**WEEK-3\_EF Core 8.0\_LAB1**

**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.

Steps:

1. What is ORM?

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

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

1. EF Core vs EF Framework:

• EF Core is cross-platform, lightweight, and supports modern features like LINQ, async queries, and compiled queries.

• EF Framework (EF6) is Windows-only and more mature but less flexible.

1. EF Core 8.0 Features:

• JSON column mapping.

• Improved performance with compiled models.

• Interceptors and better bulk operations.

1. Create a .NET Console App:

dotnet new console -n RetailInventory cd RetailInventory

5. Install EF Core Packages:

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Microsoft.EntityFrameworkCore.Design

1. What is ORM?

**ORM (Object-Relational Mapping)** is a programming technique used to convert data between **C# objects** and **relational database tables**. It allows developers to interact with the database using **.NET classes and objects**, instead of writing SQL queries directly.

**How ORM Works (Mapping Concept)**

* A **C# class** represents a **table** in the database.
* A **class property** maps to a **column**.
* A **class object (instance)** represents a **row** in the table.
* Relationships like **One-to-Many** or **Many-to-Many** are modeled using navigation properties.

**Benefits of Using ORM:**

| **Benefit** | **Explanation** |
| --- | --- |
| **Abstraction from SQL** | Developers can use C# and LINQ to access data without writing SQL manually. |
| **Improved Productivity** | Code is faster to write and easier to maintain with fewer lines. |
| **Easier Maintenance** | Changes in the database can be reflected in C# classes, improving maintainability. |
| **Reduced Risk of SQL Injection** | ORM frameworks like EF Core use parameterized queries internally, making the app safer. |
| **Relationship Modeling** | ORM allows you to model foreign key relationships and navigation properties directly in C#. |
| **Cross-Platform Compatibility** | EF Core works on Windows, Linux, and macOS, supporting multiple databases. |

**EF Core vs EF Framework**:

| **Feature** | **EF Core** | **Entity Framework (EF6)** |
| --- | --- | --- |
| **Platform** | Cross-platform (.NET Core, .NET 8+) | Windows-only (.NET Framework) |
| **Performance** | Lightweight, faster, and more efficient | Heavier with some performance overhead |
| **Modern Features** | Supports LINQ, async queries, compiled queries | Limited or no support for async LINQ |
| **Flexibility** | Modular and highly customizable | Less flexible, tightly coupled |
| **Database Providers** | Works with SQL Server, SQLite, PostgreSQL, etc. | Mostly focused on SQL Server |
| **Future Development** | Actively developed and updated (EF Core 8+) | No major updates (EF6 is legacy) |

**EF Core 8.0 Features**

| **Feature** | **Description** |
| --- | --- |
| **JSON Column Mapping** | You can now map JSON data from the database directly to C# classes — useful for semi-structured data. |
| **Compiled Models** | Improves startup and query performance by compiling the model at build time instead of runtime. |
| **Interceptors** | Enables developers to plug into EF Core’s operations (like command execution) for logging, auditing, etc. |
| **Bulk Operations** | Improved support for high-performance bulk insert, update, and delete operations — ideal for large datasets. |
| **Better Testing Support** | Easier mocking and testing of EF Core with in-memory databases and improved DI support. |

Create a .NET Console App:

dotnet new console -n RetailInventory cd RetailInventory

Install EF Core Packages:

dotnet add package Microsoft.EntityFrameworkCore.SqlServer

dotnet add package Microsoft.EntityFrameworkCore.Design

