**Superset id:- 6363303**

**Q.1 What is ORM?**

-> ORM stands for **Object-Relational Mapping**.  
 It’s a technique that helps you work with a database using **regular C# classes**, instead of writing raw SQL queries all the time.

Think of it like this: if you create a Product class in C#, ORM maps that class to a table called Products in the database. Each property of the class becomes a column in that table — for example, Name, Price, etc.

Example :-

**SQL query:-**

SELECT \* FROM Products WHERE Price > 1000;

**Using C#:-**

context.Products.Where(p => p.Price > 1000).ToList();

#### **Benefits:-**

* **Productivity**: You write less boilerplate code and move faster.
* **Maintainability**: Easier to update your code and database together.
* **No SQL clutter**: It abstracts away SQL so you can stay in C#.

**EF Core vs EF Framework:-**

**EF Core :-**

* It’s the **modern, lightweight version** of Entity Framework.
* It works across **platforms** (Windows, macOS, Linux).
* You can use advanced C# features like:-
  + **LINQ** for filtering and sorting
  + **Async/await** for better performance
  + **Compiled queries** for speed

**EF Framework (EF6) :-**

* It’s the **older version**, mostly used with full .NET Framework (not .NET Core/.NET 6+).
* It only works on **Windows**.
* It’s stable and mature, but **not as flexible** or fast as EF Core.
* Great for legacy projects, but not ideal for new ones.

**EF Core 8.0 Features :-**

#### **JSON Column Mapping**

You can now store and read **JSON objects directly in your database columns**.  
 Let’s say you have a ProductDetails object (like size, color, specs) — instead of creating extra tables, you can store all that in one JSON column.

EF Core 8 lets you **read and write** to that column as if it were a C# object. It’s really useful for flexible data.

#### 

#### **Compiled Models for Performance**

In older versions, EF had to **re-figure out the model each time the app ran**. Now in EF Core 8, you can **pre-compile your model** — that means faster startup and better performance, especially in large apps or microservices.It’s like having a shortcut instead of recalculating everything on the fly.

#### **Interceptors and Bulk Operations**

EF Core 8 gives more control behind the scenes:

* **Interceptors** let you hook into EF’s behavior — for example, logging every query, changing values, or catching errors automatically.
* **Bulk operations** let you insert or update many rows **quickly**, instead of looping through them one by one.

**Ayush Kumar (6363303)**