**WEEK 3**

**Entity Framework Core 8.0**

**Lab 1: Understanding ORM with a Retail Inventory System**

**CODE**

**PROGRAM.CS**

using System;

class Category

{

public int Id { get; set; }

public string Name { get; set; }

}

class Product

{

public int Id { get; set; }

public string Name { get; set; }

public decimal Price { get; set; }

public Category Category { get; set; }

}

class Program

{

static void Main(string[] args)

}

var category = new Category { Id = 1, Name = "Electronics" };

var product = new Product { Id = 101, Name = "Laptop", Price = 75000, Category = category };

Console.WriteLine("EF Core 8.0 Lab 1: ORM Mapping Example");

Console.WriteLine("--------------------------------------");

Console.WriteLine($"Category ID: {category.Id}, Name: {category.Name}");

Console.WriteLine($"Product ID: {product.Id}, Name: {product.Name}");

Console.WriteLine($"Price: ₹{product.Price}");

Console.WriteLine($"Belongs to Category: {product.Category.Name}");

}

}

**RetailInventory.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

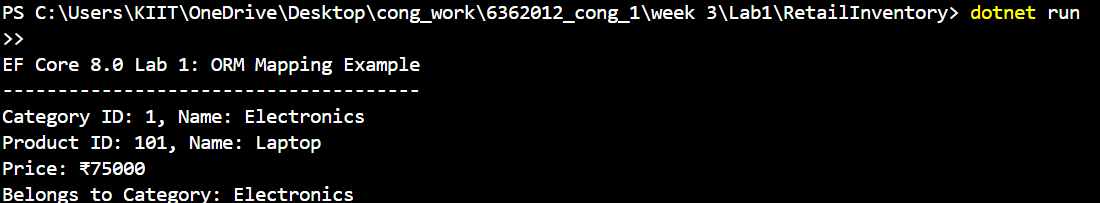
<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

</Project>

**OUTPUT:**



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

**CODE:**

**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; }

}

}

**MODELS/Product.cs**

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; }

}

}

**RetailInvestory/AppDbContext.cs**

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=localhost;Database=RetailInventoryDB;Trusted\_Connection=True;");

}

}

**Program.cs**

using System;

using System.Collections.Generic;

using RetailInventory.Models;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EF Core 8.0 Lab 2: DbContext & Models Example");

Console.WriteLine("-----------------------------------------------");

// Create dummy data to show model relationships

var electronics = new Category

{

Id = 1,

Name = "Electronics",

Products = new List<Product>

{

new Product { Id = 101, Name = "Laptop", Price = 75000 },

new Product { Id = 102, Name = "Smartphone", Price = 50000 }

}

};

var groceries = new Category

{

Id = 2,

Name = "Groceries",

Products = new List<Product>

{

new Product { Id = 201, Name = "Rice Bag", Price = 1200 },

new Product { Id = 202, Name = "Wheat Flour", Price = 800 }

}

};

// Print Categories and Products

PrintCategory(electronics);

PrintCategory(groceries);

}

static void PrintCategory(Category category)

{

Console.WriteLine($"Category: {category.Name}");

foreach (var product in category.Products)

{

Console.WriteLine($" Product: {product.Name}, Price: ₹{product.Price}");

}

Console.WriteLine();

}

}

**RetailInvestory.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="9.0.6">

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

<PrivateAssets>all</PrivateAssets>

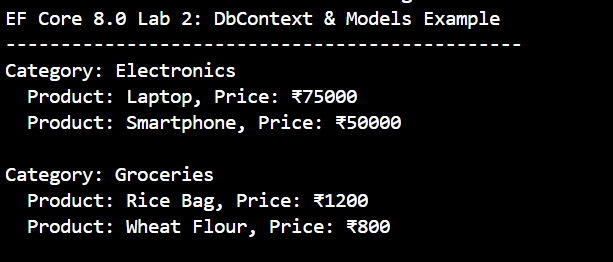
</PackageReference>

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.6" />

</ItemGroup>

</Project>

**OUTPUT:**

****

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

**CODE:**

**Migration/AppDbContextModelSnapshot.cs**

// <auto-generated />

using Microsoft.EntityFrameworkCore;

using Microsoft.EntityFrameworkCore.Infrastructure;

using Microsoft.EntityFrameworkCore.Metadata;

using Microsoft.EntityFrameworkCore.Storage.ValueConversion;

#nullable disable

namespace RetailInventory.Migrations

{

[DbContext(typeof(AppDbContext))]

partial class AppDbContextModelSnapshot : ModelSnapshot

{

protected override void BuildModel(ModelBuilder modelBuilder)

{

#pragma warning disable 612, 618

modelBuilder

.HasAnnotation("ProductVersion", "9.0.6")

.HasAnnotation("Relational:MaxIdentifierLength", 128);

SqlServerModelBuilderExtensions.UseIdentityColumns(modelBuilder);

modelBuilder.Entity("RetailInventory.Models.Category", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<string>("Name")

.IsRequired()

.HasColumnType("nvarchar(max)");

b.HasKey("Id");

b.ToTable("Categories");

});

modelBuilder.Entity("RetailInventory.Models.Product", b =>

{

b.Property<int>("Id")

.ValueGeneratedOnAdd()

.HasColumnType("int");

SqlServerPropertyBuilderExtensions.UseIdentityColumn(b.Property<int>("Id"));

b.Property<int>("CategoryId")

.HasColumnType("int");

b.Property<string>("Name")

.IsRequired()

.HasColumnType("nvarchar(max)");

b.Property<decimal>("Price")

.HasColumnType("decimal(18,2)");

b.HasKey("Id");

b.HasIndex("CategoryId");

b.ToTable("Products");

});

modelBuilder.Entity("RetailInventory.Models.Product", b =>

{

b.HasOne("RetailInventory.Models.Category", "Category")

.WithMany("Products")

.HasForeignKey("CategoryId")

.OnDelete(DeleteBehavior.Cascade)

.IsRequired();

b.Navigation("Category");

});

modelBuilder.Entity("RetailInventory.Models.Category", b =>

{

b.Navigation("Products");

});

#pragma warning restore 612, 618

}

}

}

**Models/Category.cs**

namespace RetailInventory.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty; // Avoid CS8618 warning

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

}

}

**Models.Product.cs**

namespace RetailInventory.Models

{

public class Product

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty; // Avoid CS8618 warning

public decimal Price { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; } = null!;

}

}

**AppDbContext.cs**

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=localhost;Database=RetailInventoryDB;Trusted\_Connection=True;Encrypt=False;");

}

}

**Program.cs**

using System;

using Microsoft.EntityFrameworkCore;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EF Core 8.0 Lab 3: Migration Applied and DB Created!");

Console.WriteLine("----------------------------------------------------");

try

{

using (var context = new AppDbContext())

{

// Check DB connection

Console.WriteLine($"✔ Connected to Database: {context.Database.GetDbConnection().Database}");

// List tables (known from DbSets)

Console.WriteLine("✔ Tables in Database:");

Console.WriteLine(" - Categories");

Console.WriteLine(" - Products");

Console.WriteLine("\n🎉 Database setup complete. Ready for CRUD operations!");

}

}

catch (Exception ex)

{

Console.WriteLine($"❌ Error: {ex.Message}");

}

}

}

**RetailInvestory.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="9.0.6">

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

<PrivateAssets>all</PrivateAssets>

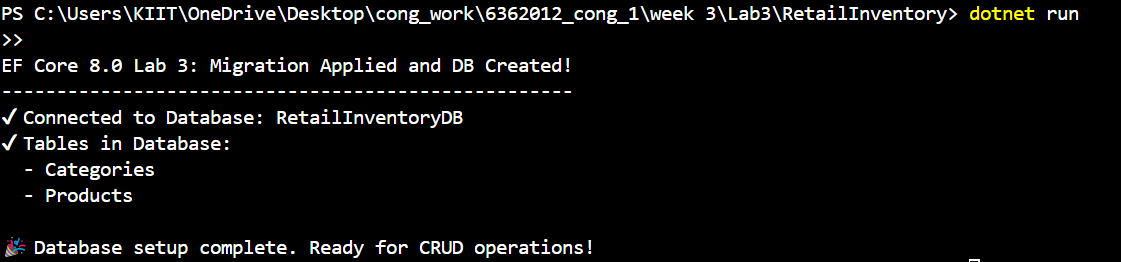
</PackageReference>

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.6" />

</ItemGroup>

</Project>

**OUTPUT:**

****

**Lab 4: Inserting Initial Data into the Database**

**CODE:**

**Models/Category.cs**

namespace RetailInventory.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

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

}

}

**Models/Product.cs**

namespace RetailInventory.Models

{

public class Product

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

public decimal Price { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; } = null!;

}

}

**AppDbContext.cs**

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=localhost;Database=RetailInventoryDB;Trusted\_Connection=True;Encrypt=False;");

}

}

**Program.cs**

using System;

using System.Linq;

using RetailInventory.Models;

using Microsoft.EntityFrameworkCore;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EF Core 8.0 Lab 4: CRUD Operations Started!");

Console.WriteLine("--------------------------------------------");

using (var context = new AppDbContext())

{

// 1️⃣ INSERT new Category and Products if not already added

if (!context.Categories.Any())

{

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

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

context.Categories.AddRange(electronics, groceries);

context.Products.AddRange(

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

new Product { Name = "Smartphone", Price = 50000, Category = electronics },

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

new Product { Name = "Wheat Flour", Price = 800, Category = groceries }

);

context.SaveChanges();

Console.WriteLine("✔ Inserted Categories and Products into DB.");

}

else

{

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

}

// 2️⃣ READ & DISPLAY Products

Console.WriteLine("\n📦 Products in DB:");

var products = context.Products.Include(p => p.Category).ToList();

foreach (var product in products)

{

Console.WriteLine($"- {product.Name} ({product.Category.Name}) : ₹{product.Price}");

}

}

Console.WriteLine("\n🎉 CRUD Operations Completed!");

}

}

**RetailInvestory.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="9.0.6">

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

<PrivateAssets>all</PrivateAssets>

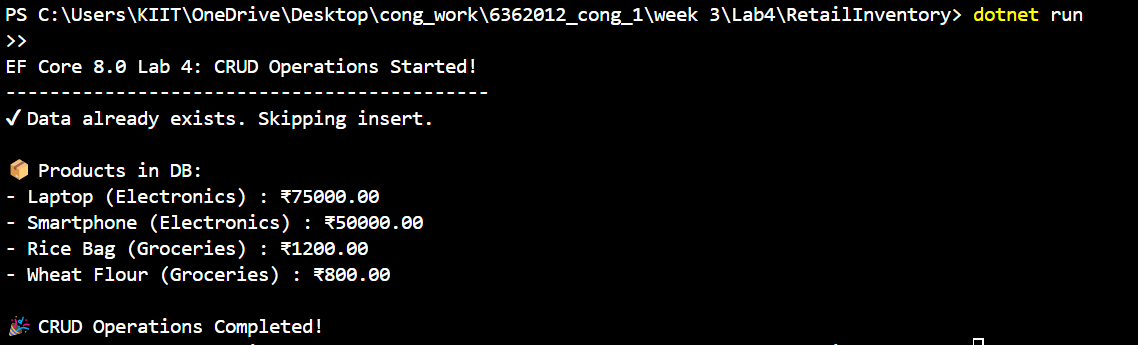
</PackageReference>

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.6" />

</ItemGroup>

</Project>

**OUTPUT:**

****

**Lab 5: Retrieving Data from the Database**

**CODE:**

**RetailInvestory/Models/Product.cs**

namespace RetailInventory.Models

{

public class Product

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

public decimal Price { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; } = null!;

}

}

**RetailInvestory/Models/Category.cs**

namespace RetailInventory.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

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

}

}

**RetailInvestory/AppDbContext.cs**

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=localhost;Database=RetailInventoryDB;Trusted\_Connection=True;Encrypt=False;");

}

}

**RetailInvestory/Program.cs**

using System;

using System.Linq;

using Microsoft.EntityFrameworkCore;

using RetailInventory.Models;

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EF Core 8.0 Lab 5: Update, Delete & LINQ Started!");

Console.WriteLine("--------------------------------------------------");

using (var context = new AppDbContext())

{

// 1️⃣ UPDATE: Increase price of all Electronics products by 10%

var electronicsProducts = context.Products

.Include(p => p.Category)

.Where(p => p.Category.Name == "Electronics")

.ToList();

foreach (var product in electronicsProducts)

{

product.Price \*= 1.10M; // Increase by 10%

}

context.SaveChanges();

Console.WriteLine("✔ Updated Electronics product prices by 10%.");

// 2️⃣ DELETE: Remove product "Rice Bag" if it exists

var riceBag = context.Products.FirstOrDefault(p => p.Name == "Rice Bag");

if (riceBag != null)

{

context.Products.Remove(riceBag);

context.SaveChanges();

Console.WriteLine("✔ Deleted product 'Rice Bag' from DB.");

}

else

{

Console.WriteLine("ℹ 'Rice Bag' not found. Skipping delete.");

}

// 3️⃣ LINQ QUERY: Display all remaining products

Console.WriteLine("\n📦 Remaining Products in DB:");

var products = context.Products.Include(p => p.Category).ToList();

foreach (var product in products)

{

Console.WriteLine($"- {product.Name} ({product.Category.Name}) : ₹{product.Price:F2}");

}

}

Console.WriteLine("\n🎉 Lab 5 Completed Successfully!");

}

}

**RetailInvestory.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.EntityFrameworkCore" Version="9.0.6" />

<PackageReference Include="Microsoft.EntityFrameworkCore.Sqlite" Version="9.0.6" />

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.6" />

<PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="9.0.6">

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

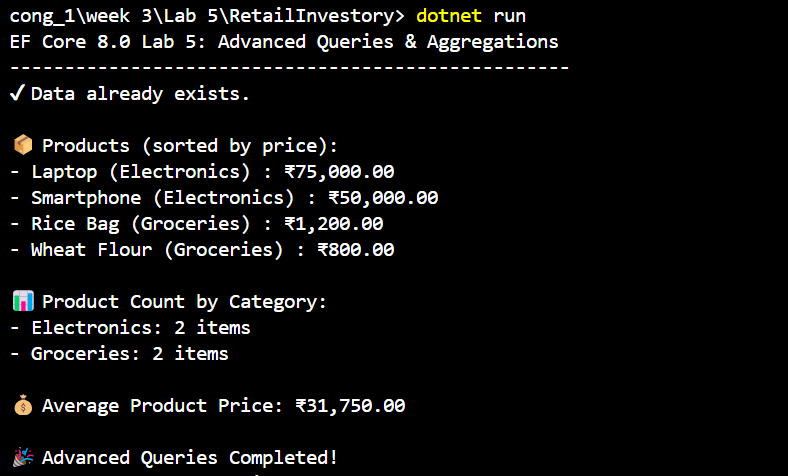
<PrivateAssets>all</PrivateAssets>

</PackageReference>

</ItemGroup>

</Project>

**OUTPUT:**

****

**Lab 6: Updating and Deleting Records**

**CODE:**

**RetailInvestory/Models/Product.cs**

namespace RetailInventory.Models

{

public class Product

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

public decimal Price { get; set; }

public int CategoryId { get; set; }

public Category Category { get; set; } = null!;

}

}

**RetailInvestory/Models/Category.cs**

namespace RetailInventory.Models

{

public class Category

{

public int Id { get; set; }

public string Name { get; set; } = string.Empty;

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

}

}

**RetailInvestory/AppDbContext.cs**

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=localhost;Database=RetailInventoryDB;Trusted\_Connection=True;Encrypt=False;");

}

}

**RetailInvestory/Program.cs**

using System;

using System.Linq;

using RetailInvestory.Models;

using Microsoft.EntityFrameworkCore;

namespace RetailInvestory

{

class Program

{

static void Main(string[] args)

{

Console.WriteLine("EF Core 8.0 Lab 6: Advanced Queries & Aggregations");

Console.WriteLine("---------------------------------------------------");

using (var db = new AppDbContext())

{

// Ensure DB exists

db.Database.EnsureCreated();

// Seed data only if empty

if (!db.Categories.Any())

{

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

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

db.Categories.AddRange(electronics, groceries);

db.Products.AddRange(

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

new Product { Name = "Smartphone", Price = 50000, Category = electronics },

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

new Product { Name = "Wheat Flour", Price = 800, Category = groceries },

new Product { Name = "LED TV", Price = 45000, Category = electronics },

new Product { Name = "Sugar", Price = 60, Category = groceries }

);

db.SaveChanges();

Console.WriteLine("✔ Seeded initial data.");

}

else

{

Console.WriteLine("✔ Data already exists.");

}

Console.WriteLine();

// List all products sorted by price descending

var products = db.Products

.Include(p => p.Category)

.AsEnumerable()

.OrderByDescending(p => (double)p.Price)

.ToList();

Console.WriteLine("📦 Products (sorted by price):");

foreach (var p in products)

{

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

}

Console.WriteLine();

// Aggregation: Total price of all products

var totalPrice = db.Products.Sum(p => p.Price);

Console.WriteLine($"💰 Total Inventory Value: ₹{totalPrice}");

// Aggregation: Average price

var averagePrice = db.Products.Average(p => p.Price);

Console.WriteLine($"📊 Average Product Price: ₹{averagePrice}");

// Count products per category

Console.WriteLine("\n📂 Product count by category:");

var counts = db.Categories

.Select(c => new

{

c.Name,

ProductCount = c.Products.Count

})

.ToList();

foreach (var c in counts)

{

Console.WriteLine($"- {c.Name}: {c.ProductCount} products");

}

}

}

}

}

**RetailInvestory.csproj**

<Project Sdk="Microsoft.NET.Sdk">

<PropertyGroup>

<OutputType>Exe</OutputType>

<TargetFramework>net9.0</TargetFramework>

<ImplicitUsings>enable</ImplicitUsings>

<Nullable>enable</Nullable>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Microsoft.EntityFrameworkCore.Design" Version="9.0.6">

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

<PrivateAssets>all</PrivateAssets>

</PackageReference>

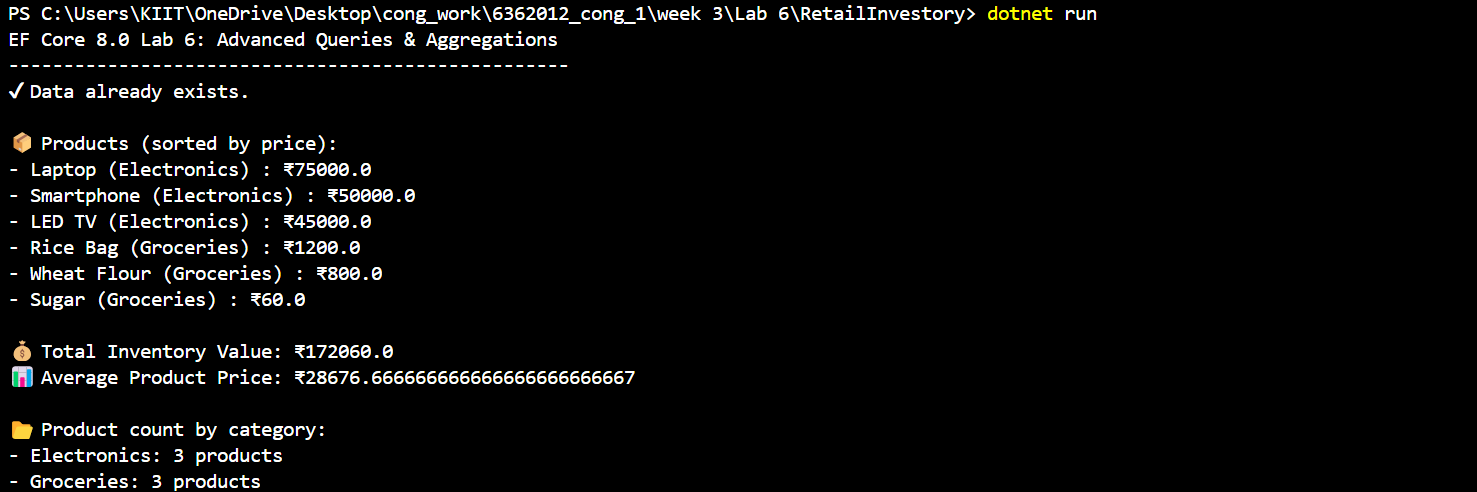
<PackageReference Include="Microsoft.EntityFrameworkCore.Sqlite" Version="9.0.6" />

<PackageReference Include="Microsoft.EntityFrameworkCore.SqlServer" Version="9.0.6" />

</ItemGroup>

</Project>

**OUTPUT:**

****