Understanding ORM and Entity Framework Core

# 1. What is ORM?

Object-Relational Mapping (ORM) is a technique that allows you to interact with a relational database using object-oriented programming languages like C#. In ORM:  
  
- Each C# class (e.g., Product, Category) maps to a database table.  
- Each object instance (e.g., a specific product) maps to a row in that table.  
- Each property in the class (e.g., Name, Price) maps to a column in the table.  
  
How it works:  
- You define your domain model as C# classes.  
- The ORM framework (like Entity Framework Core) automatically handles the translation between these classes and the underlying SQL tables.  
- You perform CRUD (Create, Read, Update, Delete) operations on objects, and the ORM generates the necessary SQL behind the scenes.  
  
Benefits:  
- Productivity: Developers can focus on business logic rather than SQL queries and database schema management.  
- Maintainability: Changes to the data model are managed in one place (the C# code), making refactoring and updates easier.  
- Abstraction: You don’t need to write repetitive SQL; the ORM handles query generation and data mapping for you.

# 2. EF Core vs. Entity Framework (EF6)

|  |  |  |
| --- | --- | --- |
| Feature/Aspect | EF Core | EF Framework (EF6) |
| Platform | Cross-platform (.NET Core, .NET 5/6/7/8, Windows, Linux, macOS) | Windows-only (.NET Framework) |
| Lightweight/Modular | Yes, rearchitected for modularity and extensibility | No, monolithic |
| Modern Features | Supports LINQ, async queries, compiled queries, NoSQL (some) | Limited LINQ, less async support |
| Maturity | Newer, evolving rapidly | More mature, stable |
| Performance | Improved, optimized for speed and resource usage | Slower in some scenarios |
| Recommended For | New .NET applications (.NET 5+) | Legacy .NET Framework apps |

EF Core is the modern, cross-platform ORM for .NET, designed for flexibility and performance.  
  
EF6 is the older, Windows-only ORM, more mature but less flexible and not recommended for new projects.

# 3. EF Core 8.0 Features

EF Core 8.0 introduces several enhancements:  
  
- JSON Column Mapping: Map C# complex types or collections directly into JSON columns in SQL Server and SQLite, enabling more flexible data structures within a relational database.  
- Compiled Models & Performance: Faster query execution, optimized connection pooling, improved lazy loading, and better change tracking for large models.  
- Bulk Operations: Native support for ExecuteUpdate() and ExecuteDelete() allows efficient bulk updates and deletes directly in the database, improving performance for large data sets.  
- Interceptors: Enhanced ability to intercept and customize database operations, useful for logging, auditing, or custom behaviors.  
- Other Improvements: Enhanced LINQ support, better schema handling, and breaking changes for improved consistency and clarity.

# 4. Create a .NET Console App

Step-by-step:  
  
Create a new console project:  
  
```  
dotnet new console -n RetailInventory  
cd RetailInventory  
```  
  
This creates a new folder and initializes a .NET console application in it.

# 5. Install EF Core Packages

You need to add EF Core packages to your project to use it with SQL Server and to enable design-time features like migrations.  
  
Install the SQL Server provider:  
  
```  
dotnet add package Microsoft.EntityFrameworkCore.SqlServer  
```  
  
Install the Design package:  
  
```  
dotnet add package Microsoft.EntityFrameworkCore.Design  
```  
  
Why these packages?  
  
- Microsoft.EntityFrameworkCore.SqlServer enables EF Core to connect to SQL Server, execute queries, and manage schema.  
- Microsoft.EntityFrameworkCore.Design provides tools for database migrations, reverse engineering, and other design-time activities (like dotnet ef CLI commands).

# Summary of Steps in Context:

- Understand ORM: ORM maps C# classes to database tables, making data access easier and code more maintainable.  
- EF Core vs EF6: EF Core is modern, cross-platform, and feature-rich; EF6 is older and Windows-only.  
- EF Core 8.0 Features: JSON mapping, bulk operations, interceptors, and major performance improvements.  
- Create Console App: Use dotnet new console to scaffold your project.  
- Install Packages: Use dotnet add package to install EF Core for SQL Server and the design-time tools.

Submitted by: Subhodeep Mukherjee