ORM?**

**ORM (Object-Relational Mapping) is a technique that connects programming language objects (like C# classes) with database tables.**

* **C# Class: Product**
* **Table in SQL: Products**
* **ORM maps: Product.Name ➝ Products.Name column**

### **Benefits:**

* **Productivity: No SQL required for basic operations.**
* **Maintainability: Strong typing and refactoring support.**
* **Abstraction: You think in terms of C# classes, not raw SQL.**

**Lab-2**

Category.cs

**using System.Collections.Generic;**

**namespace RetailInventory.Models;**

**public class Category**

**{**

**public int Id { get; set; }**

**public required string Name { get; set; }**

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

**}**

[**Product.cs**](http://product.cs)

**namespace RetailInventory.Models;**

**public class Product**

**{**

**public int Id { get; set; }**

**public required string Name { get; set; }**

**public decimal Price { get; set; }**

**public int CategoryId { get; set; }**

**public required Category Category { get; set; }**

**}**

**AppContextDb**

**using Microsoft.EntityFrameworkCore;**

**using RetailInventory.Models;**

**public class AppDbContext : DbContext**

**{**

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

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

**protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)**

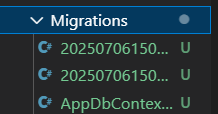
**{**

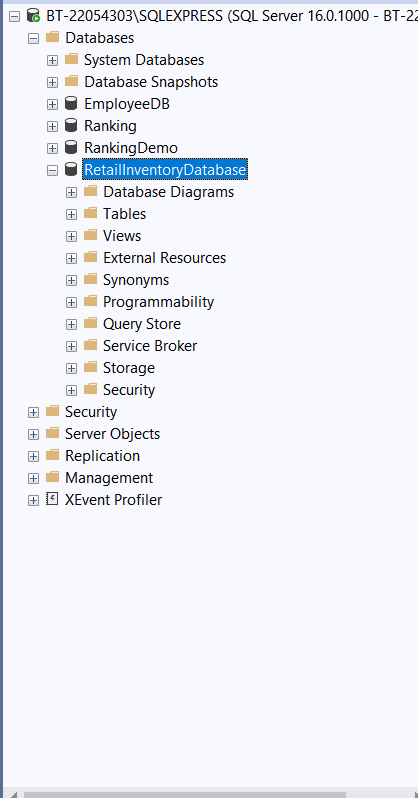
**optionsBuilder.UseSqlServer("Server=BT-22054303\\SQLEXPRESS;Database=RetailInventoryDatabase;Trusted\_Connection=True;TrustServerCertificate=True;");**

**}**

**}**

**Lab-3**

****

****

**Lab-4**

[Program.cs](http://program.cs)

using System;

using System.Threading.Tasks;

class Program {

static async Task Main(string[] args) {

using var context = new AppDbContext();

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 };

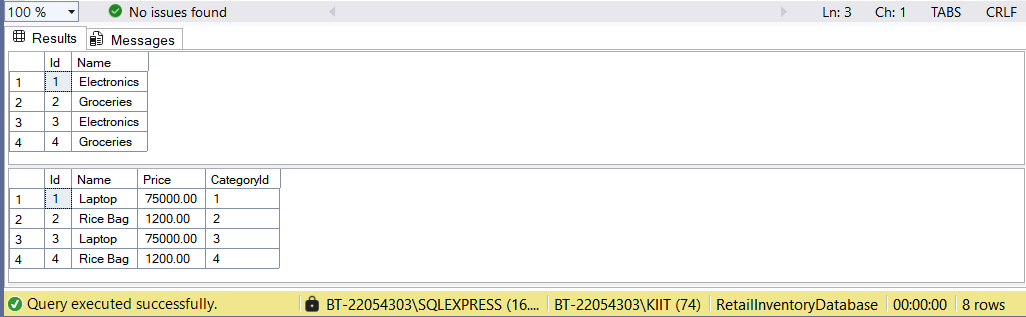
await context.Products.AddRangeAsync(product1, product2);

await context.SaveChangesAsync();

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

}

}



Lab-5

[program.cs](http://program.cs)

using System;

using System.Threading.Tasks;

using Microsoft.EntityFrameworkCore;

using RetailInventory;

class Program

{

static async Task Main()

{

using var context = new AppDbContext();

Console.WriteLine("📦 All Products:");

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

foreach (var p in products)

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

Console.WriteLine("\n🔍 Find Product with ID = 1:");

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

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

Console.WriteLine("\n💸 First Expensive Product (> ₹50000):");

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

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

}

}

