**Entity framework code first Approach**

What is code first approach :

Entity Framework introduced the Code-First approach with Entity Framework 4.1. In the Code-First approach, you focus on the domain of your application and start creating classes for your domain entity rather than design your database first and then create the classes which match your database design. The following figure illustrates the code-first approach.

Graphical user interface, application

Description automatically generated

As you can see in the above figure, EF API will create the database based on your domain classes and configuration. This means you need to start to coding first in C-Sharp and then entity framework will create the database from your code.

Example:

Diagram

Description automatically generated

Different between code-first approach and other approach:

A screenshot of a computer

Description automatically generated with low confidence

**Code first**

* Very popular because hardcore programmers don't like any kind of designers and defining mapping in EDMX xml is too complex.
* Full control over the code (no autogenerated code which is hard to modify).
* General expectation is that you do not bother with DB. DB is just a storage with no logic. EF will handle creation and you don't want to know how it does the job.
* Manual changes to database will be most probably lost because your code defines the database.

**Database first**

* Very popular if you have DB designed by DBAs, developed separately or if you have existing DB.
* You will let EF create entities for you and after modification of mapping you will generate POCO entities.
* If you want additional features in POCO entities you must either T4 modify template or use partial classes.
* Manual changes to the database are possible because the database defines your domain model. You can always update model from database (this feature works quite well).
* I often use this together VS Database projects (only Premium and Ultimate version).

**Model first**

* IMHO popular if you are designer fan (= you don't like writing code or SQL).
* You will "draw" your model and let workflow generate your database script and T4 template generate your POCO entities. You will lose part of the control on both your entities and database but for small easy projects you will be very productive.
* If you want additional features in POCO entities you must either T4 modify template or use partial classes.
* Manual changes to database will be most probably lost because your model defines the database. This works better if you have Database generation power pack installed. It will allow you updating database schema (instead of recreating) or updating database projects in VS.

I expect that in case of EF 4.1 there are several other features related to Code First vs. Model/Database first. Fluent API used in Code first doesn't offer all features of EDMX. I expect that features like stored procedures mapping, query views, defining views etc. works when using Model/Database first and DbContext (I haven't tried it yet) but they don't in Code first.

## Why Code First? Not others approach:

reasons why you might use the code first approach:

Diagram

Description automatically generated

The workflows you have to choose from are:

1. Code first creating a new database
2. Code first to an existing database
3. Model designer creating a new database
4. Existing database to generated model

Code First can do (database first and model first )

#### Greater Control

When you go DB first, you’re at the mercy of what gets generated for your models for use in your application. Occasionally the naming convention is undesirable. Sometimes the relationships and associations aren't quite what you want. Other times non transient relationships with lazy loading wreak havoc on your API responses.

While there is almost always a solution for model generation problems you might run into, going code first gives you complete and fine grained control from the get go. You can control every aspect of both your code models and your database design from the comfort of your business object. You can precisely specify relationships, constraints, and associations. You can simultaneously set property character limits and database column sizes. You can specify which related collections are to be eager loaded, or not be serialized at all. In short, you are responsible for more stuff but you’re in full control of your app design.

**Getting start with entity framework**

## Environnement Setup :

To start working with EF Code First approach you need the following tools to be installed on your system.

* Visual studio 2019 or 2022
* Sql server
* Entity Framework via NuGet Package. (EF6)

**Install EF internet Required:**

1 3

**A screenshot of a computer

Description automatically generated with medium confidence**A screenshot of a computer

Description automatically generated

Text

Description automatically generated 2

**Entity Framework code first Example:**

## **1 – create the Application**

## **2 - Create Models**

Text

Description automatically generated with low confidenceText

Description automatically generated with medium confidence