



What is EF Core?

Entity Framework **EF** is an open-source ORM framework for .NET applications supported by Microsoft. Object-Relational Mapping (ORM) is **a technique that lets you query and manipulate data from a database using an object-oriented paradigm**

It enables developers to work with data using objects of domain specific classes without focusing on the underlying database tables and columns where this data is stored.

With the Entity Framework, developers can work at a higher level of abstraction when they deal with data, and can create and maintain data-oriented applications with less code compared with traditional applications.

Entity Framework Core (EF Core) is a complete rewrite of Entity Framework 6 that was first released in 2016.

EF Core is a cross-platform product that runs on .NET Core.

EF Core was designed to provide a developer experience similar to EF6. Most of the top-level APIs remain the same.

Why EF Core?

Before release of .NET 3.5, developers used to write [ADO.NET](https://docs.microsoft.com/en-us/ado/ado-net) code or Enterprise Data Access Block to save or retrieve application data from the underlying database.

One had to open a connection to the database, create a **DataSet** to fetch or submit the data to the database, convert data from the **DataSet** to .NET objects or vice-versa to apply business rules.

This was a cumbersome and error prone process. Microsoft then provided a framework called "Entity Framework" to automate all these database related activities for your application.

- It provides auto generated code which reduces development time. Less code is written by the developer.
- It provides capability of programming a conceptual model.
- It abstracts the need for knowledge about underlying data store by providing a universal syntax (LINQ) for all object queries whether it is database or not.

However, Its not all roses here at Ef Core

- It is not available for every RDMS case in point MariaDB.
- It is not necessarily good to use in huge data domain models.

How to Use EF Core

- Code-first approach. In this approach, developers start writing entities (domain classes) and context class and then create the database from these classes using migration commands. Developers who follow the Domain-Driven Design (DDD) principles, prefer to begin with coding their domain classes first and then generate the database required to persist their data.
- Database-first approach. Under this development approach, developers generate the context and entities for an existing database using the wizard integrated in Visual Studio or by executing EF commands.

💡 EF 6 supports the database-first approach extensively but EF Core has a limited support.

The DbContext

The **DbContext** class is an integral part of Entity Framework. An instance of **DbContext** represents a session with the database which can be used to query and save instances of your entities to a database.

This is the same class that performs the following tasks:

- Managing database connections.
- Configuring model & relationships.
- Querying the database.
- Saving data to the database.
- Configuring model change tracking.
- Caching
- Transaction management

The **DbContext** has a number of methods that as a developer using ef core will need to use to perform actions on data models. Here's a few of the commonest

Method	What it does
Add	Adds a new entity to DbContext with Added state and starts tracking it. This new entity data will be inserted into the database when SaveChanges() is called.
AddRange	Adds a collection of new entities to DbContext with Added state and starts tracking it. This new entity data will be inserted into the database when SaveChanges() is called.
SaveChanges	Execute INSERT, UPDATE or DELETE command to the database for the entities with Added, Modified or Deleted state.
Find	Finds an entity with the given primary key values.
Remove	Removes a specific entity.
RemoveRange	Removes a collection of specific entities.
Where	Allows you to add filter expressions.



These methods have asynchronous variations that can be used in async applications