**Entity Framework Core 8.0 Hands\_On**

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

**Scenario:**

You’re building an inventory management system for a retail store. The store wants to

track products, categories, and stock levels in a SQL Server database.

**Objective:**

Understand what ORM is and how EF Core helps bridge the gap between C# objects and

relational tables.

**Step 1: What is ORM?**

ORM (Object-Relational Mapping) is a programming technique that allows developers to interact with a database using objects from their programming language — in our case, C#.

Instead of writing SQL commands to insert, read, update, or delete data, ORM tools like Entity Framework Core help us do all this using normal C# code.

**Benefits of using ORM:**

* **Productivity**: We write less code and do more work faster.
* **Maintainability**: Our C# code and database structure stay in sync.
* **Abstraction**: We don’t need to manually write SQL queries — everything is done through C# objects.

ORM makes the life of a developer easier by bridging the gap between **object-oriented programming** and **relational databases**.

**Step 2: EF Core vs EF Framework**

| **Feature** | **EF Core (Recommended)** | **EF Framework (EF6)** |
| --- | --- | --- |
| Platform Support | Cross-platform (.NET Core) | Windows-only(.NET Framework) |
| Performance | Lightweight and faster | Heavier and slower |
| Modern Features | LINQ, async, compiled queries | Limited modern features |
| Current Support | Actively developed (v8.0) | No new features, only fixes |

**Step 3: Key Features of EF Core 8.0**

Entity Framework Core 8.0 introduces powerful new features that improve performance, flexibility, and developer experience.

**1. JSON Column Mapping**

EF Core 8 allows mapping complex objects to a single database column in JSON format. Useful when you want to store data like:

{ "Color": "Red", "Size": "Large" }

**2. Compiled Models**

EF Core 8 compiles the data model at build time, which makes apps start faster and use less memory at runtime.

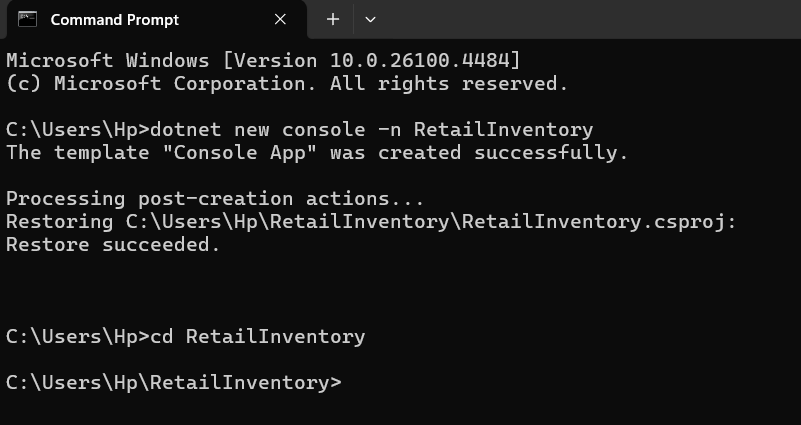
**3. Interceptors**

You can hook into EF’s execution pipeline and log, change, or monitor queries before they hit the database — great for debugging or analytics.

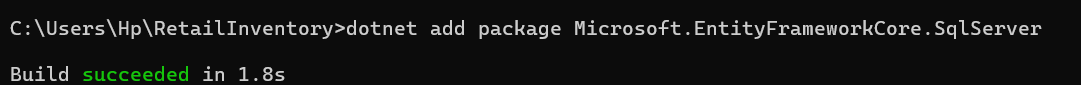
**4. Improved Bulk Operations**

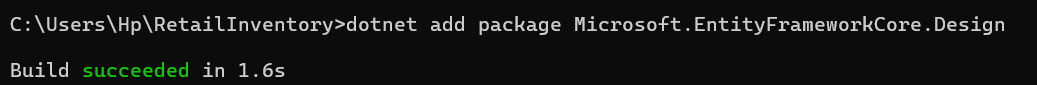
EF Core 8 handles large-scale inserts, updates, and deletes much more efficiently — very useful for enterprise-level applications.

**Step 4: Creating a Console App**

****

**Step 5: Installing EF Core Packages**

****

****

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

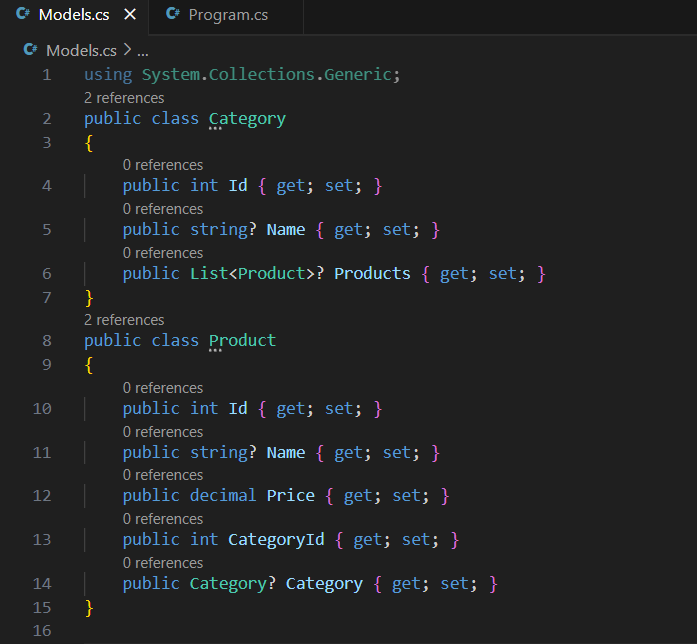
**Scenario:**

The retail store wants to store product and category data in SQL Server.

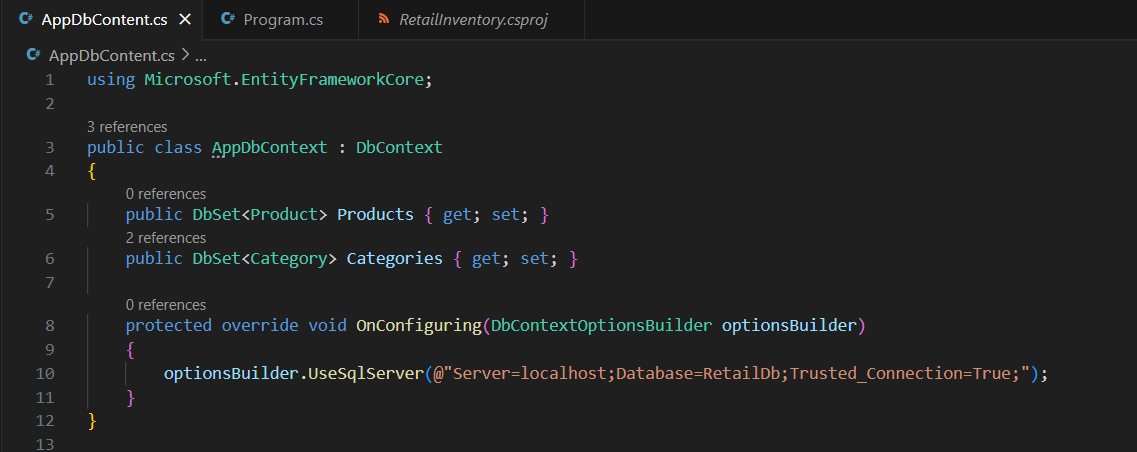
**Objective:**

Configure DbContext and connect to SQL Server.

**STEP 1 : Create Models**

****

**STEP 2 & 3: Create AppDbContext and Add Connection String**

****

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

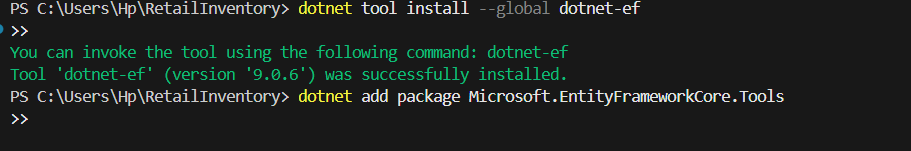
**Scenario:**

The retail store's database needs to be created based on the models you've defined. You’ll use EF Core CLI to generate and apply migrations.

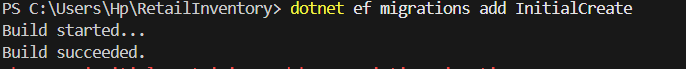
**Objective:**

Learn how to use EF Core CLI to manage database schema changes.

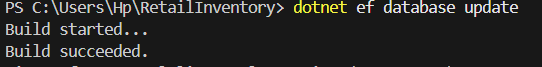
**Steps: 1.** Install EF Core CLI



**STEP 2 :** Create Initial Migration:

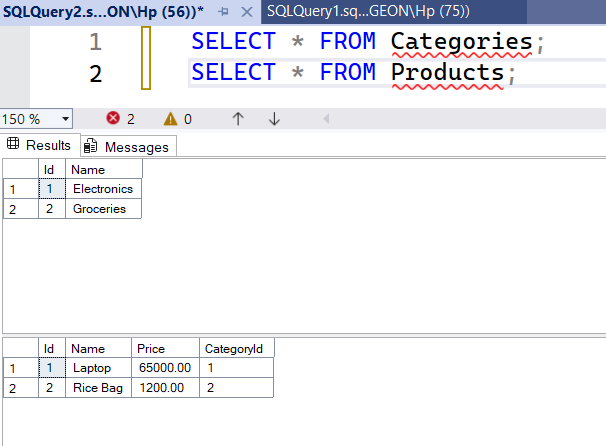


**STEP 3 :** Apply Migration to Create Database:



**STEP 4 : Verify in SQL Server:**

Open SQL Server Management Studio (SSMS) and confirm that tables Products and Categories are created.



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

**Scenario:**

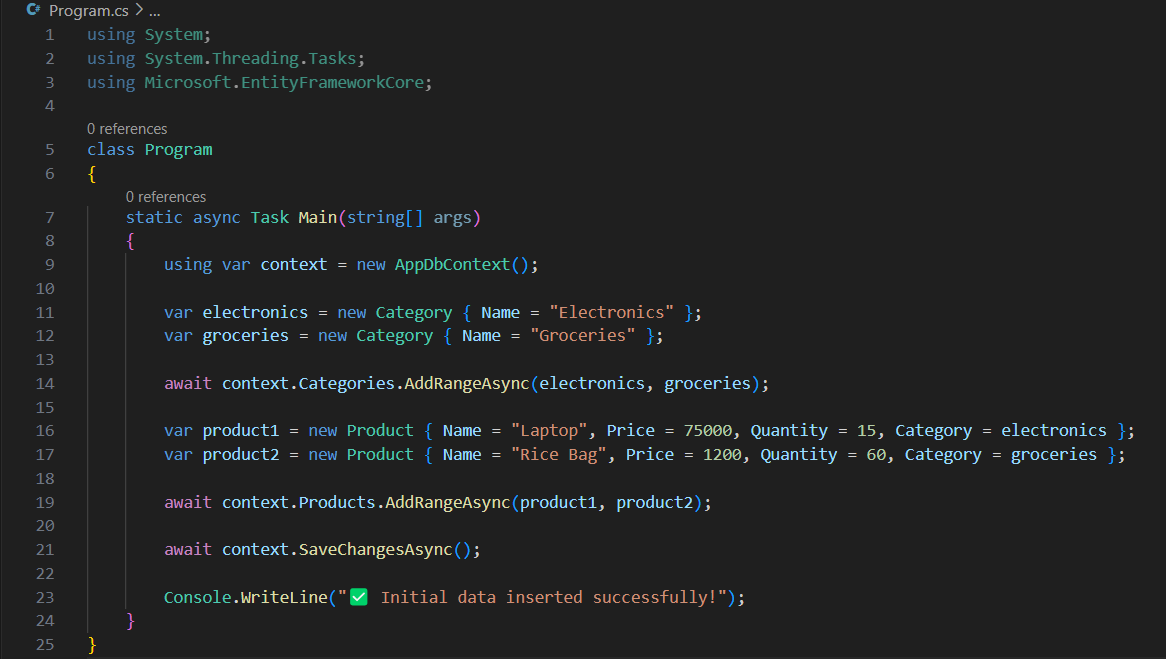
The store manager wants to add initial product categories and products to the system.

**Objective:**

Use EF Core to insert records using AddAsync and SaveChangesAsync.

**Steps:**

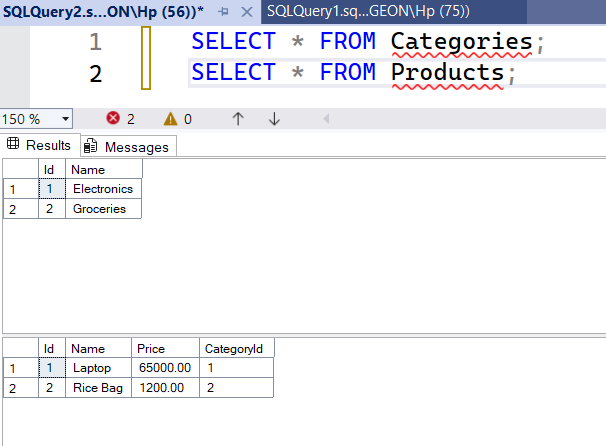
**STEP 1. Insert Data in Program.cs:**

****

**Step 2: Run the App**

****

**STEP 3:Verify in SQL Server:** Check that the data is inserted correctly.

****

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

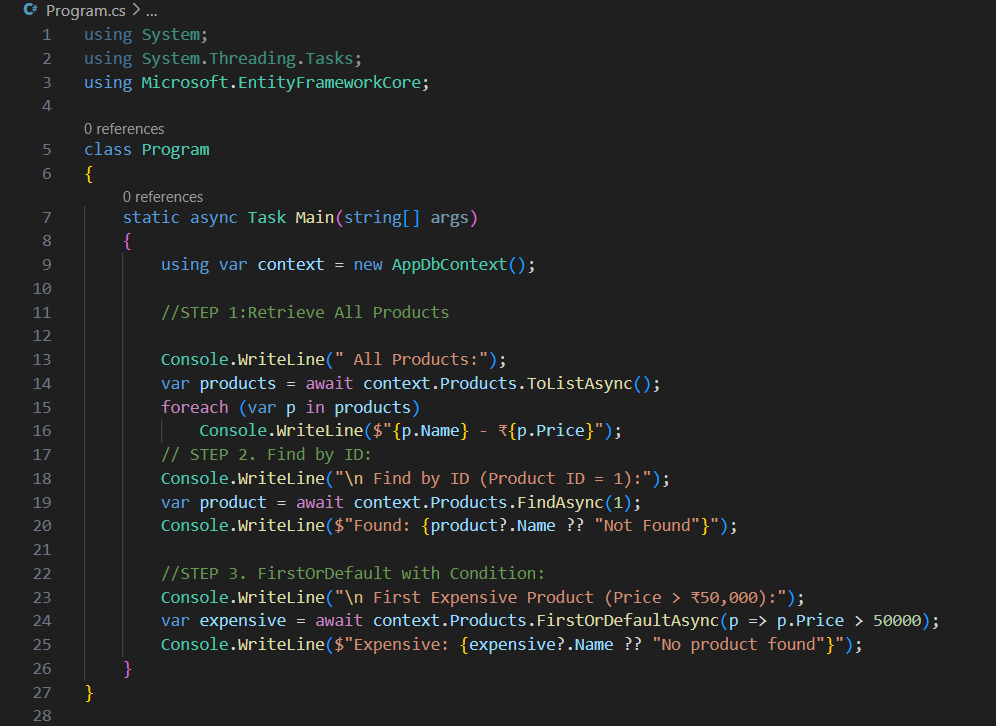
**Scenario:**

The store wants to display product details on the dashboard.

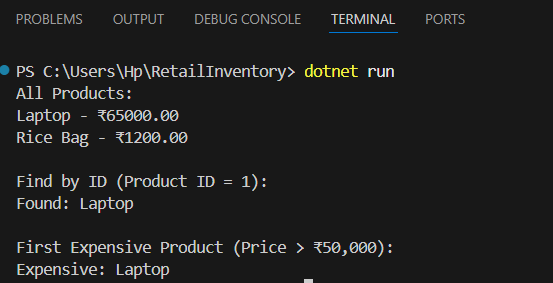
**Objective:**

Use Find, FirstOrDefault, and ToListAsync to retrieve data**.**

**CODE :**

****

**OUTPUT:**

****