**Ashutosh Dash  
Superset ID – 6365404  
EF Core 8.0 Guided Hands-On Exercises**

**LAB – 1**

**Employee.cs**

namespace EFCoreLab1.Models

{

public class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public string Department { get; set; }

}

}

**AppDbContext.cs**

using EFCoreLab1.Models;

using Microsoft.EntityFrameworkCore;

using System.Collections.Generic;

namespace EFCoreLab1

{

public class AppDbContext : DbContext

{

public DbSet<Employee> Employees { get; set; }

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

{

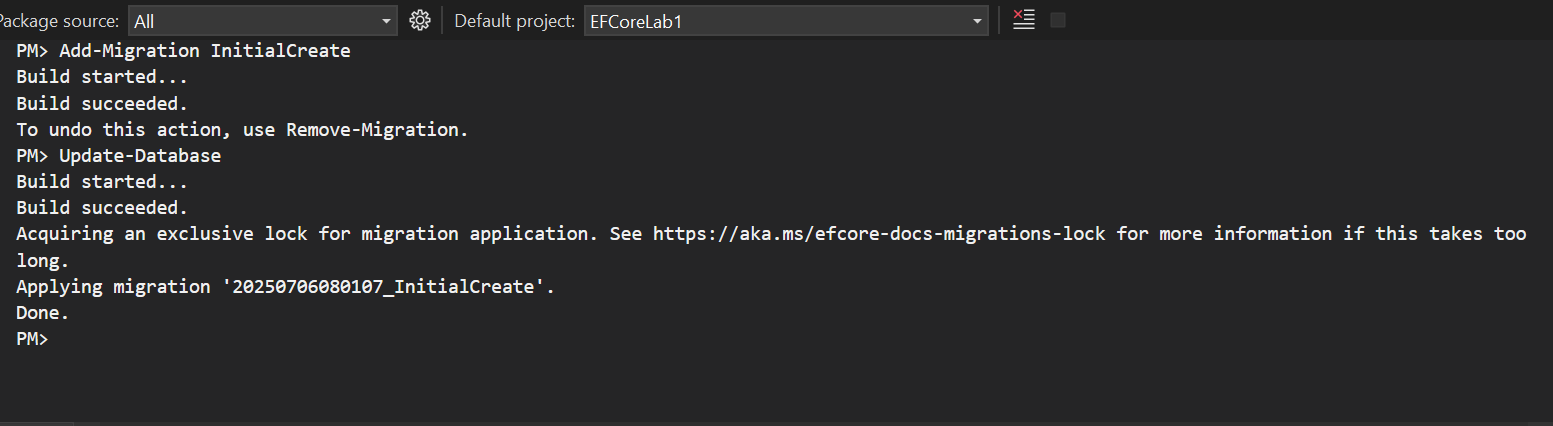
optionsBuilder.UseSqlServer("Server=localhost;Database=EFCoreLabDb;Trusted\_Connection=True;TrustServerCertificate=True;");

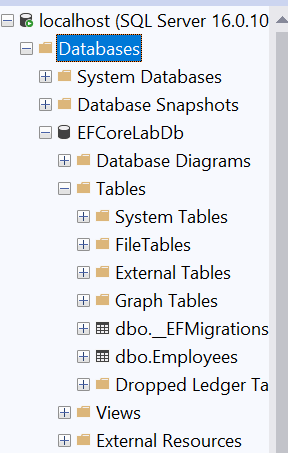
}

}

}

**OUTPUT**





**LAB – 2**

**Category.cs**

using System.Collections.Generic;

namespace EFCoreLab1.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; }

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

}

}

**Product.cs**

namespace EFCoreLab1.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 EFCoreLab1.Models;

using Microsoft.EntityFrameworkCore;

namespace EFCoreLab1

{

public class AppDbContext : DbContext

{

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

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

protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

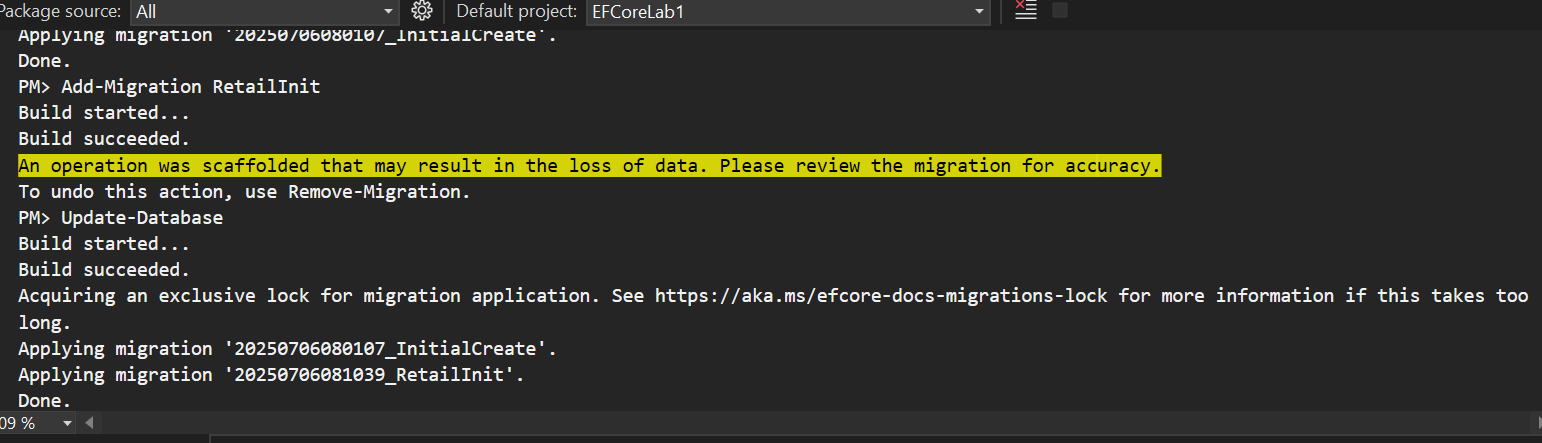
{

optionsBuilder.UseSqlServer("Server=localhost;Database=RetailDB;Trusted\_Connection=True;TrustServerCertificate=True;");

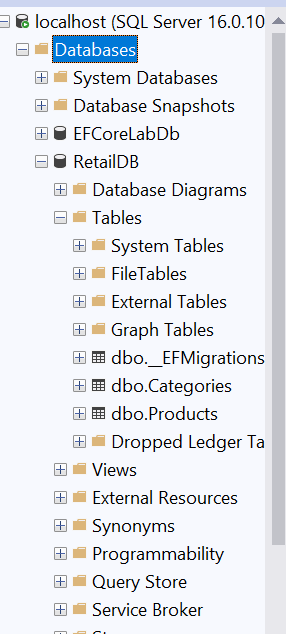
}

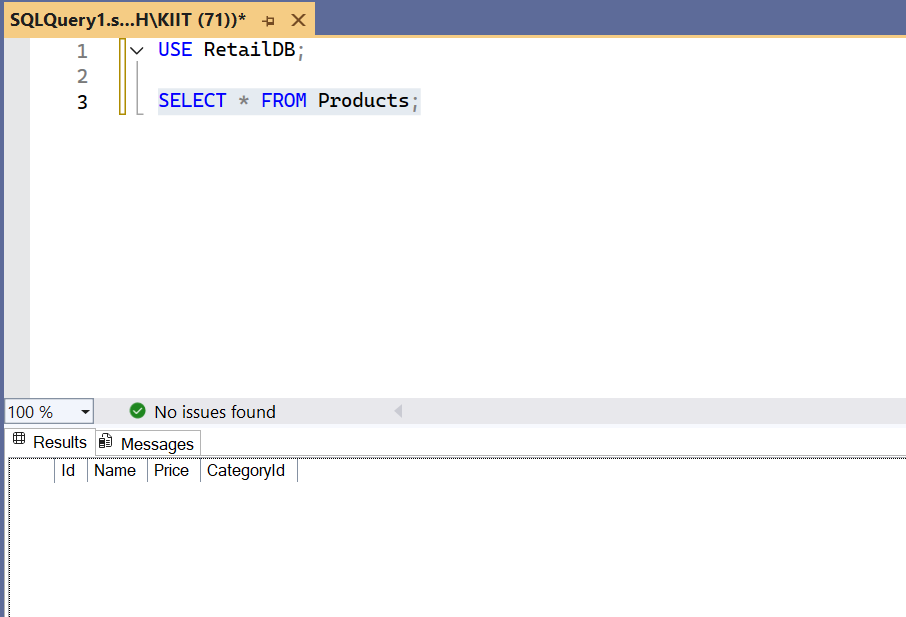
}

}



**LAB – 3**





**LAB – 4**

**Program.cs**

using EFCoreLab1;

using EFCoreLab1.Models;

var context = new AppDbContext();

var electronics = new Category { Name = "Electronics" };

var groceries = new Category { Name = "Groceries" };

await context.Categories.AddRangeAsync(electronics, groceries);

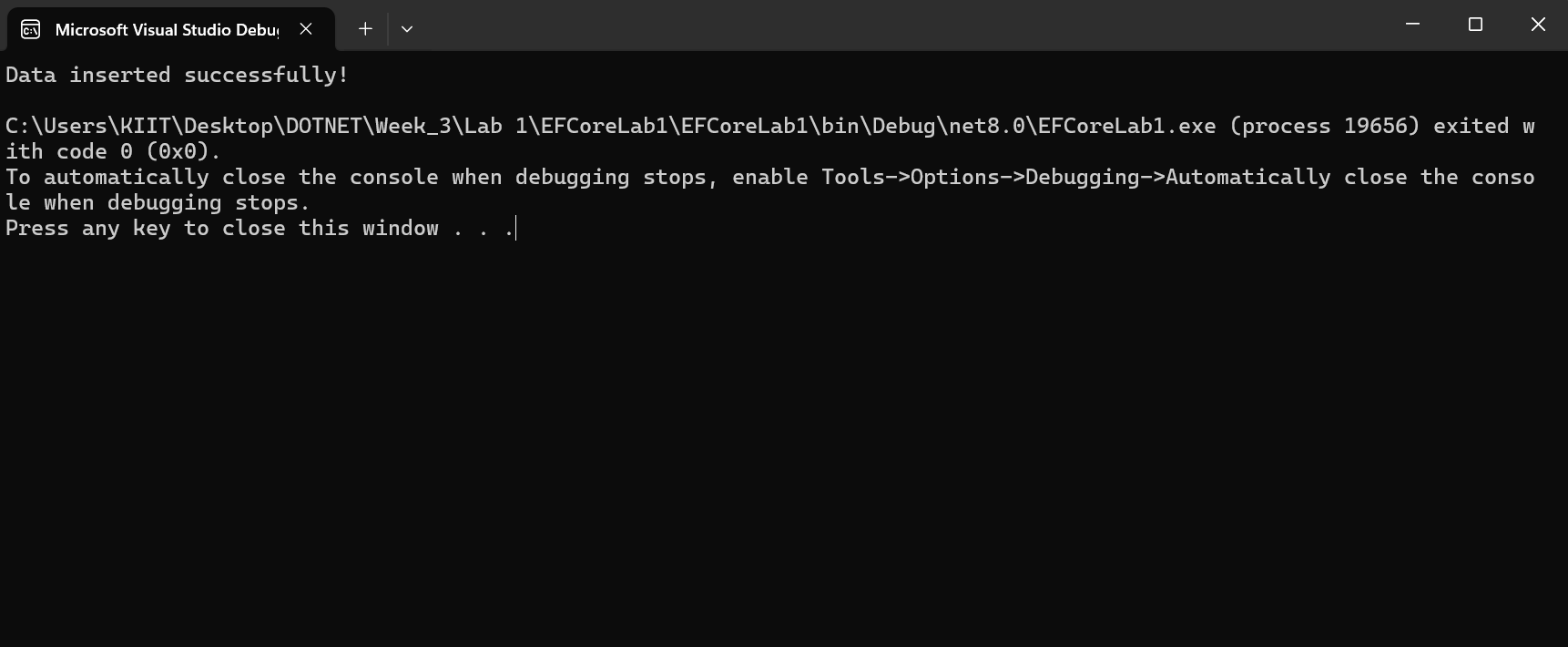
var product1 = new Product { Name = "Laptop", Price = 75000, Category = electronics };

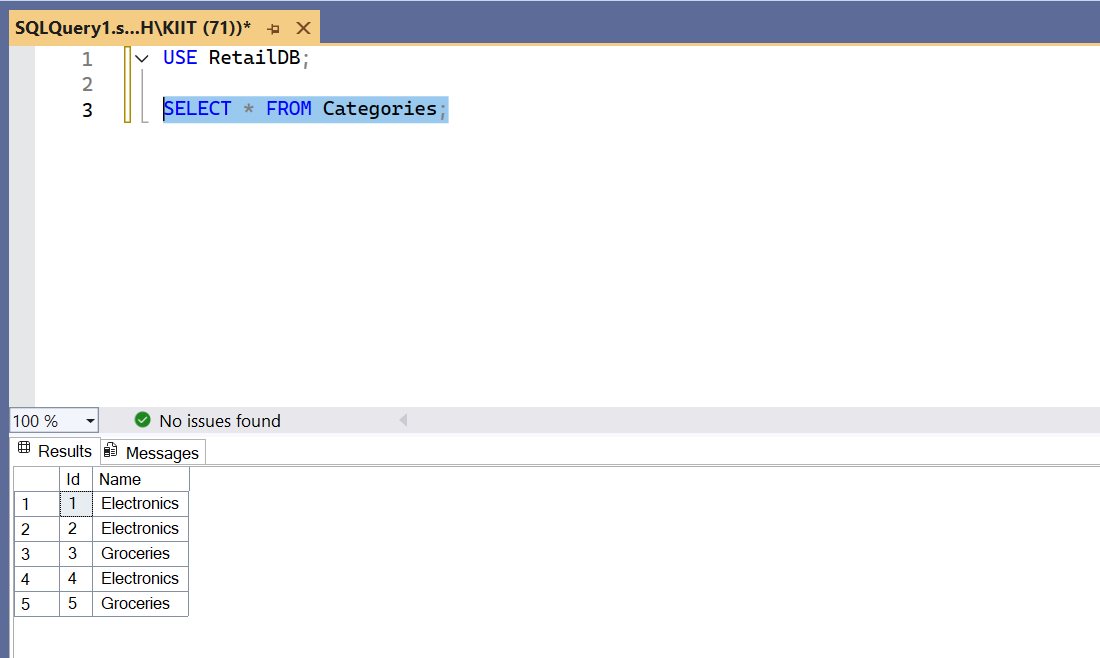
var product2 = new Product { Name = "Rice Bag", Price = 1200, Category = groceries };

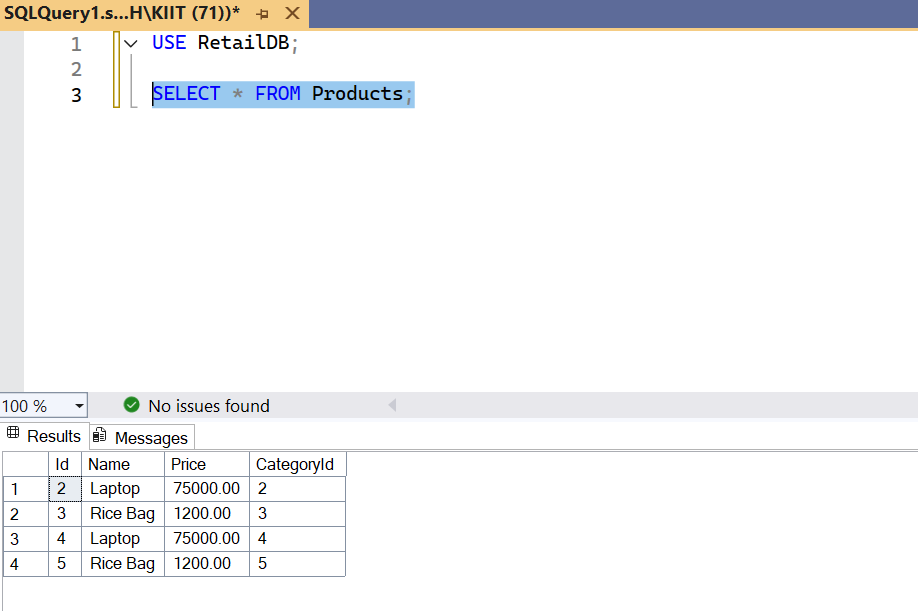
await context.Products.AddRangeAsync(product1, product2);

await context.SaveChangesAsync();

Console.WriteLine("Data inserted successfully!");







**LAB – 5**

**Program.cs**

using EFCoreLab1;

using EFCoreLab1.Models;

using Microsoft.EntityFrameworkCore;

var context = new AppDbContext();

// 1. Retrieve All Products

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

Console.WriteLine("\nAll Products:");

foreach (var p in products)

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

// 2. Find by ID

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

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

// 3. FirstOrDefault with condition

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

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

