

Programmation .Net

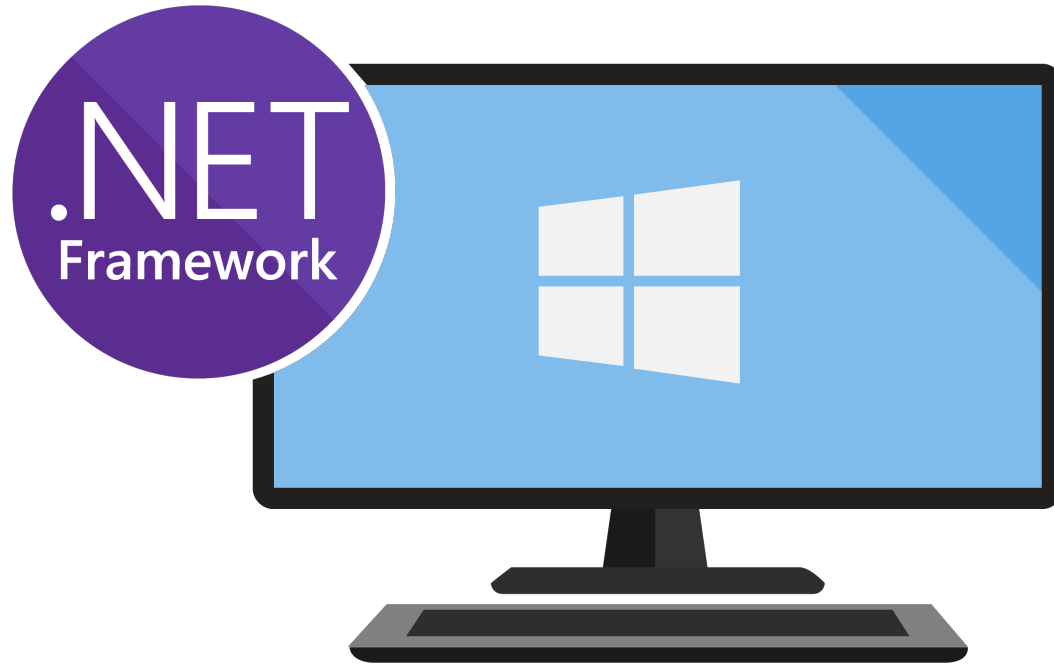
Théorie

Structure du cours

- Théorie 15 heures
- 50%
- Laboratoire
- 50%

Table des matières

- Entity Framework
- LINQ
- Architecture en couche
 - DAL et BLL, WCF
- WPF, MVVM
- ASP.NET



Entity Framework

Objectifs

Sans EF

```
SqlConnection connection = new SqlConnection(connectionString);
```



```
SqlCommand command = new SqlCommand(@"INSERT INTO [CodeTunnel].[Users]  
VALUES (@Username, @Password,  
@Administrator)", connection);
```

```
command.Parameters.Add(new SqlParameter("Username", "Alex"));  
command.Parameters.Add(new SqlParameter("Password", "Exp3rtC0d3r1"));  
command.Parameters.Add(new SqlParameter("Administrator", true));
```

```
connection.Open();
```

```
int rowsAdded = command.ExecuteNonQuery();
```

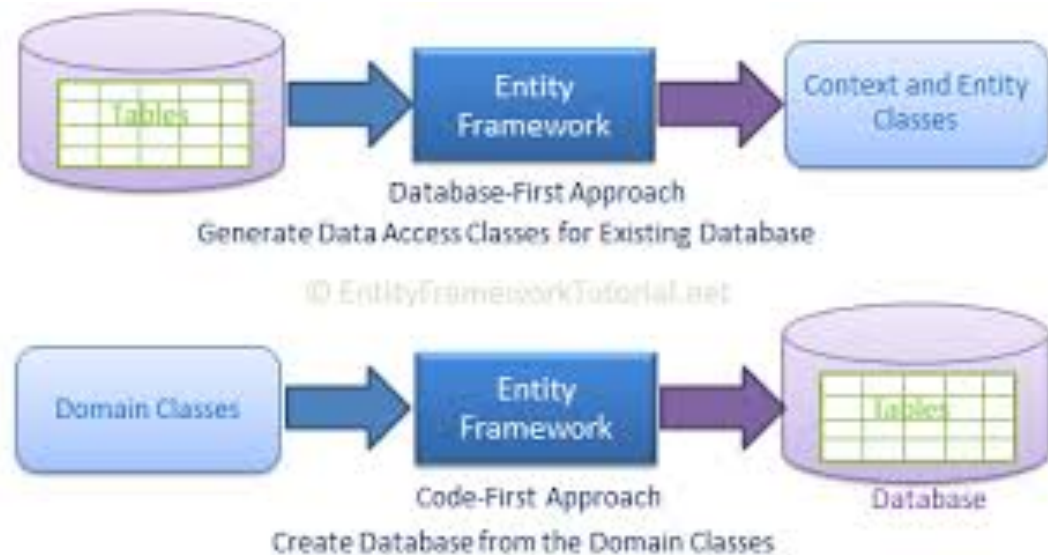
```
connection.Close();
```

Entity Framework

- ORM (Object Relational Mapping)
- Simplification
- Augmenter de la maintenabilité
- Augmenter la portabilité

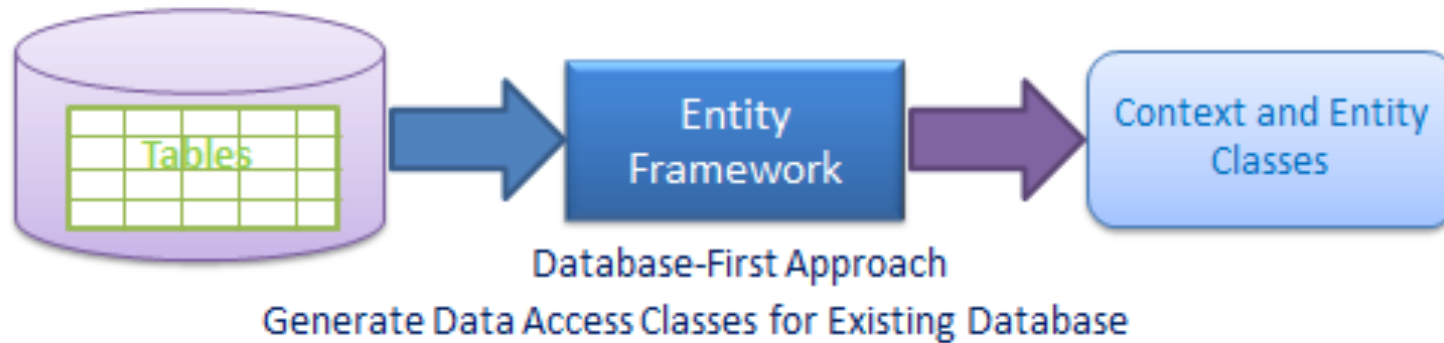
3 Types différents

- Database First
- Model First
- Code First



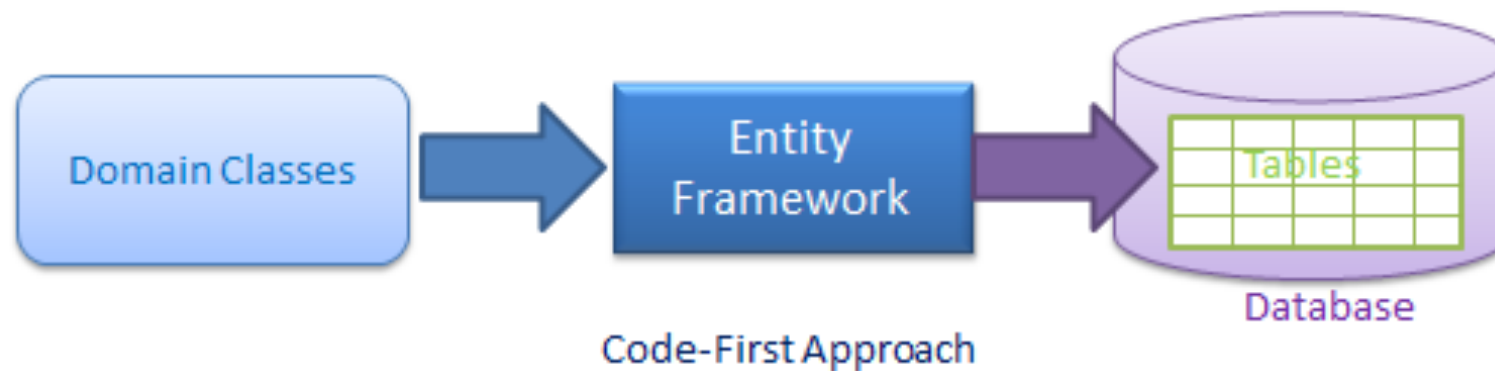
Database First

- À partir d'une base de données
- Application existante



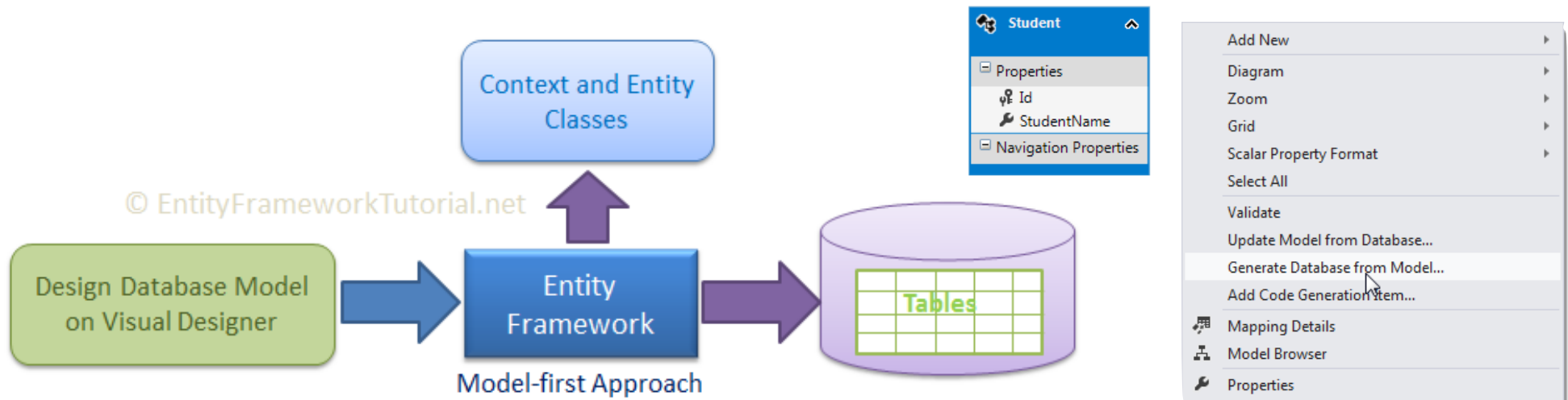
Code First

- À partir du code directement (classes)
- Générer la base de données correspondante



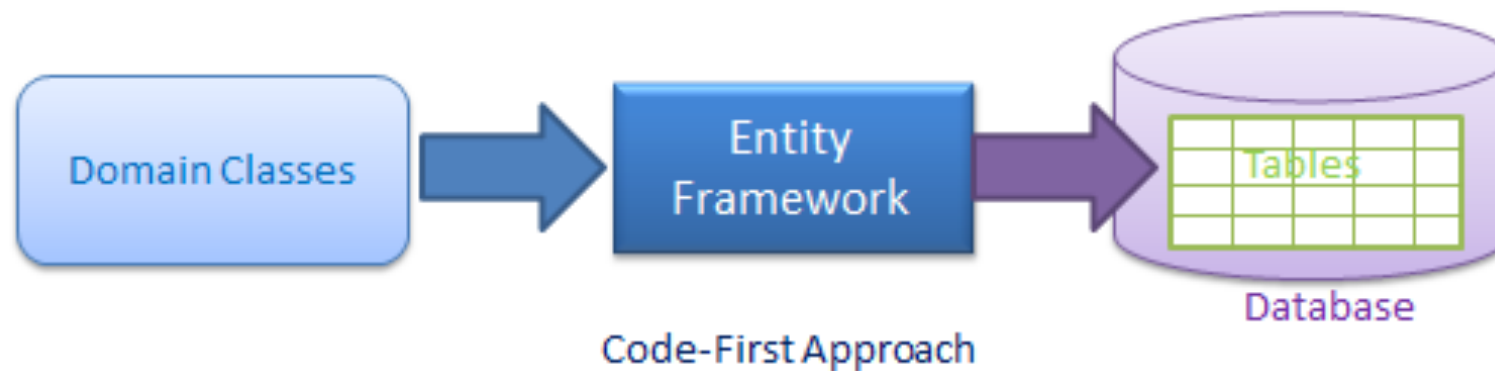
Model First

- À partir d'un programme



Pour le Labo

- Générer à partir des classes la DB
- Remplir la DB par programmation



Comment

Entity Framework

Installation

- Installer SQL Server
- Installer SQL Server Management Studio
- Installer EntityFramework, dans le projet

Essayez SQL Server localement ou dans le cloud



SQL Server 2017 local

Créez des applications intelligentes à l'aide d'une plateforme de données hybride, évolutive pour les scénarios d'usage exigeants. Commencez votre essai gratuit de 180 jours de SQL Server 2017 sous Windows.

[Télécharger la version d'évaluation gratuite ↓](#)



SQL Server dans le cloud

Profitez de la disponibilité élevée intégrée, de la sécurité et de l'intelligence d'Azure SQL Database et utilisez le moteur SQL familier, sans subir la complexité d'une gestion de l'infrastructure. Commencez à utiliser SQL Database gratuitement dans Azure.

[Démarrer gratuitement >](#)

Ou téléchargez une édition spécialisée gratuite



Developer

SQL Server 2017 Developer est une édition gratuite comprenant toutes les fonctionnalités, cédée sous licence pour être utilisée comme base de données de développement et de test dans des environnements non dédiés à la production.

[Télécharger maintenant ↓](#)



Express

SQL Server 2017 Express est une édition gratuite de SQL Server, idéale pour le développement et la production d'applications de bureau, d'applications web et de petites applications serveur.

[Télécharger maintenant ↓](#)

https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017



SQL Docs

Overview

Install

Secure

Develop

Administer

Analyze

Reference

Download SQL Server

Search

Docs / SQL / Tools / SQL Server Management Studio (SSMS) / Download SSMS

Bookmark Feedback Edit Share Theme Sign in

SQL Server 2017

Filter by title

Welcome to SQL Server

SQL docs navigation tips

Previous versions 2014-2005

> Overview

> Business continuity

> Database design

> Development

> Internals & Architecture

> Installation

> Migrate & load data

> Manage, monitor, & tune

> Query data

> Reporting & Analytics

> Security

> Tools

Tools

> Azure Data Studio

> Command prompt utilities

> Database Tuning Advisor (DTA)

> Distributed Replay

> SQL Server Configuration Manager

> SQLCMD

> SSB Diagnose

> SQL Server Data Tools (SSDT)

> SQL Server Management Studio (SSMS)

Download SSMS

Download PDF

Download SQL Server Management Studio (SSMS)

07/26/2019 • 4 minutes to read • +22

APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

SQL Server Management Studio (SSMS) is an integrated environment for managing any SQL infrastructure, from SQL Server to Azure SQL Database. SSMS provides tools to configure, monitor, and administer instances of SQL Server and databases. Use SSMS to deploy, monitor, and upgrade the data-tier components used by your applications, and build queries and scripts.

Use SSMS to query, design, and manage your databases and data warehouses, wherever they are - on your local computer, or in the cloud.

SSMS is free!

Download SSMS 18.2

SSMS 18.2 is now available, and is the latest general availability (GA) version of *SQL Server Management Studio* that provides support for SQL Server 2019!

[Download SQL Server Management Studio 18.2](#)

SSMS 18.2 is the latest general availability (GA) version of SSMS. If you have a previous GA version of SSMS 18 installed, installing SSMS 18.2 upgrades it to 18.2. If you have an older *preview* version of SSMS 18.x installed, you must uninstall it before installing SSMS 18.2.

Version Information

- Release number: 18.2
- Build number: 15.0.18142.0
- Release date: July 25, 2019

If you have comments or suggestions, or you want to report issues, the best way to contact the SSMS team is at [UserVoice](#).

The SSMS 18.x installation doesn't upgrade or replace SSMS versions 17.x or earlier. SSMS 18.x installs side by side with previous versions so both versions are available for use.

If a computer contains side-by-side installations of SSMS, verify you start the correct version for your specific needs. The latest version is labeled **Microsoft SQL Server Management Studio 18**

Is this page helpful?

Yes No

In this article

[Download SSMS 18.2](#)

[Available languages \(SSMS 18.2\)](#)

[New in this release \(SSMS 18.2\)](#)

[Supported SQL offerings \(SSMS 18.2\)](#)

[Supported operating systems \(SSMS 18.2\)](#)

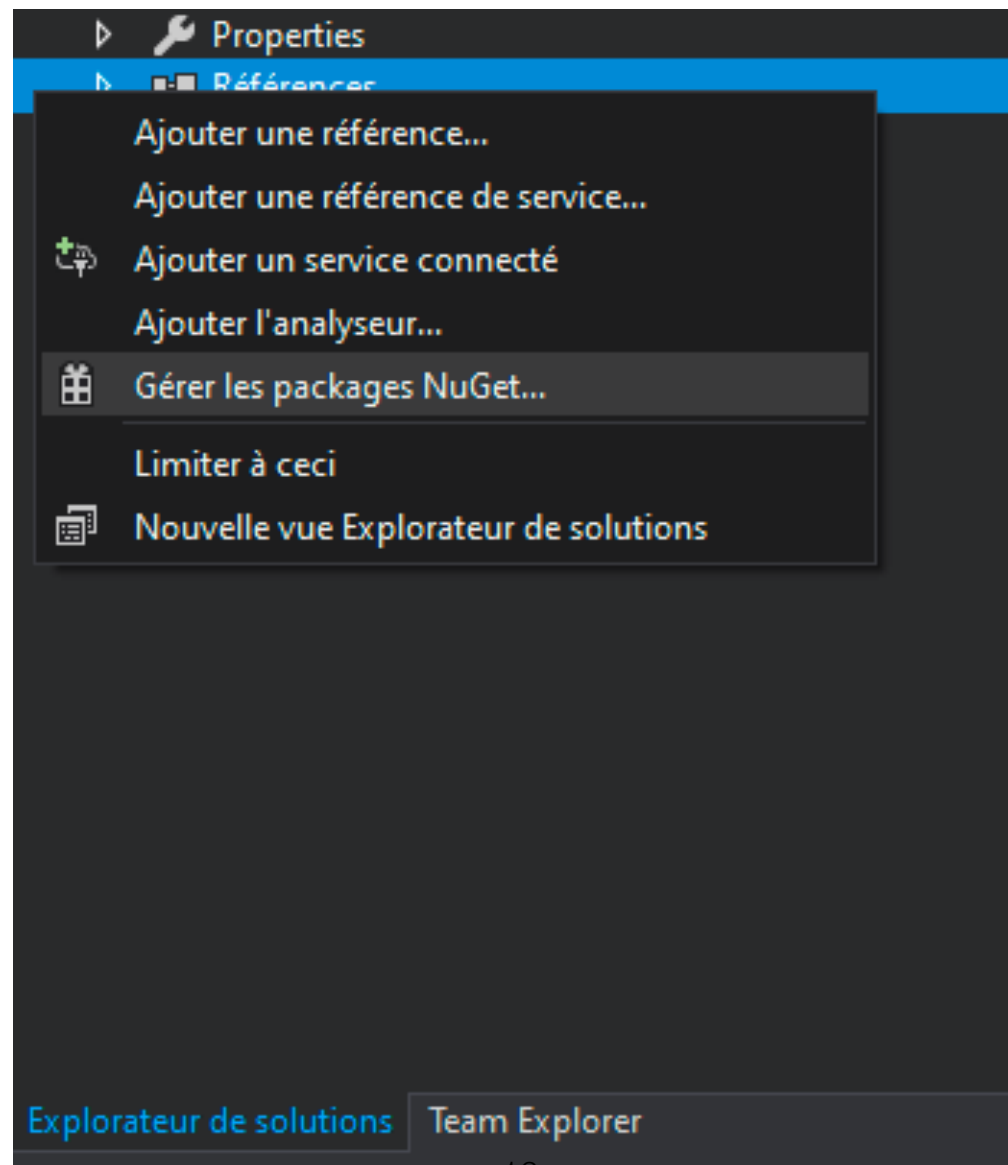
[Release notes \(SSMS 18.2\)](#)

[Previous SSMS releases](#)

[Get help for SQL tools](#)

[See also](#)

[Contribute SQL documentation](#)



Parcourir

Installé

Mises à jour

Recherche (Ctrl+L)

☐ Inclure la version préliminaire

Gestionnaire de package NuGet : Demo1_EntityFramework

Source de package : nuget.org

Newtonsoft.Json

par James Newton-King, 280M téléchargements

v12.0.2

Microsoft.Extensions.Logging

par Microsoft, 103M téléchargements

v2.2.0

Microsoft.Extensions.DependencyInjection

par Microsoft, 91.4M téléchargements

v2.2.0

Castle.Core

par Castle Project Contributors, 72.6M téléchargements

v4.4.0

EntityFramework

par Microsoft, 71.2M téléchargements

v6.2.0

Moq

par Daniel Cazzulino, kzu, 70.1M téléchargements

v4.13.0

jQuery

par jQuery Foundation, Inc., 70M téléchargements

v3.4.1

Microsoft.AspNet.Mvc

par Microsoft, 65.8M téléchargements

v5.2.7

WindowsAzure.Storage

par Microsoft, 57.1M téléchargements

v9.3.3

Chaque package vous est concédé sous licence par son propriétaire. NuGet n'est pas responsable des packages tiers et n'octroie aucune licence les concernant.

☐ Ne plus afficher ce message

EntityFramework

nuget.org

Version :

Dernière version stable 6.2.0

Installer

Optimiser

6.2.0

6.1.3

6.1.2

6.1.1

6.1.0

6.0.2

6.0.1

6.0.0

5.0.0

4.3.1

4.3.0

4.2.0

4.1.10715

4.1.10331

Descriptif

Entity Framework is Microsoft's recommended data access technology for new applications.

Auteur(s)

6.0.0

Licence :

5.0.0

Date de publication

4.3.1

URL du projet

4.3.0

Signaler un problème

4.2.0

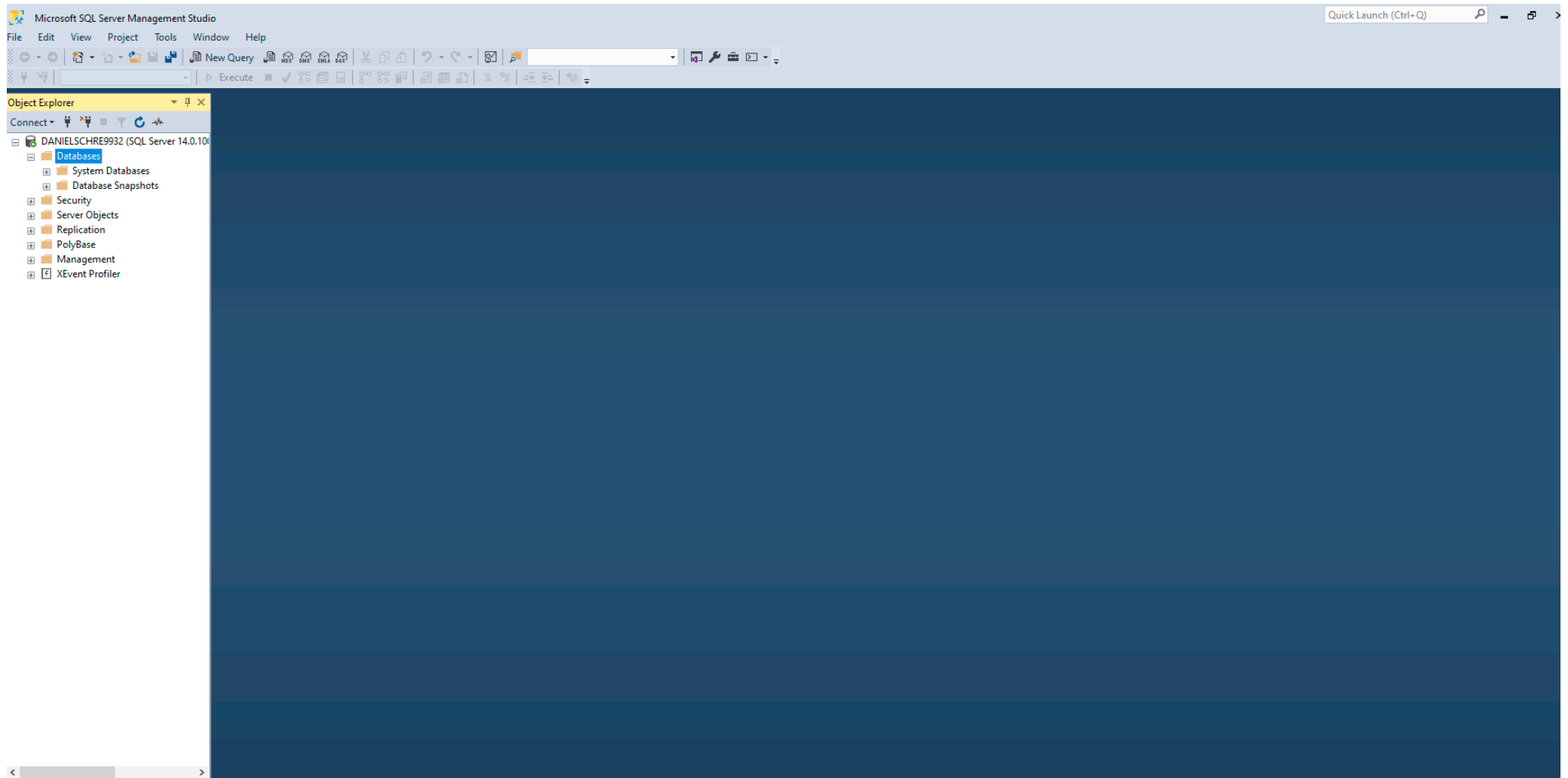
Balises :

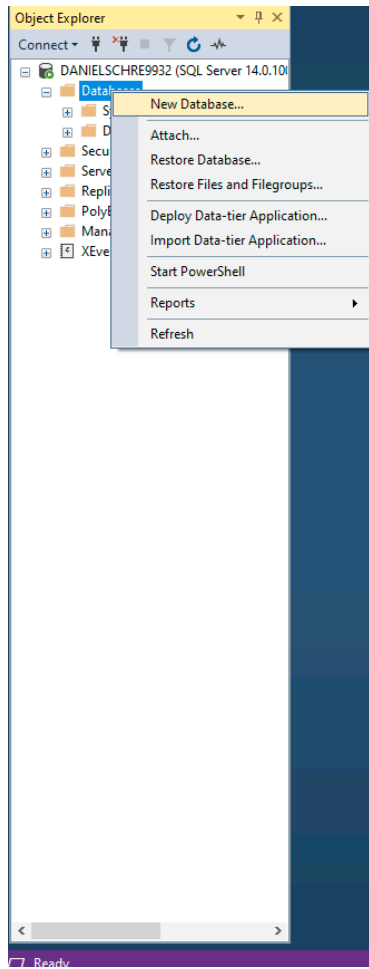
O/RM, Database, Microsoft, EntityFramework, EF, ADO.NET, Data

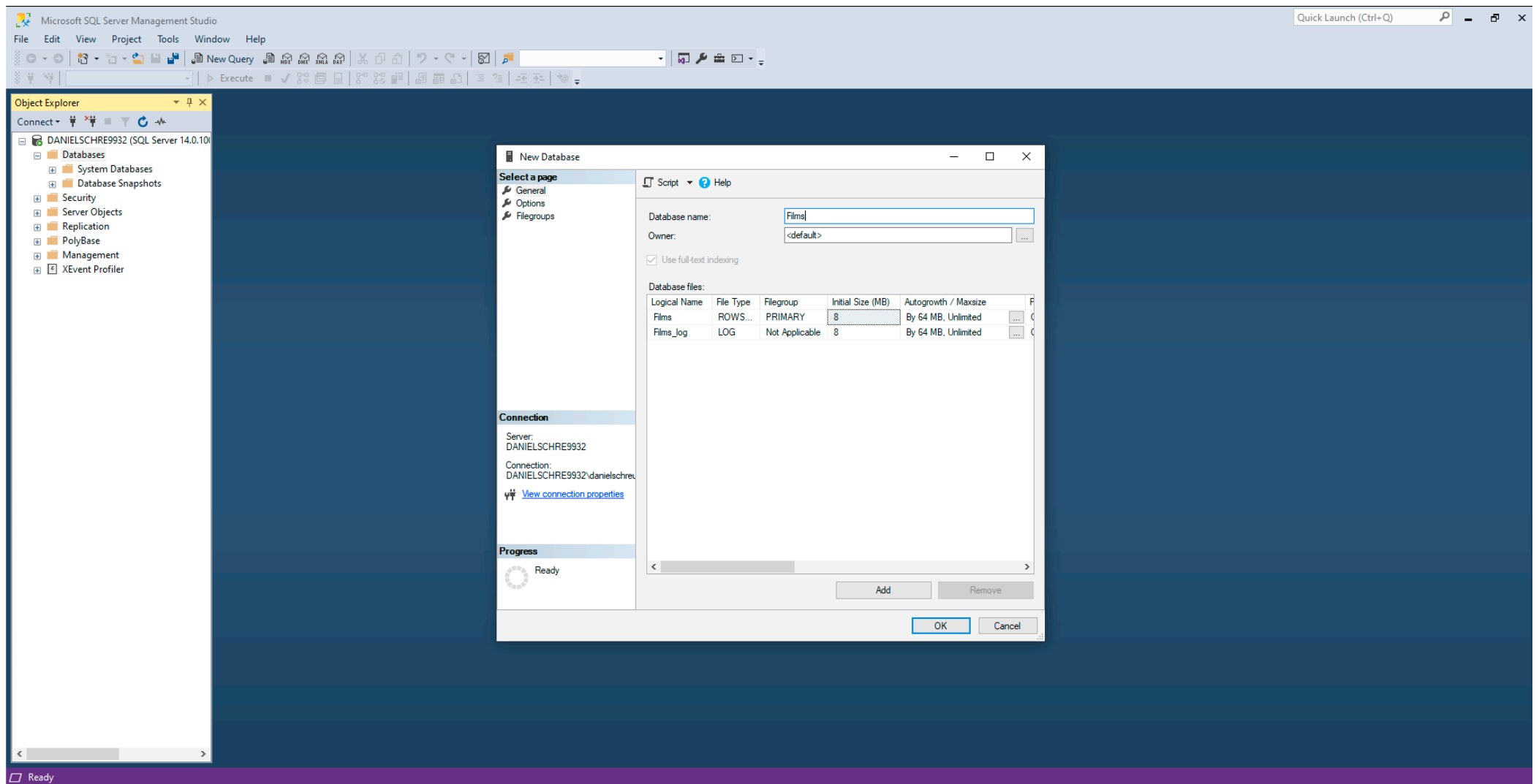
Dépendances

Aucune dépendance

17







Les classes

Entity Framework

Connexion vers la BD

- Une classe
- Dérive de DbContext
- Avec les propriétés pour chacune des tables

Connexion vers la BD

```
public class SchoolContext : DbContext
{
    public SchoolContext() : base("DB-NAME")
    {
        Database.SetInitializer<SchoolContext>(new SchoolDBInitializer());
    }

    public DbSet<Student> Students { get; set; }
    public DbSet<Grade> Grades { get; set; }
    public DbSet<Course> Courses { get; set; }
    public DbSet<StudentAddress> StudentAddresses { get; set; }
}
```

Un manager

- Possède une DbContext
- Méthodes d'accès à la DB
- Et enregistrement des modifications

BdManager (DAL)

```
public class BdManager
{
    public BdManager()
    {
        SchoolContext = new SchoolContext();
    }

    public ICollection<Student> GetNStudentFrom(int from, int count)
    {
        return SchoolContext.student.
            OrderBy(student => student.StudentName).
            Skip(from).
            Take(count).
            ToList();
    }
}
```


Les entités

- Représente une table de la BD
- POCO (Plain Old C# Object)

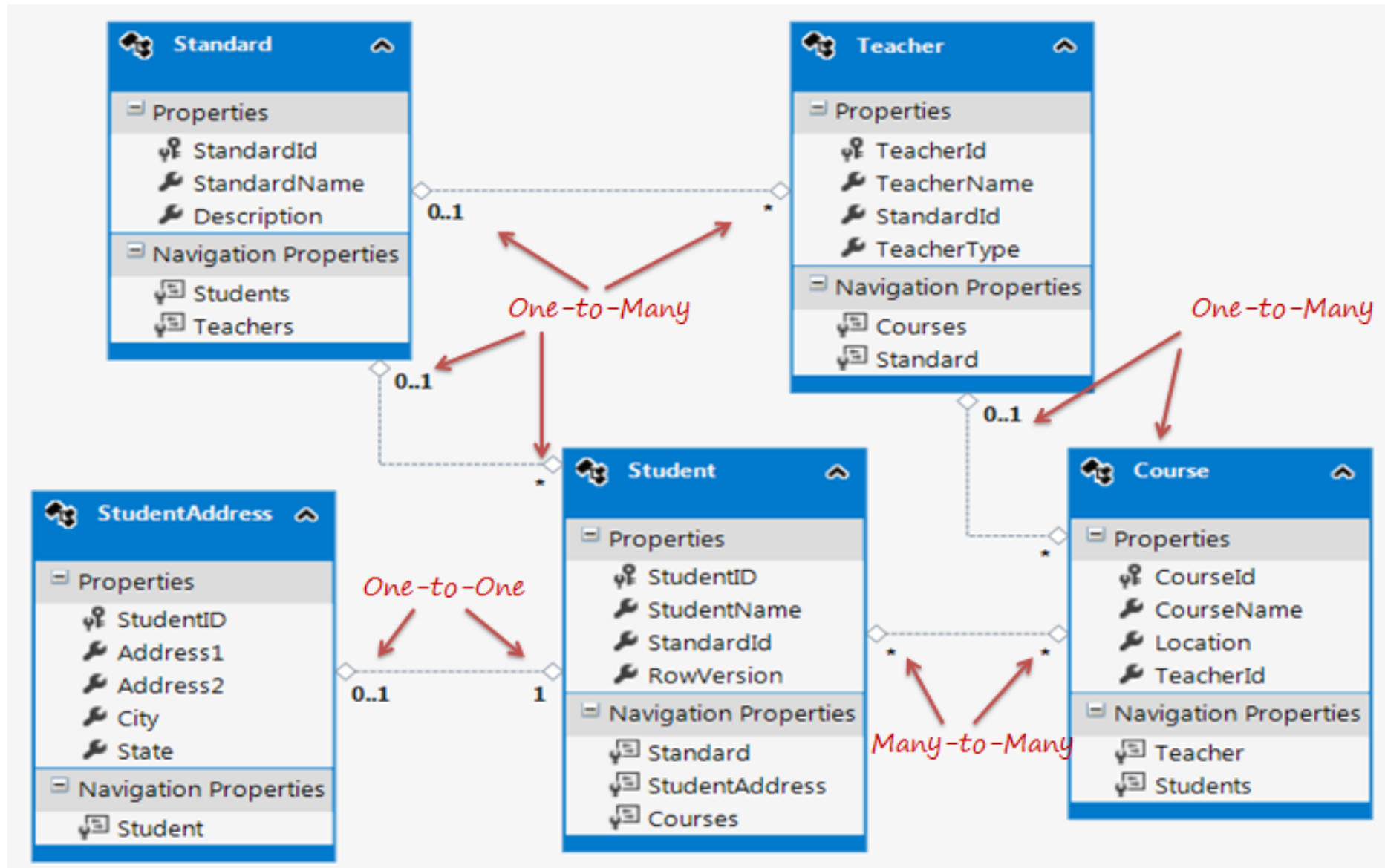
Virtual

```
class MyBaseClass
{
    public virtual string Name { get; set; }
    private int num { get; set; };
}

class MyDerivedClass : MyBaseClass
{
    private string name;
    public override string Name
    {
        get { return name; }
        set { (!string.IsNullOrEmpty(value)) ? name = value : name = "Unknown"; }
    }
}
```



Base de données (exemples)



One to One

```
public class Student
{
    public int StudentID { get; set; }
    public string StudentName { get; set; }
    public DateTime? DateOfBirth { get; set; }
    public decimal Height { get; set; }
    public float Weight { get; set; }
    public byte[] RowVersion { get; set; }

    public virtual StudentAddress Address { get; set; }
}
```

```
public class StudentAddress
{
    public int StudentAddressId { get; set; }
    public string Address1 { get; set; }
    public string City { get; set; }
    public int Zipcode { get; set; }
    public string State { get; set; }

    public virtual Student Student { get; set; }
}
```

One to Many

```
public class Course
{
    public int CourseId { get; set; }
    public string CourseName { get; set; }

    public virtual Teacher Teacher { get; set; }
}
```

```
public class Teacher
{
    public int Id { get; set; }
    public string TeacherName { get; set; }
    public TeachingMode ModeOfTeaching { get; set; }

    public virtual ICollection<Course> Courses { get; set; }
}
```

Many to Many

```
public class Course
{
    public int CourseId { get; set; }
    public string CourseName { get; set; }

    public virtual Teacher Teacher { get; set; }

    public virtual ICollection<Student> Students { get; set; }
}

public class Student
{
    public string StudentName { get; set; }
    public virtual StudentAddress Address { get; set; }

    public virtual ICollection<Course> Courses { get; set; }
}
```

Save

```
using (var ctx = new SchoolContext(true))
{
    var student = new Student()
    {
        StudentName = "Bill",
        Height = 1.7m,
        Weight = 70,
        DateOfBirth = DateTime.Today
    }

    ctx.Students.Add(student);
    ctx.SaveChanges();
}
```