

# ORM: ENTITYFRAMEWORK CORE

Cours présenté par Tarek Ayari

# ORM ? QU'EST-CE QUE C'EST ?

- ORM signifie **Object-Relational Mapping**.
- Un ORM est une **couche d'abstraction d'accès à la base de données** qui donne l'illusion de ne plus travailler avec des requêtes mais de **manipuler des objets**.
- L'avantage de cette couche d'abstraction est qu'il n'y a plus besoin de se soucier du système de base de données utilisé, c'est l'ORM qui a la charge de transformer les requêtes pour les rendre compatibles avec la base de données.
- Ce niveau d'abstraction supplémentaire fait **correspondre le monde objet** (programmation orientée objet) et le monde **relationnel** (les bases de données relationnelles classiques et massivement utilisées aujourd'hui).
- Plus précisément, **les données vont être normalisées sous forme d'entités** qui auront entre elles trois types d'interactions : **One-To-One, One-To-Many, Many-To-Many**.

# ENTITY FRAMEWORK CORE

- Entity Framework Core est la nouvelle version d'Entity Framework après EF 6.x.
- Il s'agit d'une version open source, légère, extensible et multiplateforme .
- EF Core est destiné à être utilisé avec les applications .NET Core. Cependant, il peut également être utilisé avec les applications standard basées sur le framework .NET 4.5+.

EF Core sur GitHub : <https://github.com/aspnet/EntityFrameworkCore>

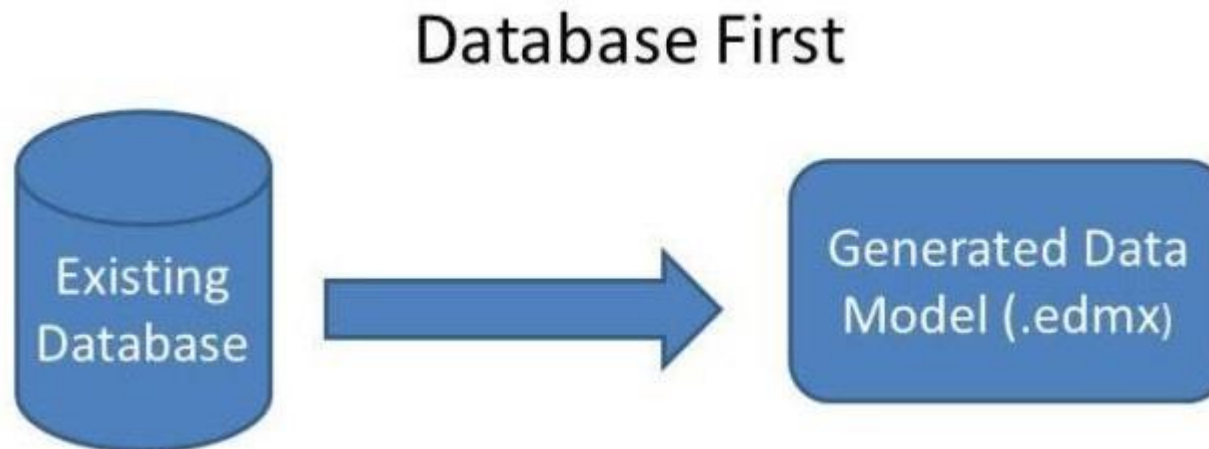
Version de EF Core	Date de sortie	Date Expiration	Framework Cible
EF Core 1.0	Juin 2016	27 Juin 2019	.NET Standard 1.3
EF Core 1.1	Novembre 2016	27 Juin 2019	.NET Standard 1.3
EF Core 2.0	Août 2016	1 Octobre 2018	.NET Standard 2.0
EF Core 3.0	Septembre 2019	3 Mars 2020	.NET Standard 2.1
EF Core 3.1 (LTS)	Décembre 2019	3 Décembre 2022	.NET Standard 2.0
EF Core 5.0	Novembre 2020	10 Mai 2022	.NET Standard 2.1
<b>EF Core 6.0 (LTS)</b>	<b>Novembre 2021</b>	<b>8 Novembre 2024</b>	<b>.NET 6</b>
EF Core 7.0	Novembre 2022 (estimée)	-	.NET 7

Historique des versions EF CORE

# LES DIFFÉRENTES APPROCHES ENTITY FRAMEWORK

## L'approche Database First

Comme son nom l'indique, cette méthode implique que vous créiez votre base de données de A à Z en SQL puis que vous l'importiez dans votre code. C'est une nouvelle fois le Modèle Logique de Données qui fera le lien entre la base de données et l'ORM. La seule différence c'est que cette fois-ci c'est Visual Studio qui va générer le schéma par une procédure de rétro ingénierie.

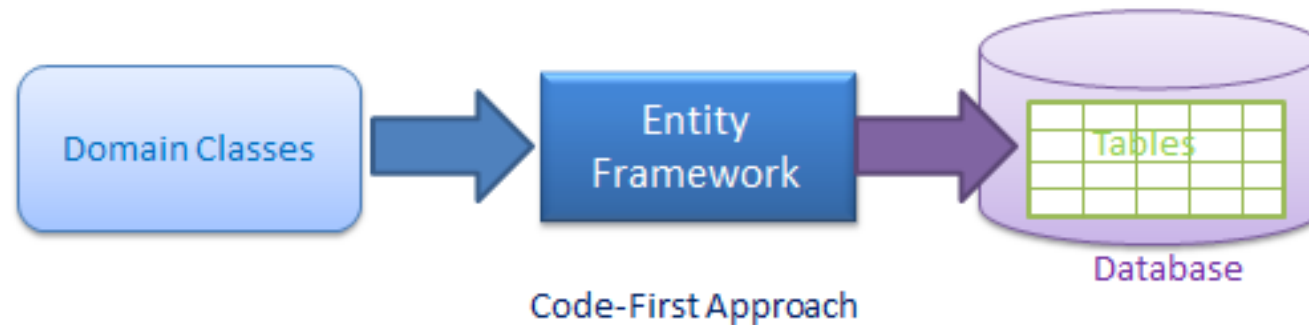


# LES DIFFÉRENTES APPROCHES ENTITY FRAMEWORK

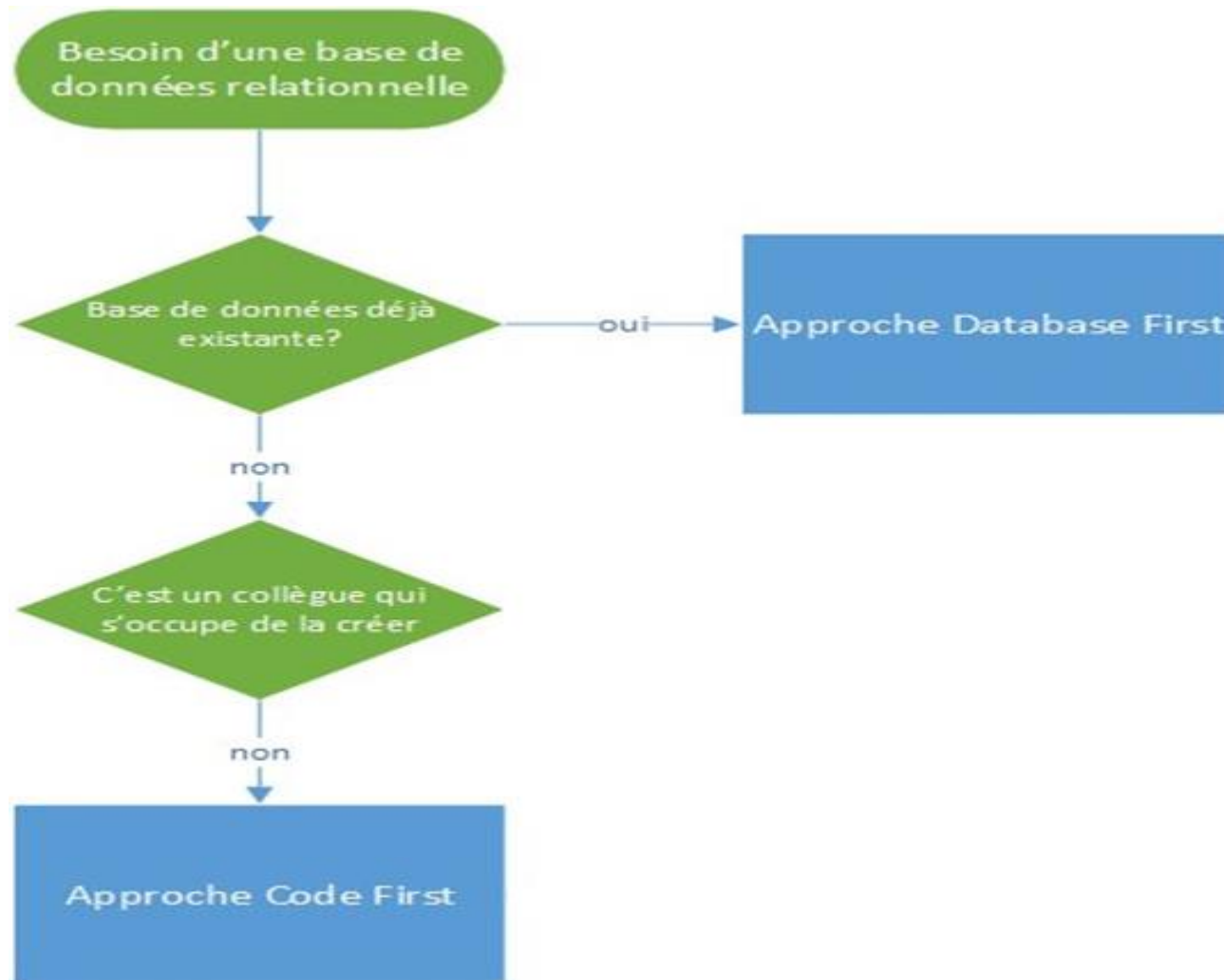
## L'approche Code First

Cette dernière approche, qui est celle que nous avons choisi, part d'un constat : ce qu'un développeur sait faire de mieux, c'est coder.

Cette approche, qui est la plus commune dans le monde des ORM consiste à vous fournir trois outils pour que vous, développeur, n'ayez qu'à **coder des classes** comme vous le feriez habituellement et **en déduire le modèle** qui doit être généré.



# COMMENT CHOISIR SON APPROCHE



# FOURNISSEURS DE BASE DE DONNÉES DE BASE EF

Entity Framework Core utilise un modèle de fournisseur pour accéder à de nombreuses bases de données différentes. EF Core inclut des fournisseurs en tant que packages NuGet que vous devez installer.

Le tableau suivant répertorie les fournisseurs de bases de données et les packages NuGet pour EF Core.

Base de données	Forfait NuGet
serveur SQL	<a href="#"><u>Microsoft.EntityFrameworkCore.SqlServer</u></a>
MySQL	<a href="#"><u>MySql.Data.EntityFrameworkCore</u></a>
PostgreSQL	<a href="#"><u>Npgsql.EntityFrameworkCore.PostgreSQL</u></a>
SQLite	<a href="#"><u>Microsoft.EntityFrameworkCore.SQLite</u></a>
SQL Compact	<a href="#"><u>EntityFrameworkCore.SqlServerCompact40</u></a>
En mémoire	<a href="#"><u>Microsoft.EntityFrameworkCore.InMemory</u></a>

# INSTALLER ENTITY FRAMEWORK CORE

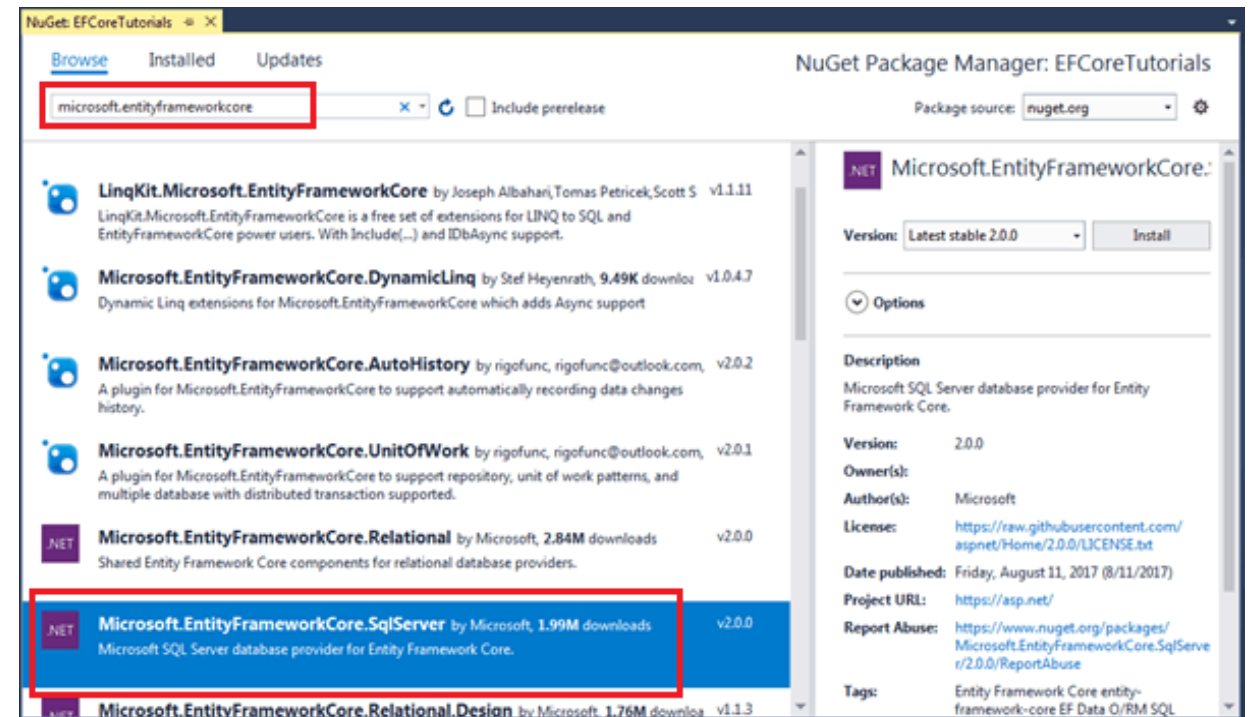
EF Core ne fait pas partie de .NET Core et du framework .NET standard.

Il est disponible sous forme de package NuGet. Vous devez installer les packages NuGet pour les deux éléments suivants afin d'utiliser EF Core dans votre application :

1. EF Core
2. Fournisseur de base de données EF Core
3. Outils de EF Core

## Installer le fournisseur de base de données EF Core

Nous voulons accéder à la base de données MS SQL Server, nous devons donc installer le `Microsoft.EntityFrameworkCore.SqlServerpackage` NuGet



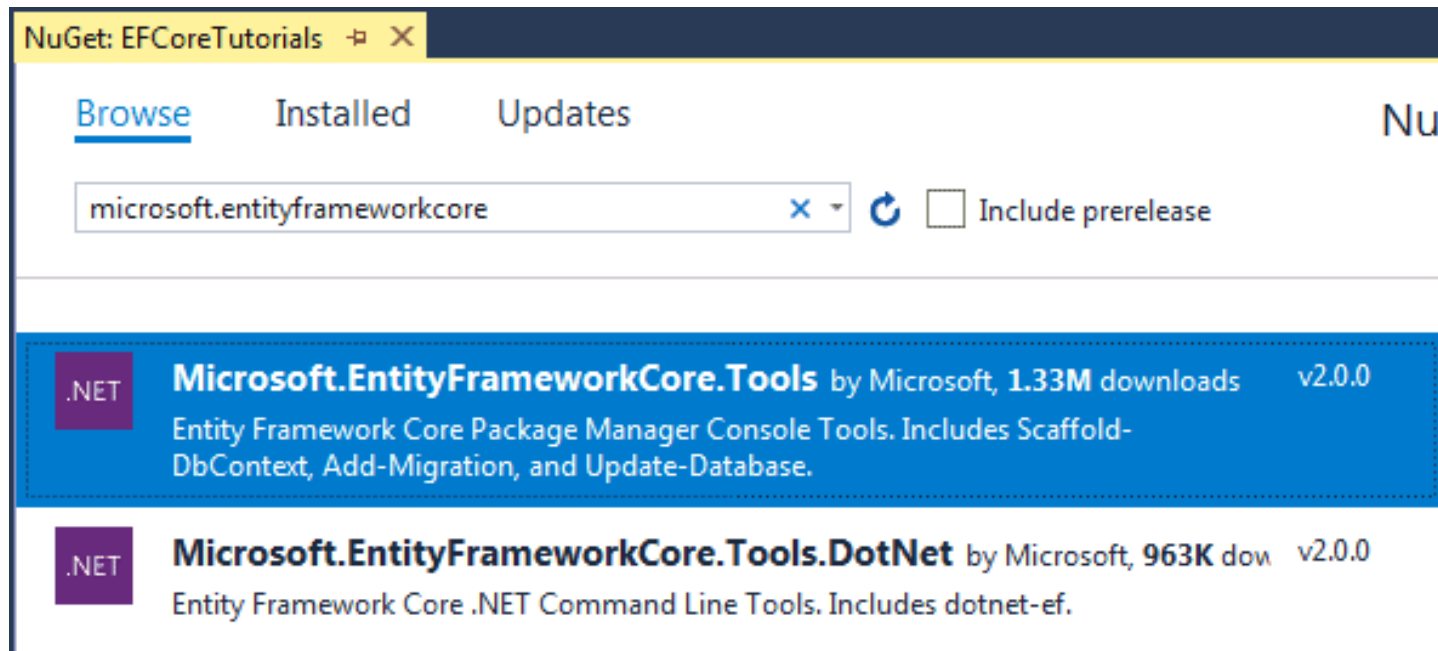


# INSTALLER ENTITY FRAMEWORK CORE

## Installer les outils de base EF

Vous devez également installer les outils EF pour exécuter les commandes EF Core.

Ceux-ci facilitent l'exécution de plusieurs tâches liées à EF Core dans votre projet au moment de la conception, telles que les migrations



# UNE API DE DESCRIPTION

Quand vous utilisez un ORM, il faudra parfois décrire les relations entre les classes. En effet, certaines sont complexes et peuvent être interprétées de plusieurs manières, il vous faudra donc dire explicitement ce qu'il faut faire.

Vous pouvez aussi vouloir nommer vos champs et tables autrement que ce qu'indique la convention.

Pour cela, vous avez deux grandes méthodes avec EntityFrameworkCore.

## Les annotations

Les annotations de données ajoutent une signification supplémentaire aux données en ajoutant des balises d'attribut. Il est utilisé pour configurer les classes qui mettront en évidence les configurations .

Les annotations les plus importantes seront :

- **[Index]** : elle permet de s'assurer qu'une entrée est unique dans la table, et ainsi accélérer fortement les recherches.
- **[Key]** : si votre **clef primaire** ne doit pas s'appeler ID ou bien qu'elle est plus complexe, il faudra utiliser l'attribut Key.
- **[Required]**: il s'agit du même [Required] que pour les vues1, il a juste un autre effet : il interdit les valeurs NULL, ce qui est une contrainte très forte.

# UNE API DE DESCRIPTION

## La Fluent API

La seconde méthode a été nommée "Fluent API". L'API Fluent permet de coder les relations entre les entités. Cela veut dire que le développeur va utiliser des méthodes d'extensions, fournies par le framework, pour mapper le modèle à la base de données et définir les clés primaires, clés étrangères.

La Fluent API est beaucoup plus avancée en termes de fonctionnalités et vous permet d'utiliser les outils les plus optimisés de votre base de données.

## Les migrations Code First

Migrations Code First est la méthode recommandée pour faire évoluer votre schéma de base de données d'application

Les migrations sont de simples fichiers de code (on ne vous a pas menti en disant que c'était du Code First) qui expliquent à la base de données comment se mettre à jour au fur et à mesure que vous faites évoluer votre modèle.

Les migrations fournissent un ensemble d'outils pour les opérations suivantes :

- Créer une base de données initiale qui fonctionne avec votre modèle EF
- Générer des migrations pour suivre les changements que vous appliquez à votre modèle EF
- Mettre à jour votre base de données avec ces changements