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

**ORM** stands for Object-Relational Mapping.  
It is a programming technique that allows developers to interact with a relational database using object-oriented programming languages like C#.

**How ORM maps C# classes to database tables:**

* A C# class represents a database table.
* Properties of the class represent columns in the table.
* Objects of the class represent rows/records in the table.

**EF Core vs EF Framework**

Entity Framework Core (EF Core) is a modern, lightweight, and cross-platform version of the Entity Framework. It is designed to work with .NET Core and newer .NET versions and includes support for modern features such as asynchronous queries using async/await, compiled queries for better performance, and support for multiple database providers. EF Core is modular and more flexible, making it suitable for cloud and cross-platform applications.

On the other hand, Entity Framework (EF 6) is the older version that works only on the Windows platform with the .NET Framework. While EF 6 is mature and stable, it lacks some of the modern features found in EF Core and is not suitable for cross-platform development. However, it may still be preferred in legacy applications due to its long-term support and extensive features.

**EF Core 8.0 Features**

EF Core 8.0 introduces powerful features to enhance performance and developer experience:

**🔸 JSON Column Mapping**

* Allows mapping complex C# objects to JSON columns in SQL databases.
* Useful for storing flexible or hierarchical data inside a single column.

**🔸 Compiled Models**

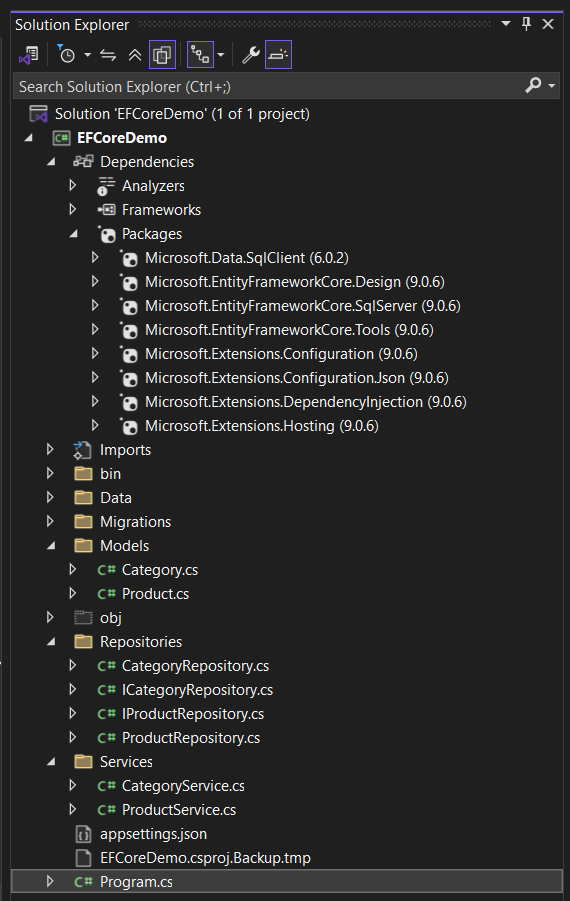
* Improves startup and query performance by **pre-compiling the EF Core model**.
* Great for large applications and microservices.

**🔸 Interceptors**

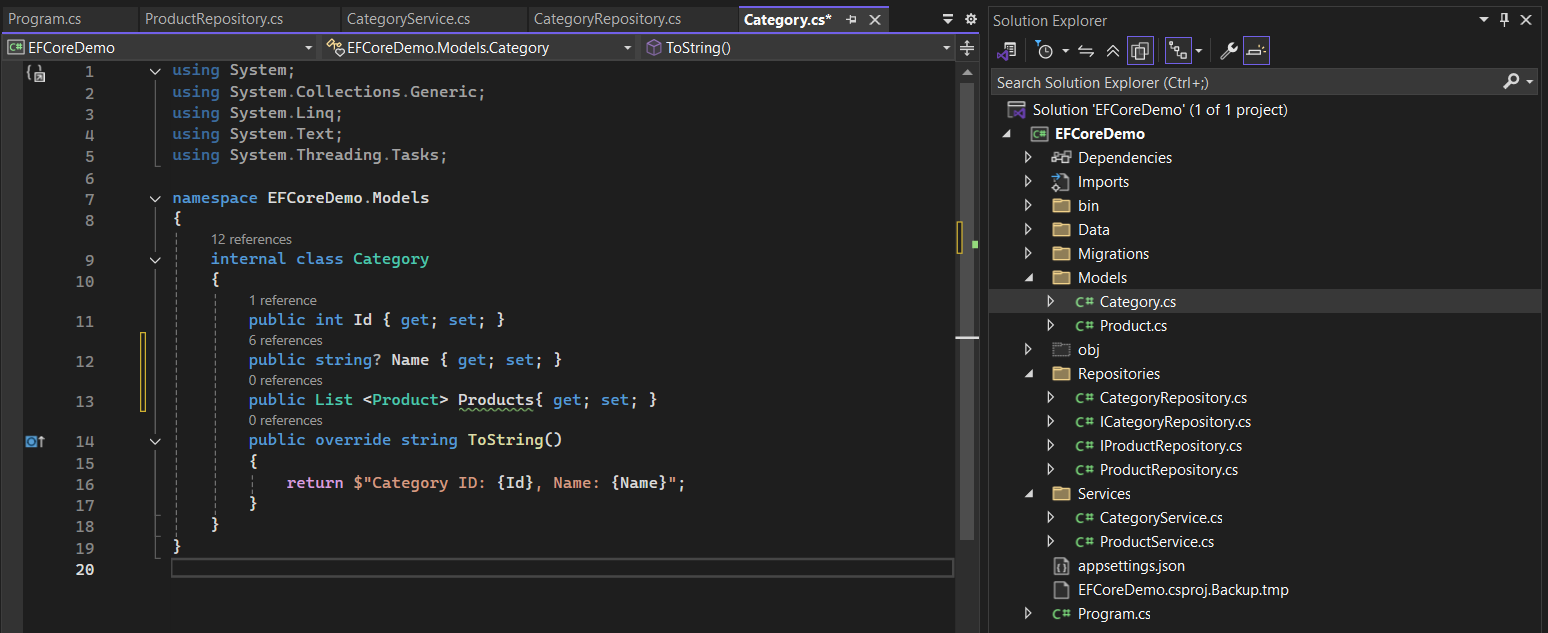
* Intercept and log or modify database operations like queries, saves, etc.
* Good for auditing, caching, or enforcing rules.

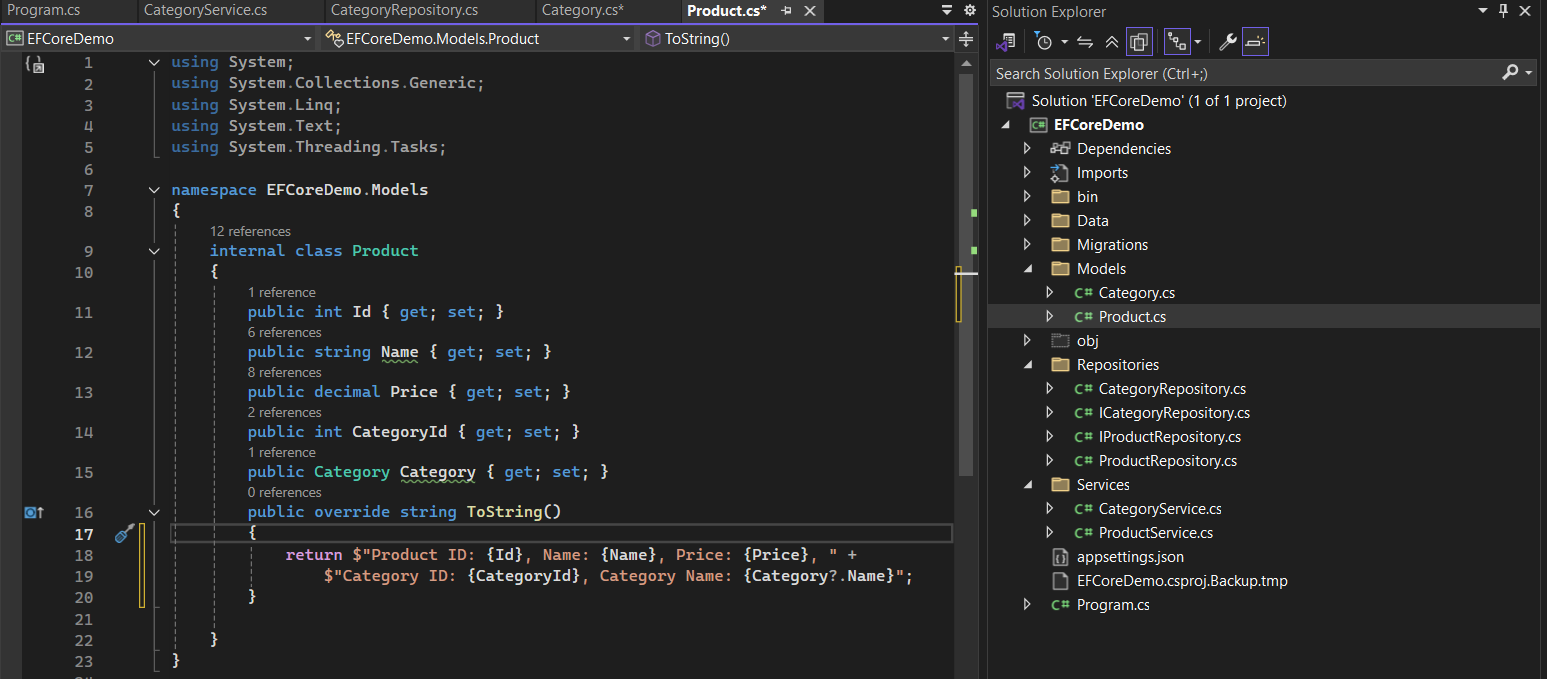
**🔸 Better Bulk Operations**

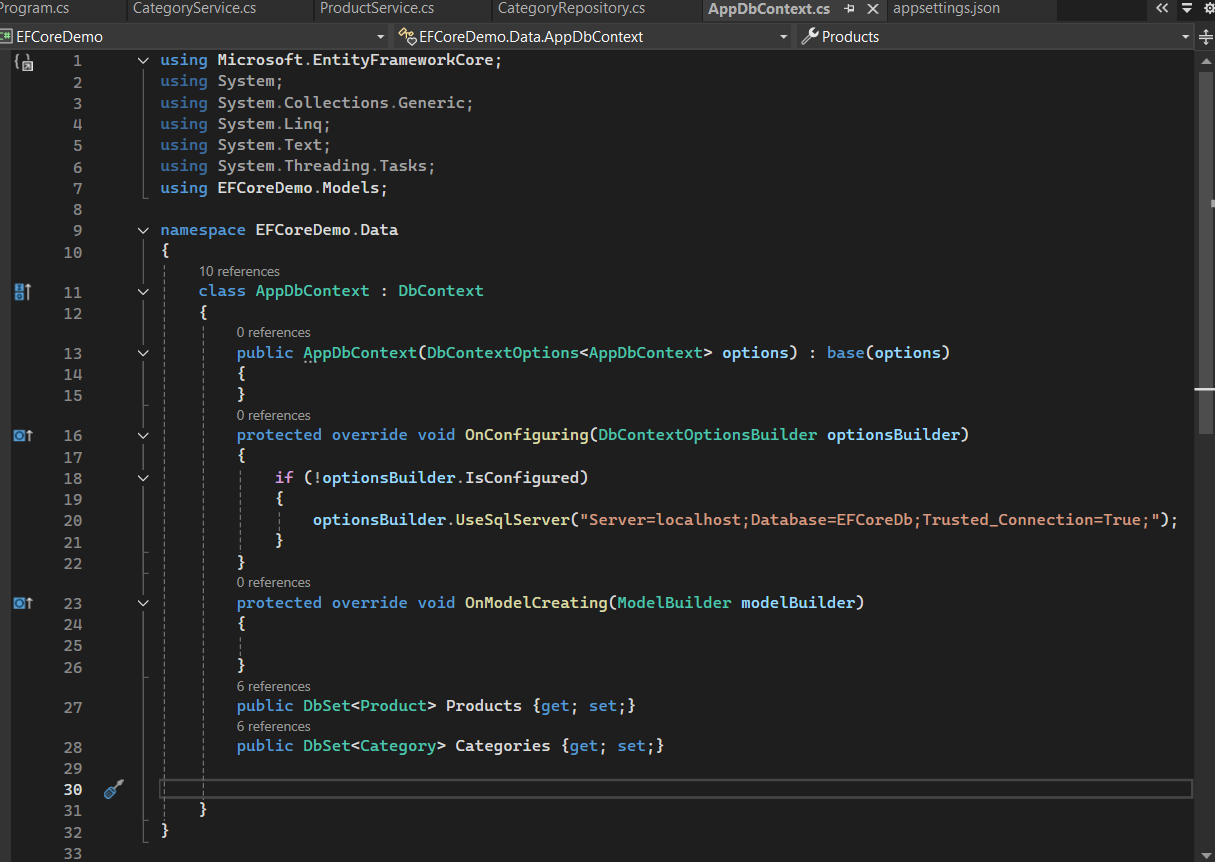
* Significant performance improvements when inserting or updating large data sets.

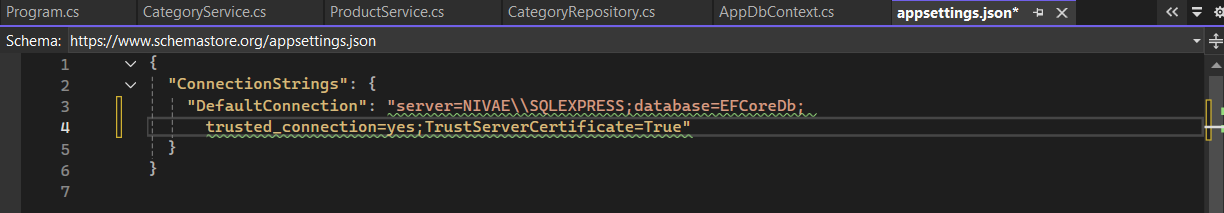


**Lab 2: Setting Up the Database Context for a Retail Store**

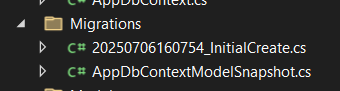


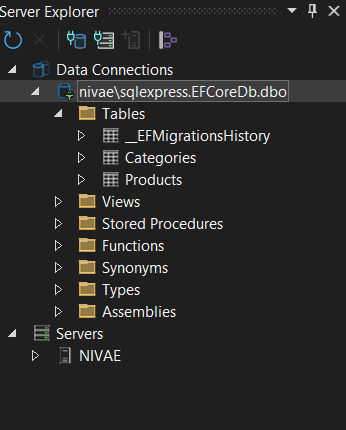


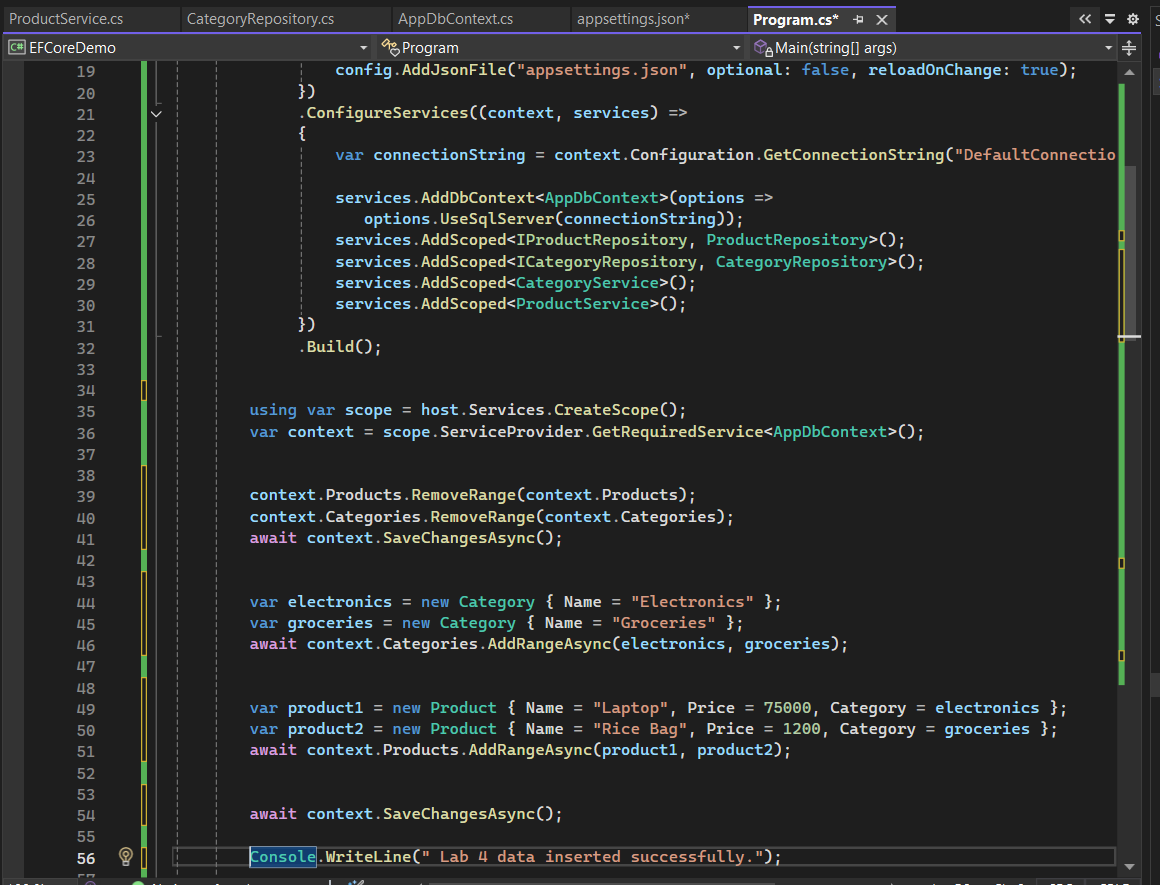


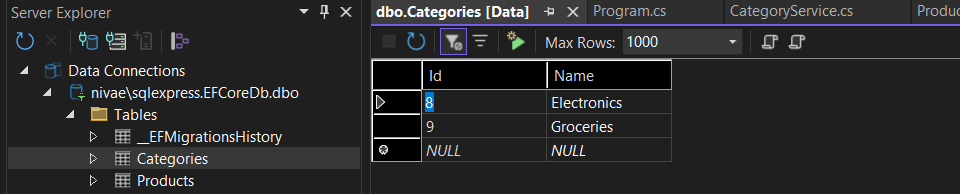


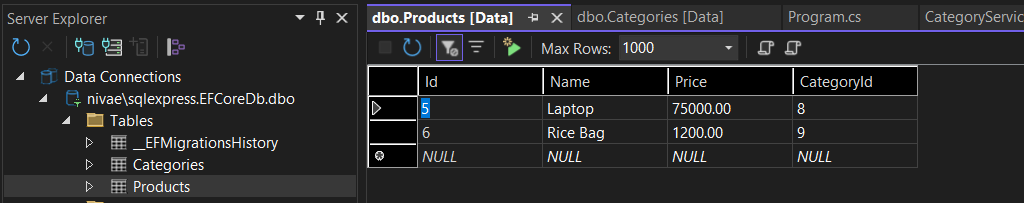
**Lab 3: Using EF Core CLI to Create and Apply Migrations**



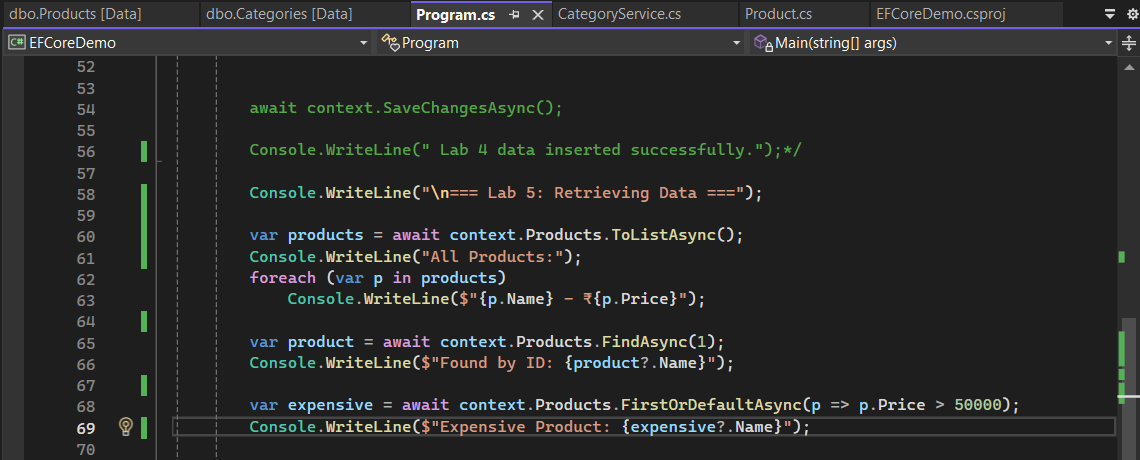


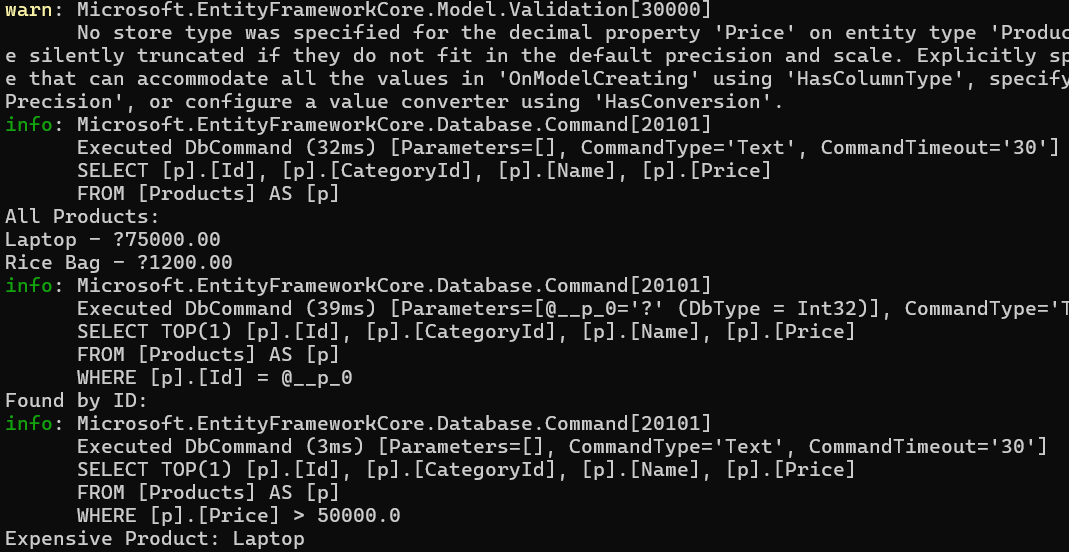
**Lab 4: Inserting Initial Data into the Database**



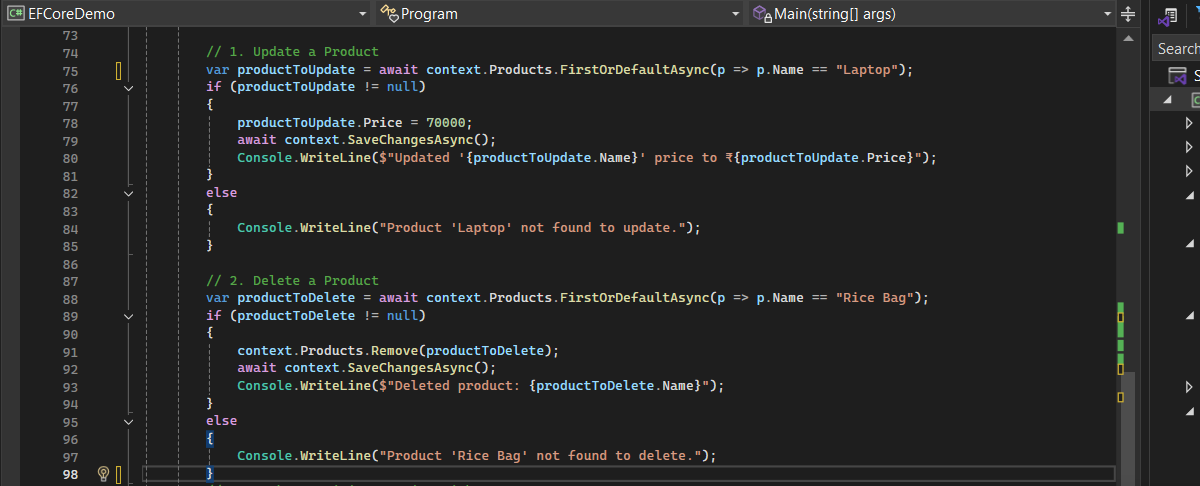


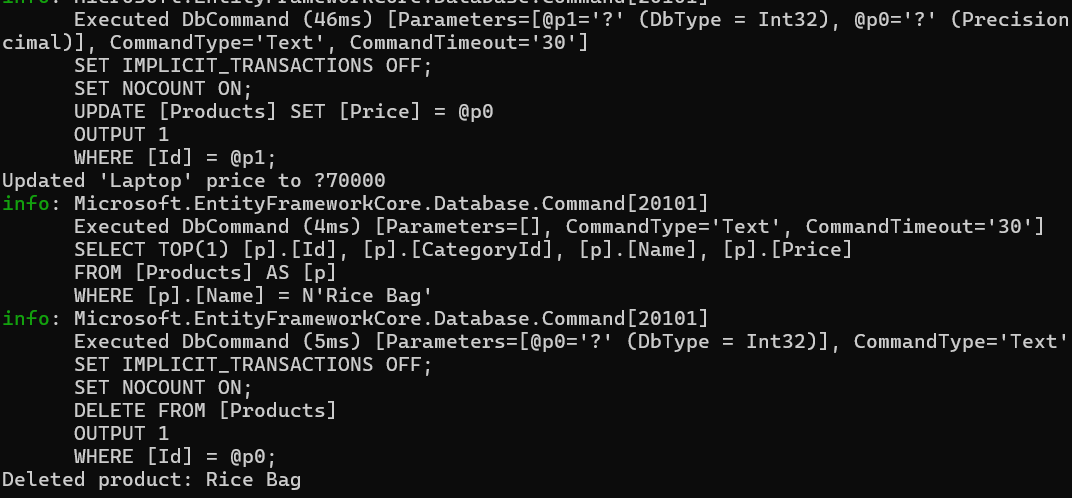
**Lab 5: Retrieving Data from the Database**





**Lab 6: Updating and Deleting Records**





**Lab 7: Writing Queries with LINQ**

