Lab 5: Retrieving Data from the Database

// Required NuGet Packages:

// - Microsoft.EntityFrameworkCore

// - Microsoft.EntityFrameworkCore.Sqlite

// - Microsoft.EntityFrameworkCore.Tools

using System;

using System.Collections.Generic;

using System.Linq;

using Microsoft.EntityFrameworkCore;

namespace RetailInventoryApp

{

// Model: Category

public class Category

{

public int CategoryId { get; set; }

public string Name { get; set; }

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

}

// Model: Product

public class Product

{

public int ProductId { get; set; }

public string Name { get; set; }

public int Quantity { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; }

}

// DbContext

public class RetailDbContext : DbContext

{

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

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

protected override void OnConfiguring(DbContextOptionsBuilder options)

{

options.UseSqlite("Data Source=retail.db");

}

}

class Program

{

static void Main(string[] args)

{

using var db = new RetailDbContext();

Console.WriteLine("📦 All Products:");

var allProducts = db.Products.Include(p => p.Category).ToList();

foreach (var product in allProducts)

{

Console.WriteLine($"- {product.Name} (Qty: {product.Quantity}) | Category: {product.Category.Name}");

}

Console.WriteLine("\n🔎 Filtered Products (Quantity > 50):");

var filtered = db.Products

.Where(p => p.Quantity > 50)

.OrderByDescending(p => p.Quantity)

.Include(p => p.Category)

.ToList();

foreach (var product in filtered)

{

Console.WriteLine($"- {product.Name} (Qty: {product.Quantity}) | Category: {product.Category.Name}");

}

Console.WriteLine("\n📁 Categories with Products:");

var categories = db.Categories.Include(c => c.Products).ToList();

foreach (var cat in categories)

{

Console.WriteLine($"\n📂 {cat.Name}:");

foreach (var p in cat.Products)

{

Console.WriteLine($" - {p.Name} (Qty: {p.Quantity})");

}

}

}

}

}

Out put :

