Lab 1: Understanding ORM with a Retail Inventory System

a)Category.cs :

namespace RetailInventoryApp.Models

{

    public class Category

    {

        public int CategoryId { get; set; }

        public string Name { get; set; }

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

    }

}

b)Product.cs :

using System.ComponentModel.DataAnnotations.Schema;

namespace RetailInventoryApp.Models

{

    public class Product

    {

        public int ProductId { get; set; }

        public string Name { get; set; }

        public int Quantity { get; set; }

        public decimal Price { get; set; }

        // Foreign key

        public int CategoryId { get; set; }

        // Navigation property

        public Category Category { get; set; }

    }

}

c)Program.cs:

using RetailInventoryApp.Models;

class Program

{

    static void Main()

    {

        using (var context = new RetailContext())

        {

            // CREATE

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

            context.Categories.Add(category);

            context.SaveChanges();

            var product = new Product

            {

                Name = "Smartphone",

                Quantity = 50,

                Price = 29999,

                CategoryId = category.CategoryId

            };

            context.Products.Add(product);

            context.SaveChanges();

            // READ

            Console.WriteLine("Products:");

            foreach (var p in context.Products)

            {

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

            }

            // UPDATE

            product.Quantity += 10;

            context.SaveChanges();

            // DELETE

            context.Products.Remove(product);

            context.SaveChanges();

        }

    }

}

d)RetailContext.cs :

using Microsoft.EntityFrameworkCore;

using RetailInventoryApp.Models;

public class RetailContext : DbContext

{

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

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

    protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)

    {

        optionsBuilder.UseSqlite("Data Source=retailinventory.db");

    }

}

Output:

