

ADO.Net Entity Framework

Nouhaila Bensalah

Chercheuse en IA/NLP

nouhaila.bensalah@etu.fstm.ac.ma



1 Entity Framework

- Le fonctionnement de Entity Framework
- L'architecture de Entity Framework
- Les différentes approches pour créer une base de données

2 Database First approach

1 Entity Framework

- Le fonctionnement de Entity Framework
- L'architecture de Entity Framework
- Les différentes approches pour créer une base de données

2 Database First approach

ORM

- Programme informatique jouant le rôle du traducteur entre le modèle relationnel et le modèle objet.
- Permet d'interroger et manipuler les données à partir d'une base de données à l'aide d'un paradigme orienté objet.
- Deux composants dans les ORM :
 - Entités : Instanciation d'une classe (étudiant, enseignant, cours, projet, etc).
 - Gestionnaire d'entités: à utiliser pour persister les entités dans la base de données.

- Entity Framework est un framework ORM open source pour les applications .NET prises en charge par Microsoft permettant de créer une couche d'accès aux données liées à une base de données relationnelle.
- permettant aux développeurs de manipuler des données à l'aide d'objets de classes C# sans se concentrer sur les tables et colonnes d'une base de données (où ces données sont stockées).
- permettant aux développeurs de créer et maintenir des applications orientées données avec moins de code, par rapport aux applications traditionnelles, grâce à LinQ To Entities.

1 Entity Framework

- Le fonctionnement de Entity Framework
- L'architecture de Entity Framework
- Les différentes approches pour créer une base de données

2 Database First approach

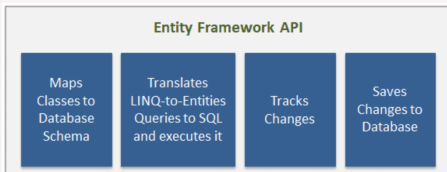
Le fonctionnement de Entity Framework

Dans le cas d'Entity Framework

- Entité (Une classe qui correspond à une table de base de données) = POCO (Plain Old CLR Object) + [classe de configuration ou décorateurs]
- Le gestionnaire d'entités : Linq to Entities

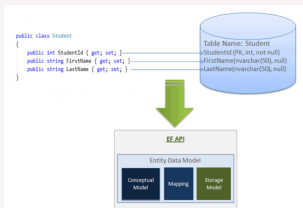
Fonctionnement de Entity Framework

L'API Entity Framework a la possibilité de:



EDM: Entity Data Model

EDM est une représentation en mémoire de l'ensemble des métadonnées.

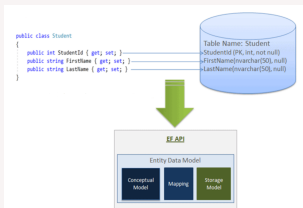


- EDM consiste en trois parties: Conceptuelle, Mapping et Stockage.
 - Conceptuelle : contient les classes du modèle et ses relations.
 - Stockage : contient le modèle physique de la base : tables, vues, procédures stockées, les relations et les clés.
 - Mapping : définit les mécanismes de passage du modèle conceptuel au stockage.

Le fonctionnement de Entity Framework

Maps Classes to Database scheme

EDM est une représentation en mémoire de l'ensemble des métadonnées.



A l'aide de EDM, EF peut:

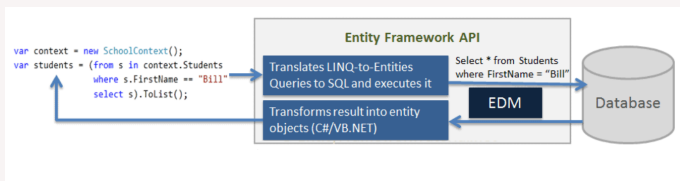
- Effectuer des opérations **CRUD** (Create, Read, Update and Delete).
- Créer des requêtes **SQL** à partir de requêtes **LINQ**, créer des commandes **INSERT**, **UPDATE** et **DELETE** et transformer le résultat de la base de données en objets d'entité.

Le fonctionnement de Entity Framework

Translates LINQ-to-Entities Queries to SQL

A l'aide d'EDM, l'API EF:

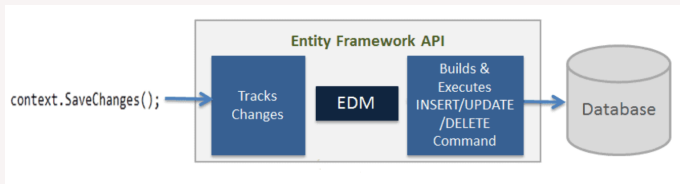
- Traduit les requêtes LINQ-to-Entities en requêtes SQL pour les bases de données relationnelles.
 - LinQ to Entity : Un langage de requête pour un modèle orienté objet. Il retourne les entités définies dans le modèle conceptuel.
- Reconvertit les résultats (requêtes SQL) en objets d'entité.



Le fonctionnement de Entity Framework

Tracks and saves changes to database

- Lorsque la méthode **SaveChanges()** est appelée, l'API EF déduit les commandes **INSERT**, **UPDATE** et **DELETE** en fonction de l'état des entités.
- Le **ChangeTrack** suit les changements survenus sur les entités.



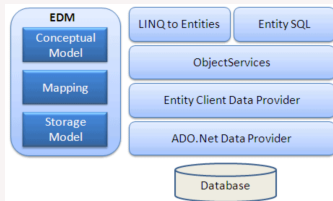
1 Entity Framework

- Le fonctionnement de Entity Framework
- L'architecture de Entity Framework
- Les différentes approches pour créer une base de données

2 Database First approach

L'architecture de Entity Framework

- EDM and LinQ to Entities
- Object Service : est le point d'entrée pour l'accès aux données d'une base de données. Représente le processus de conversion des données retournées par entity client data provider à une entité de structure objet.
- Entity Client Data Provider : la tâche principale est de transformer une expression de LinQ to Entity à une requête SQL.
- ADO.Net Data Provider : communiquer avec la base en utilisant le standard ADO.Net.



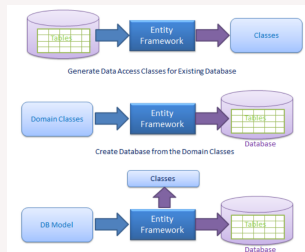
1 Entity Framework

- Le fonctionnement de Entity Framework
- L'architecture de Entity Framework
- Les différentes approches pour créer une base de données

2 Database First approach

Troix approches

- 1 Database First : On crée la base de données (où on a une base de données qui existe déjà) et Entity Framework génère nos entités à partir de cette base de données.
- 2 Code First : On crée les entités puis et Entity Framework génère la base de données.
- 3 Model First : On crée notre modèle (de classe) et Entity Framework génère la base de données et les entités correspondantes.



1 Entity Framework

- Le fonctionnement de Entity Framework
- L'architecture de Entity Framework
- Les différentes approches pour créer une base de données

2 Database First approach

Database First approach

Soit la base de données (BD_ACHAT) définie par le modèle relationnel suivant : **CLIENT** (numcli, nomcli, ville, categorie, compte)

- 1 Chaque ligne décrit un client ; les colonnes décrivent successivement le numéro du client (numcli), son nom (nomcli), sa ville (ville), sa catégorie (categorie) et l'état de son compte (compte). L'identifiant primaire est constitué de numcli.
- 2 Les enregistrements:

NOMCLI	VILLE	CATEGORIE	COMPTE
AMINE	RABAT	A	11250.00
DRISS	CASABLANCA	A	12300
IMANE	MARRAKECH	D	20
YASSINE	TANGER	C	100
ZINEB	CASABLANCA	B	1000
MAROUANE	KENITRA	D	15
HOUDA	RABAT	A	15000
ALI	TANGER	NULL	0
IMANE	RABAT	B	1500
DRISS	CASABLANCA	NULL	0

Table: Table CLIENT

Database First approach

- Installer MS SQL (Microsoft Structured Query Language) et SSMS (SQL Server Management Studio). See [This tutorial](#).
- Créer une base de données (BD_ACHAT) sous SSMS
 - create database BD_ACHAT; use BD_ACHAT;
 - create table CLIENT
(NUMCLI int not null IDENTITY(1, 1),
NOMCLI varchar(25) not null,
VILLE varchar(25) not null,
CATEGORIE char(1),
COMPTE decimal(9,2),
CONSTRAINT pk_cl_ncli primary key (NUMCLI));
 - insert into CLIENT (NOMCLI, VILLE , CATEGORIE, COMPTE) values
('AMINE' , 'RABAT' , 'A' , 11250.00),
('DRISS' , 'CASABLANCA' , 'A' , 12300),
('IMANE' , 'MARRAKECH' , 'D' , 20),
('YASSINE' , 'TANGER' , 'C' , 100),
('ZINEB' , 'CASABLANCA' , 'B' , 1000);

Étapes à suivre

- Créer un nouveau projet sous Visual Studio **Fichier** → **Nouveau** → **Projet**.
- Cliquer sur **Installé** et choisir **C#**.
- Étendre la rubrique **Web** et sélectionner **Application web ASP.NET (.NET Framework)**.
- Remplir le champs **Nom** par: **App_ACHAT**.
- Sélectionner un modèle **MVC**.
- Valider et attendre la fin de création du projet.

EF designer à partir de la base de données

- Faire un clic droit sur **Models** et aller dans **Ajouter** → **Nouvel élément** .
- Sélectionner **ADO.NET Entity Data Model** et cliquer sur **Ajouter**.
- Saisir un nom (**Model1**) et cliquer sur **Ajouter**.
- Les quatres options disponibles:
 - EF Designer from database pour the database-first approach
 - Empty EF Designer model pour the model-first approach
 - Empty Code First model et Code First from database pour Code-First approach.
- Sélectionner **EF designer à partir de la base de données** pour le contenu du modèle et cliquer sur **Ajouter**.

Connecter le serveur sql avec visual studio

- Faire un clic sur **Nouvelle connexion** et saisir le nom du serveur (Faire un clic sur Object explorer dans SSMS).
- Sélectionner le nom de la base de données **BD_ACHAT**.

Choix des objets(table)

- Sélectionner les objets(les tables) que vous voulez inclure dans votre modèle (**CLIENT**).
- Cocher **Mettre au pluriel ou au singulier les noms des objets générés** et cliquer sur **Terminer**.