

## TP N° 7

### Model First avec Entity Framework

Dans le but de réaliser une application web pour la gestion des notes des étudiants, en utilisant l'approche **Model First** avec Entity Framework.

Ce TP suivra les étapes décrites dans les parties ci-dessous, et permettra de construire progressivement un modèle relationnel riche, directement à partir du designer **EDMX** dans Visual Studio.

Les entités métier concernées sont :

- Etudiant
- Groupe
- Matiere
- Professeur
- Note

### Création du projet ASP.NET MVC

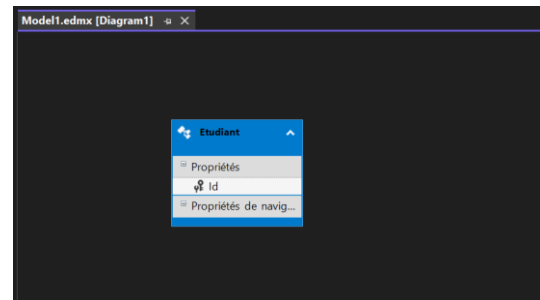
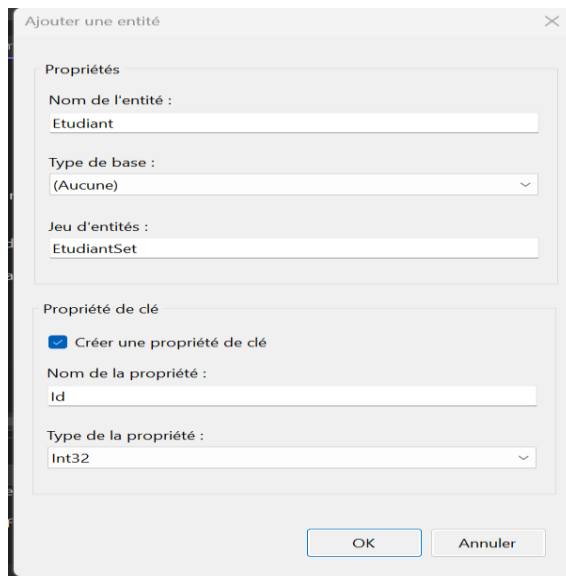
- Créer un nouveau projet sous Visual Studio
- Choisir le type de projet ASP.NET Web Application (.NET Framework)
- Nommer le projet **GestionEtudiant**
- Choisir le modèle MVC par la suite
- Et créer le projet

### Partie 1 : Création du modèle de données (Model1.edmx)

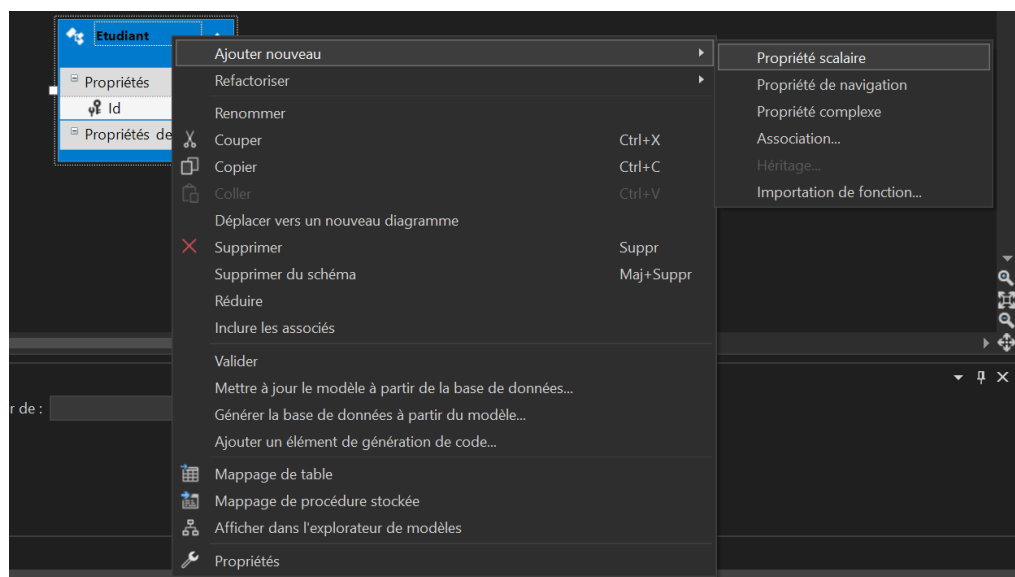
1. Clic droit sur le dossier Models → Ajouter > Nouvel Element
2. Choisir : **ADO.NET Entity Data Model**
3. Nom du fichier : **Model1.edmx**
4. Choisir : **modèle vide EF Designer**
5. Cliquer sur Terminer

### Partie 2 : Modélisation des entités Etudiant et Groupe

1. Visualiser Model1.edmx
2. Clique droit dans la fenêtre apparue et ajouter nouveau > Entité
3. Ajouter les éléments suivants :



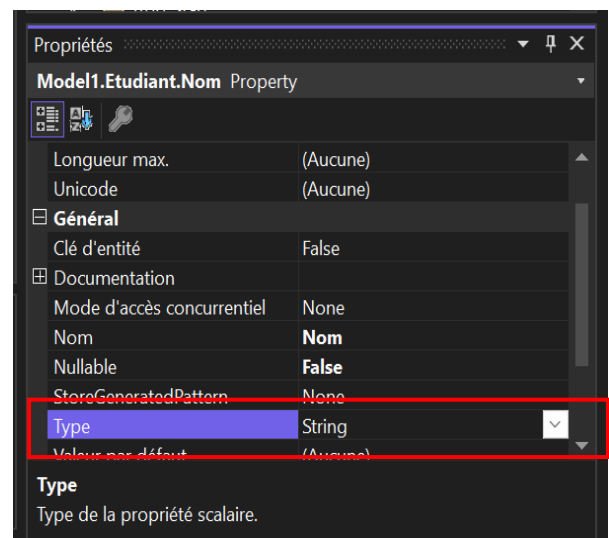
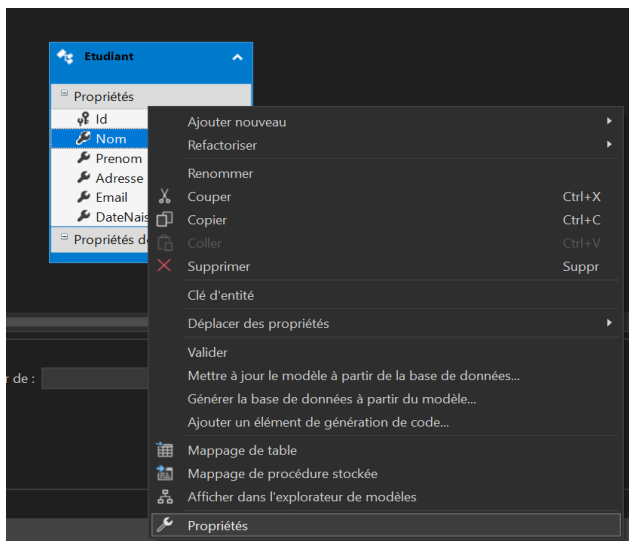
4. Ajouter les propriétés de l'entité Etudiant selon les étapes suivantes : Cliquez droit puis ajouter nouveau > Propriété scalaire



## Entité Etudiant

- Clé primaire : Id (Identity)
  - Propriétés :
    - Nom (string)
    - Prenom (string)
    - Adresse (string)
    - Email (string)
    - DateNaissance (DateTime)
5. Modifier le type de chaque propriété :
    - Cliquer sur la propriété désirée, puis cliquez droit > Propriétés et puis modifier le type
    - Email peut être nullable = true

- Prenom, Nom → nullable = false



Vous pouvez limiter la longueur maximale d'un string, exemple :

- **Champ** : Longueur max.
- Exemple : mets 100 pour limiter Prenom à 100 caractères en base de données.

Ajouter une valeur par défaut

- **Champ** : Valeur par défaut
- Par exemple : « N/A » ou 0 selon le type

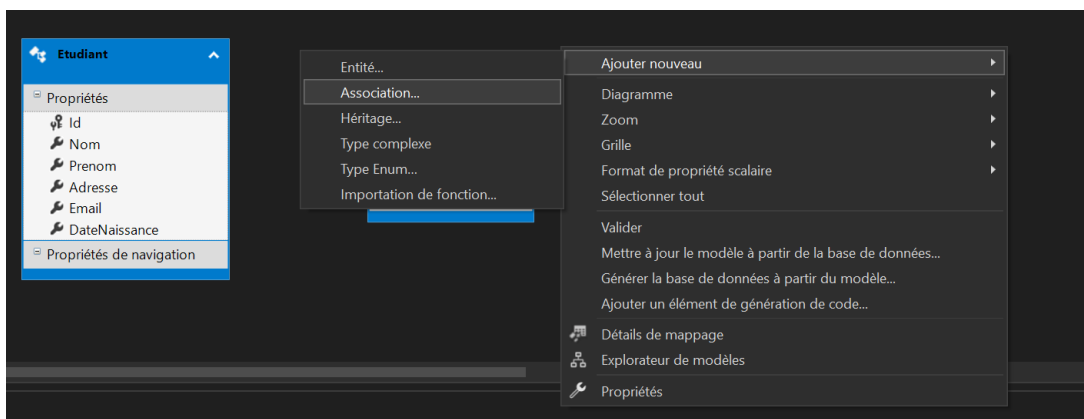
6. De même manière ajouter l'entité Groupe

**Entité Groupe :**

- Clé primaire : Id (Identity)
- Propriété :
  - Nom (string)

7. Ajouter Relation Groupe (1) → Etudiant (N)

- Clic droit sur le designer > Ajouter Nouveau > Association



- De : Groupe → 1

- À : Etudiant → \*
- Cocher : Ajouter clé étrangère  
⇒ Cela ajoutera GroupeId dans Etudiant et la navigation Etudiant.Groupe, Groupe.Etudiants

Ajouter une association

Nom de l'association : EtudiantGroupe

Terminaison

Entité : Etudiant

Multiplicité : \* (plusieurs)

☒ Propriété de navigation : Groupe

Terminaison

Entité : Groupe

Multiplicité : 1 (un)

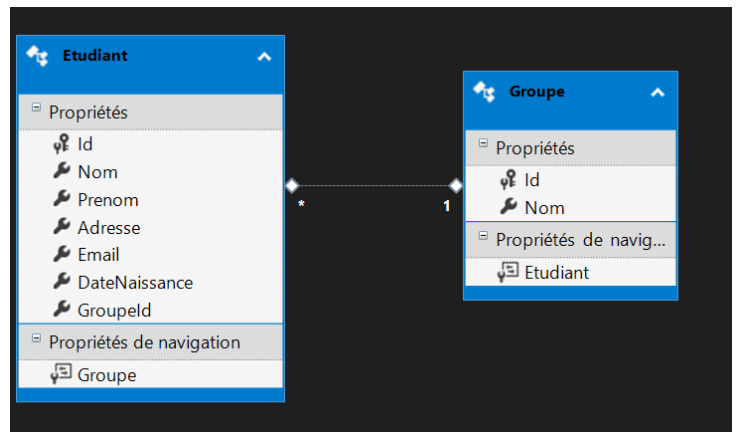
☒ Propriété de navigation : Etudiant

☒ Ajouter les propriétés de la clé étrangère à l'entité 'Etudiant'

Etudiant peut avoir 1 (un) instance de Groupe. Utilisez Etudiant.Groupe pour accéder à l'instance Groupe.

Groupe peut avoir \* (plusieurs) instances de Etudiant. Utilisez Groupe.Etudiant pour accéder aux instances Etudiant.

OK Annuler



### Partie 3 : Modélisation des entités Matière et Professeur

1. Ajouter les entités suivantes :

#### **Entité Matière :**

- Clé primaire : Id (Identity)
- Propriétés :
  - Nom (string)
  - Coefficient (int)

#### **Entité Professeur :**

- Clé primaire : Id (Identity)
- Propriétés :
  - Nom (string)
  - Prenom (string)
  - Email (string)
  - Specialite (string)

#### **Relation Professeur (1) → Matière (N)**

- Clic droit sur le designer > Ajouter Nouveau > Association
- De : Professeur → 1
- À : Matière → \*
- Cocher : Ajouter clé étrangère  
⇒ Cela ajoute ProfesseurId dans Matière

Ajouter une association

Nom de l'association :  
ProfesseurMatiere

Terminaison	Terminaison
Entité : Professeur	Entité : Matiere
Multiplicité : 1 (un)	Multiplicité : * (plusieurs)
<input checked="" type="checkbox"/> Propriété de navigation : Matiere	<input checked="" type="checkbox"/> Propriété de navigation : Professeur

☒ Ajouter les propriétés de la clé étrangère à l'entité 'Matiere'

Professeur peut avoir \* (plusieurs) instances de Matiere. Utilisez Professeur.Matiere pour accéder aux instances Matiere.

Matiere peut avoir 1 (un) instance de Professeur. Utilisez Matiere.Professeur pour accéder à l'instance Professeur.

OK Annuler

## Partie 4 : Modélisation de l'entité Note et relations

1. Ajouter l'entité suivante :

### Entité Note :

- Clé primaire : Id (Identity)
- Propriétés :
  - Valeur (double)

2. Ajouter les relations suivantes :

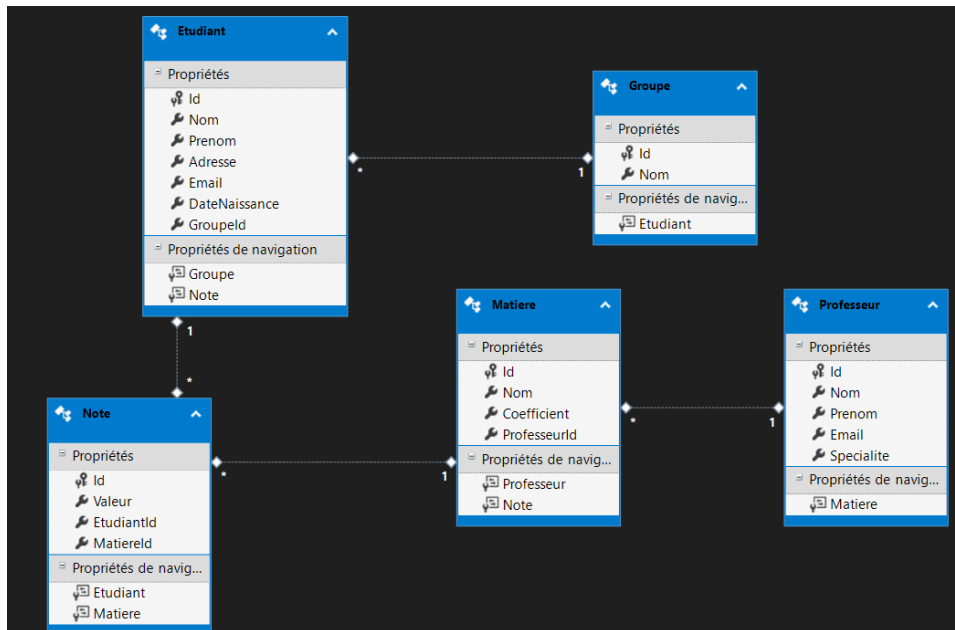
### Relation Etudiant (1) → Note (N)

- Clic droit > Ajouter nouveau > Association
- De : Etudiant → 1
- À : Note → \*
- Cocher : Ajouter clé étrangère  
⇒ Ajoute EtudiantId dans Note

### Relation Matiere (1) → Note (N)

- Même procédure
- De : Matiere → 1
- À : Note → \*  
⇒ Ajoute MatiereId dans Note

Résultat :

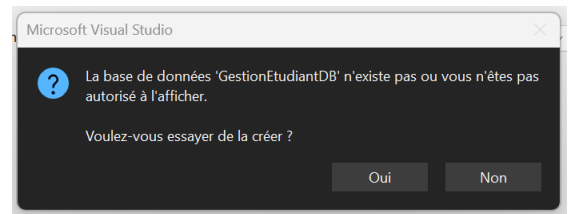
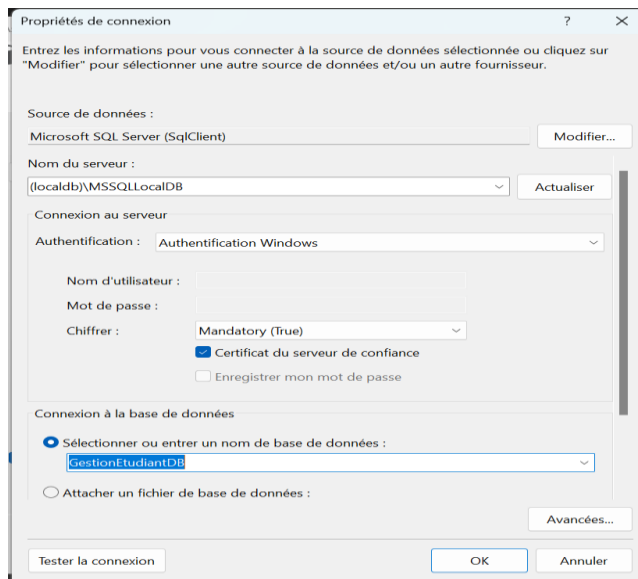


## Partie 5 : Configuration générale du modèle

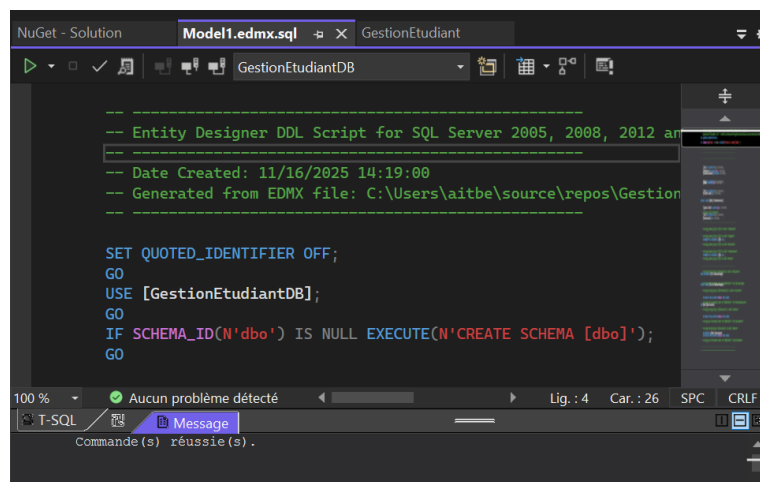
- Cliquer sur l'espace vide du designer
- Ouvrir la fenêtre **Properties**
- Vérifier les propriétés du modèle : namespace, pluralization, etc.
- Renommer les propriétés de navigation si nécessaire
- En cas de problème, ajouter le Package EntityFramework via gestionnaire de package NuGet

## Partie 6 : Génération de la base de données

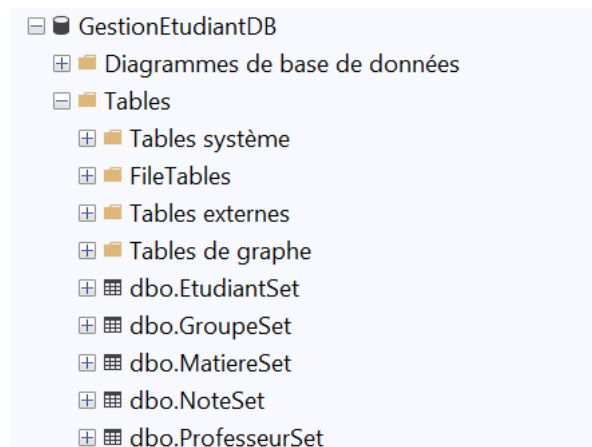
1. Clic droit sur le designer > Générer la base de données à partir du modèle
2. Configurer nouvelle connexion SQL Server
3. Cliquer sur Suivant
4. Connecter au serveur utilisé
5. Entrer le nom de la base de données : **GestionEtudiantDB**



6. Visual Studio génère un script SQL (Model1.edmx.sql)
7. Cliquer sur Terminer
8. Clic droit sur le fichier .sql > Exécuter



⇒ Une base de données avec les 5 tables est maintenant créée :



## **Partie 7 : Insertion des données dans chaque table (via SQL Server)**

Il faut respecter l'ordre des dépendances (clés étrangères) :

1. GroupeSet
2. ProfesseurSet
3. MatiereSet (dépend de Professeur)
4. EtudiantSet (dépend de Groupe)
5. NoteSet (dépend de Etudiant + Matiere)

```
use GestionEtudiantDB;
```

**GroupeSet :**

```
INSERT INTO GroupeSet (Nom) VALUES  
( 'GI2' ),  
( 'GC1' ),  
( 'RT2' ),  
( 'IA1' );
```

**ProfesseurSet :**

```
INSERT INTO ProfesseurSet (Nom, Prenom, Email, Specialite) VALUES  
( 'El Khatib', 'Ahmed', 'a.khatib@emsi.ma', 'Programmation avancée' ),  
( 'Naciri', 'Salma', 's.naciri@emsi.ma', 'Réseaux informatiques' ),  
( 'El Fassi', 'Amine', 'a.fassi@emsi.ma', 'Mathématiques' ),  
( 'Belkadi', 'Youssef', 'y.belkadi@emsi.ma', 'Génie Civil' );
```

**MatiereSet :**

```
INSERT INTO MatiereSet (Nom, Coefficient, ProfesseurId) VALUES  
( 'Structures de Données', 3, 1 ),  
( 'Programmation Orientée Objet', 4, 1 ),  
( 'Réseaux & Protocoles', 3, 2 ),  
( 'Intelligence Artificielle', 4, 2 ),  
( 'Résistance des Matériaux', 4, 4 ),  
( 'Béton Armé & Construction', 4, 4 ),  
( 'Mathématiques Appliquées', 3, 3 );
```

**EtudiantSet :**

```
INSERT INTO EtudiantSet (Nom, Prenom, Adresse, Email, DateNaissance,  
GroupeId) VALUES  
( 'El Amrani', 'Yassine', 'Casablanca', 'y.elamrani@emsi.ma', '2000-03-15',  
1 ),  
( 'Benslimane', 'Aya', 'Rabat', 'a.benslimane@emsi.ma', '2001-06-20', 2 ),  
( 'Tazi', 'Oussama', 'Fès', 'o.tazi@emsi.ma', '1999-12-01', 3 ),  
( 'Najih', 'Ikram', 'Marrakech', 'i.najih@emsi.ma', '2002-01-10', 4 );
```

**NoteSet :**

```
INSERT INTO NoteSet (Valeur, EtudiantId, MatiereId) VALUES  
( 15.5, 1, 1 ),  
( 16.0, 1, 2 ),  
( 13.0, 2, 5 ),  
( 17.5, 3, 3 ),  
( 18.0, 4, 4 ),  
( 14.5, 4, 7 );
```

## **Partie 8 : Génération des contrôleurs et vues**

Pour chaque entité (Etudiant, Groupe, Matiere, Professeur, Note) :

1. Clic droit sur Controllers > Ajouter > Controller
2. Choisir : MVC 5 Controller with views, using Entity Framework
3. Configurer :
  - Model class : choisir l'entité
  - Data context : Model1Container
4. Cliquer sur Add

Visual Studio génère :

- Un contrôleur avec les méthodes CRUD
- Les vues Razor (Index, Create, Edit, Delete, etc.)

### **Exécution et tests**

- Exécuter le projet
- Accéder aux différentes vues : /Etudiants, /Notes, etc.