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

1. **What is ORM?**

• Explain how ORM maps C# classes to database tables.

• Benefits: Productivity, maintainability, and abstraction from SQL.

**->ORM (Object-Relational Mapping)** is a programming technique that allows developers to interact with a **relational database** (like SQL Server) using **object-oriented programming** concepts instead of writing raw SQL queries.

ORM frameworks like **Entity Framework Core (EF Core)** in C# map classes and their properties to tables and columns in the database:

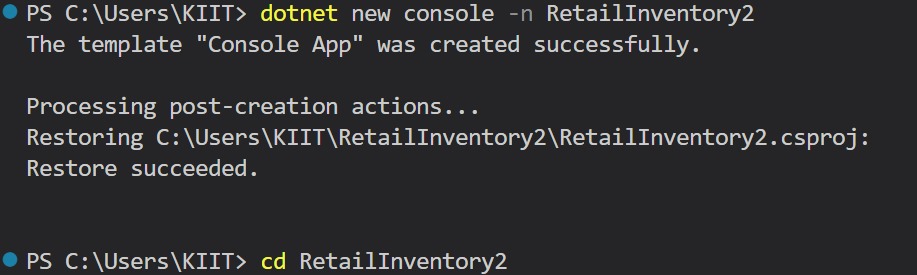
| **C# Class** | **SQL Table** |
| --- | --- |
| public class Product | Products |
| public int Id | Id (Primary Key) |
| public string Name | Name (Column) |
| public decimal Price | Price (Column) |

| **Benefit** | **Description** |
| --- | --- |
| **Productivity** | You write **less boilerplate SQL**; EF Core handles SQL generation for CRUD. |
| **Maintainability** | Changes in models automatically update the DB schema via **migrations**. |
| **Abstraction** | You can query using **LINQ** instead of raw SQL — making code cleaner. |

1. **Create a .NET Console App:**

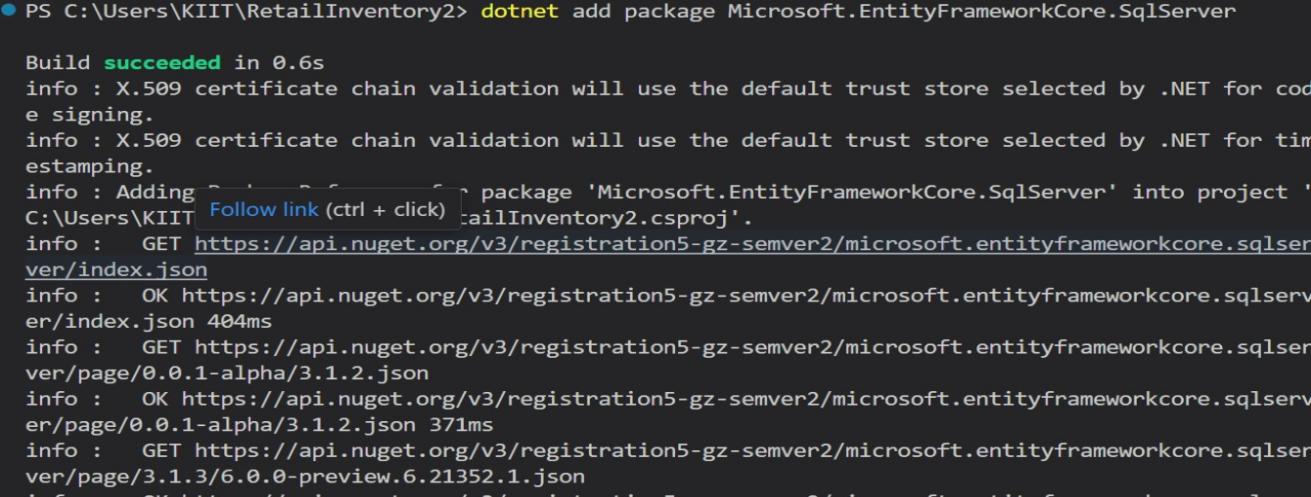
dotnet new console -n RetailInventory

cd RetailInventory



1. **Install EF Core Packages:**

dotnet add package Microsoft.EntityFrameworkCore.SqlServer



dotnet add package Microsoft.EntityFrameworkCore.Design

