**Week-3**

# **LAB-1**

# 

# **Understanding ORM and EF Core**

## **1. What is ORM?**

ORM (Object-Relational Mapping) is a programming technique that allows developers to interact with relational databases using object-oriented languages like C#. ORM frameworks map C# classes to database tables and their properties to table columns, enabling seamless data operations without writing raw SQL queries.

### **How ORM Maps C# Classes to Database Tables**

* Each C# class corresponds to a database table.
* Each property in a class corresponds to a column in that table.
* Relationships between classes (like one-to-many) correspond to foreign keys in the database.

### **Benefits of ORM**

* Productivity: Work with objects instead of writing SQL manually.
* Maintainability: Easily update database schema via migrations as models change.
* Abstraction: ORM abstracts database-specific SQL, allowing switching databases without much code change.

## **2. EF Core vs EF Framework**

* EF Core
  + Cross-platform and lightweight.
  + Supports modern features like LINQ, asynchronous queries, and compiled queries.
* EF Framework (EF6)
  + Windows-only and more mature.
  + Less flexible and limited to older .NET Frameworks.

## **3. EF Core 8.0 Features**

* JSON column mapping support.
* Improved performance with compiled models.
* Interceptors and enhanced bulk operations.

## **4. Create a .NET Console App**

Use the following commands to create a console application:

dotnet new console -n RetailInventory  
 cd RetailInventory

## **5. Install EF Core Packages**

dotnet add package Microsoft.EntityFrameworkCore.SqlServer   
 dotnet add package Microsoft.EntityFrameworkCore.Design