There are 3 approaches through which Entity framework is implemented.

1. Database First
2. Code First
3. Model First

Of these Database first and Code First  are the most used ones. In this article I will not discuss model first approach.

First let us understand what are code first and database first

**Code First Approach**In code first approach we will first create entity classes with properties defined in it. Entity framework will create the database and tables based on the entity classes defined. So database is generated from the code. When the dot net code is run database will get created.

**Advantages**

1. You can create  the database and tables from your business objects.
2. You can specify which related collections are to be eager loaded, or not be serialized at all.
3. Database version control.
4. Good for small applications.

**Disadvantages**

1. You have to write everything related to database in the visual studio code.
2. For stored procedures you have to map stored procedure using Fluent API and write Stored Procedure inside the code.
3. If you want to change anything in the database tables you to make changes in the entity classes in the code file and run the update-database from the  package manager console.
4. Not preferred for Data intensive applications.

**Database First Approach**In this approach Database and tables are created first. Then you create entity Data Model using the created database.

**Advantages**

1. Simple to create the data model
2. Graphical  user interface.
3. Mapping and creation of keys and relationships are easy as you need not have to write any code .
4. Preferred for data intense and large applications

**Disadvantages**

1. Using an existing database to generate a .edmx model file and the associated code models results in a giant pile of auto generated code.
2. When you need to add any functionality to generated model you have to extend the model class generated.

Choosing the appropriate approach is purely based on the applications you are developing.

**Entity Framework 6**

Entity Framework 6 (EF6) is a tried and tested data access technology. It was first released in 2008, as part of .NET Framework 3.5 SP1 and Visual Studio 2008 SP1. Starting with the 4.1 release it has shipped as the [EntityFramework](https://www.nuget.org/packages/EntityFramework/) NuGet package. EF6 runs on the .NET Framework 4.x, which means it runs only on Windows.

EF6 continues to be a supported product, and will continue to see bug fixes and minor improvements.

**Entity Framework Core**

Entity Framework Core (EF Core) is a complete rewrite of EF6 that was first released in 2016. It ships in Nuget packages, the main one being [Microsoft.EntityFrameworkCore](https://www.nuget.org/packages/Microsoft.EntityFrameworkCore/). EF Core is a cross-platform product that can run on .NET Core or .NET Framework.

EF Core was designed to provide a developer experience similar to EF6. Most of the top-level APIs remain the same, so EF Core will feel familiar to developers who have used EF6.