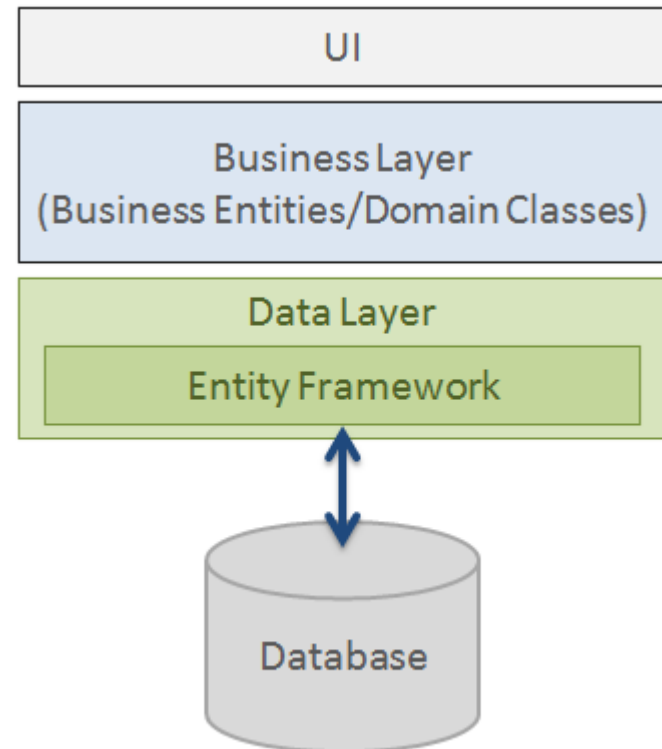


Entity Framework và ADO.NET

Nguyễn Văn Mạnh

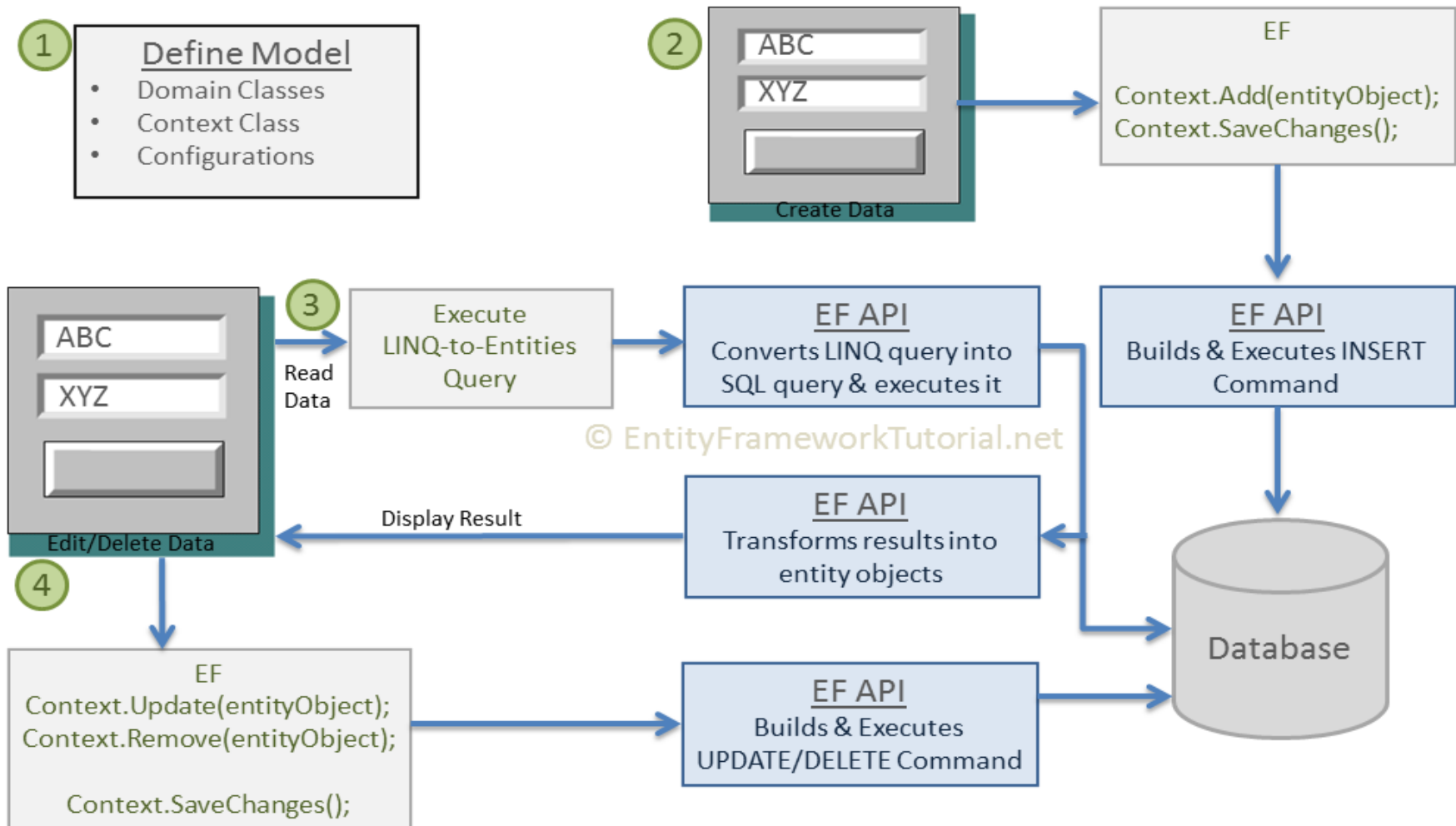
Tổng quan về Entity Framework

- Entity Framework là ORM framework mã nguồn mở bên cạnh đó là Linq to SQL
- EF được cung cấp bởi Microsoft
- EF cho phép ta có thể làm việc với dữ liệu thông qua đối tượng
- Phiên bản mới nhất là EF6 và EF Core

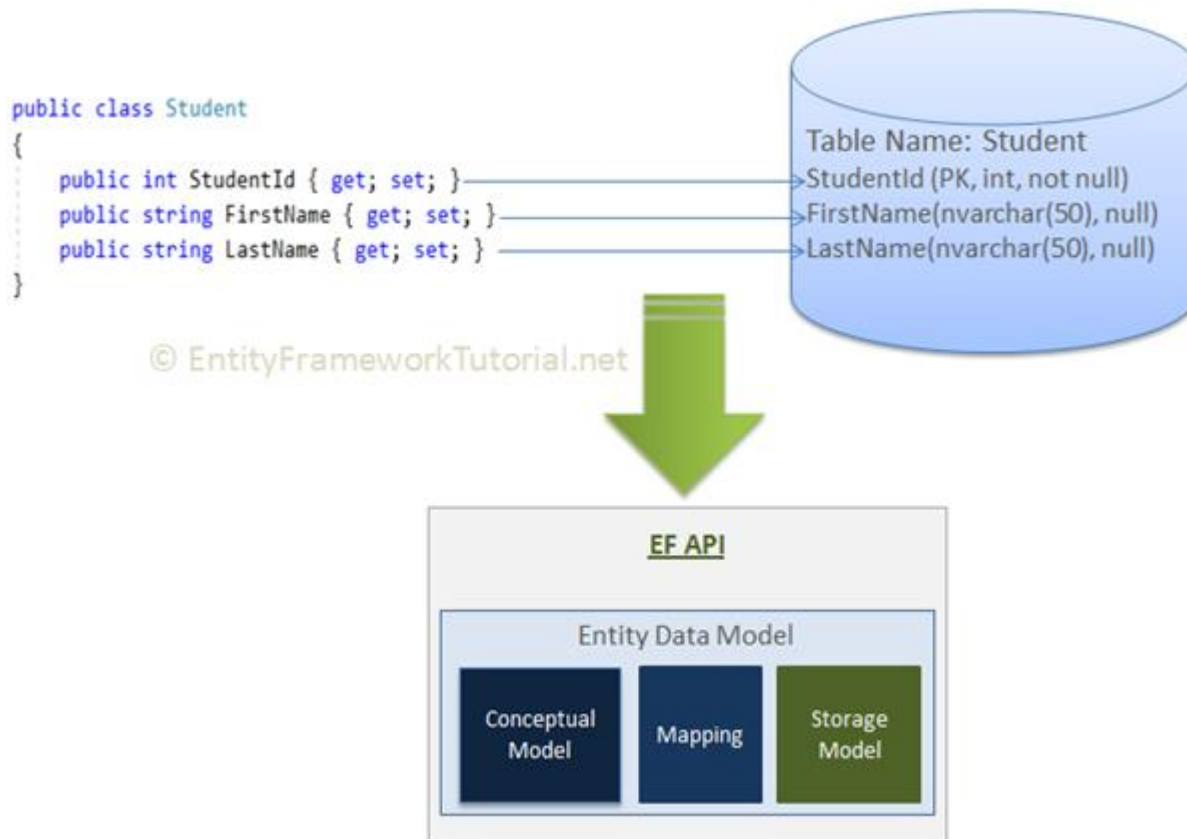


© EntityFrameworkTutorial.net

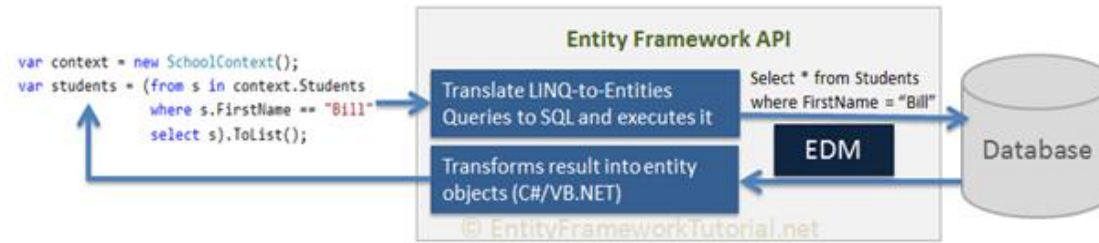
Luồng xử lý của Entity Framework



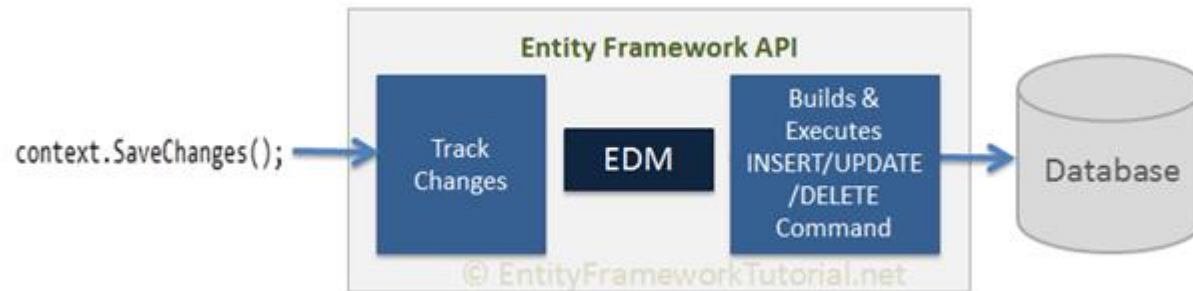
Entity Data Model (EDM)



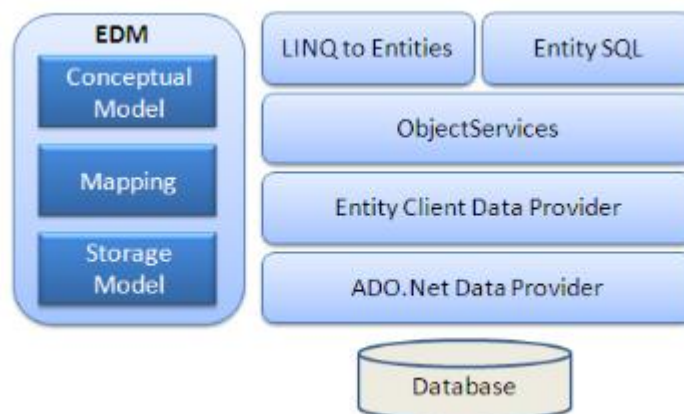
- Querying



- Saving



Kiến trúc tổng quan của Entity Framework



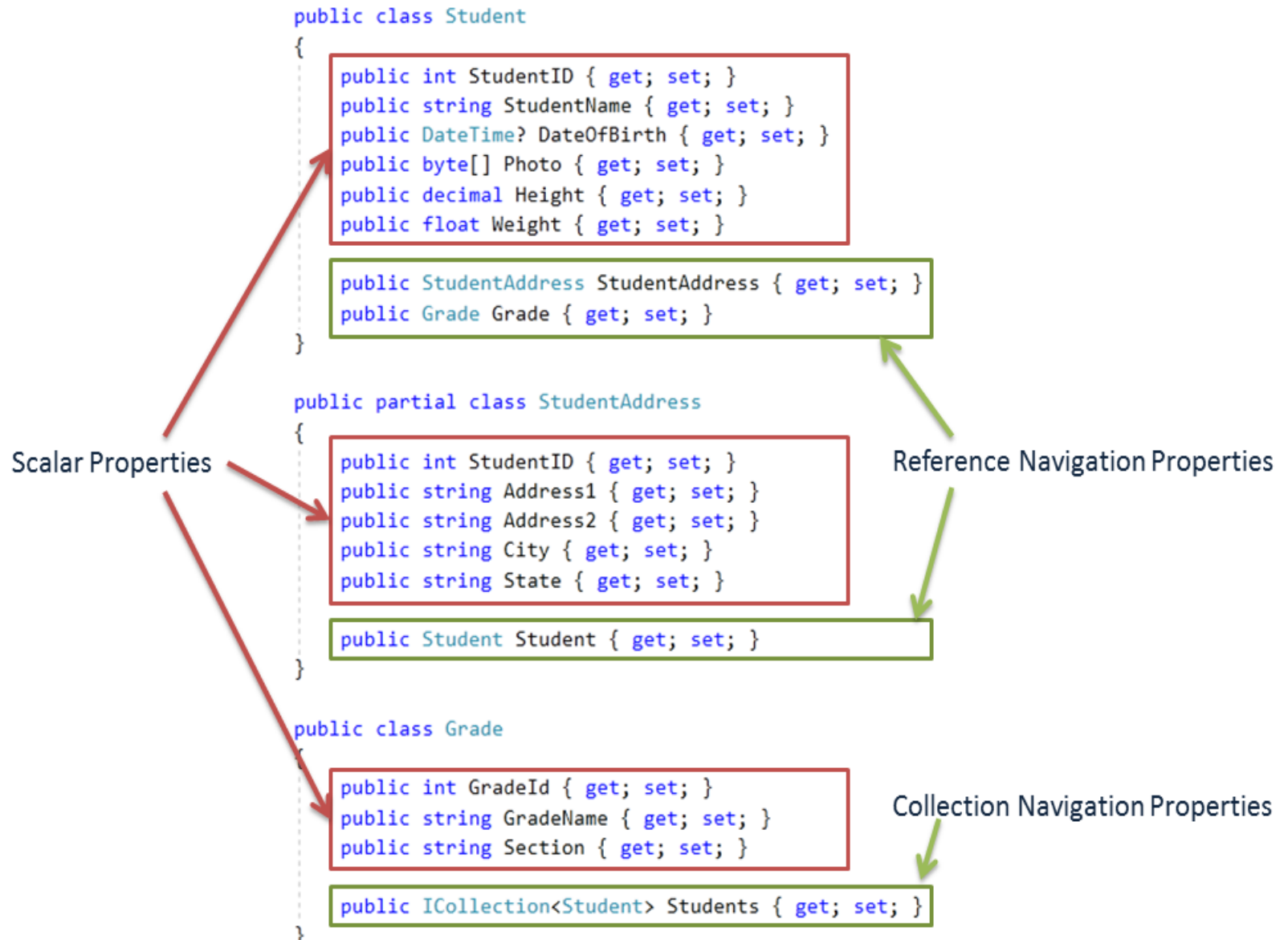
Context Class

- Context class để query hoặc lưu trữ data, thực hiện savechanges()

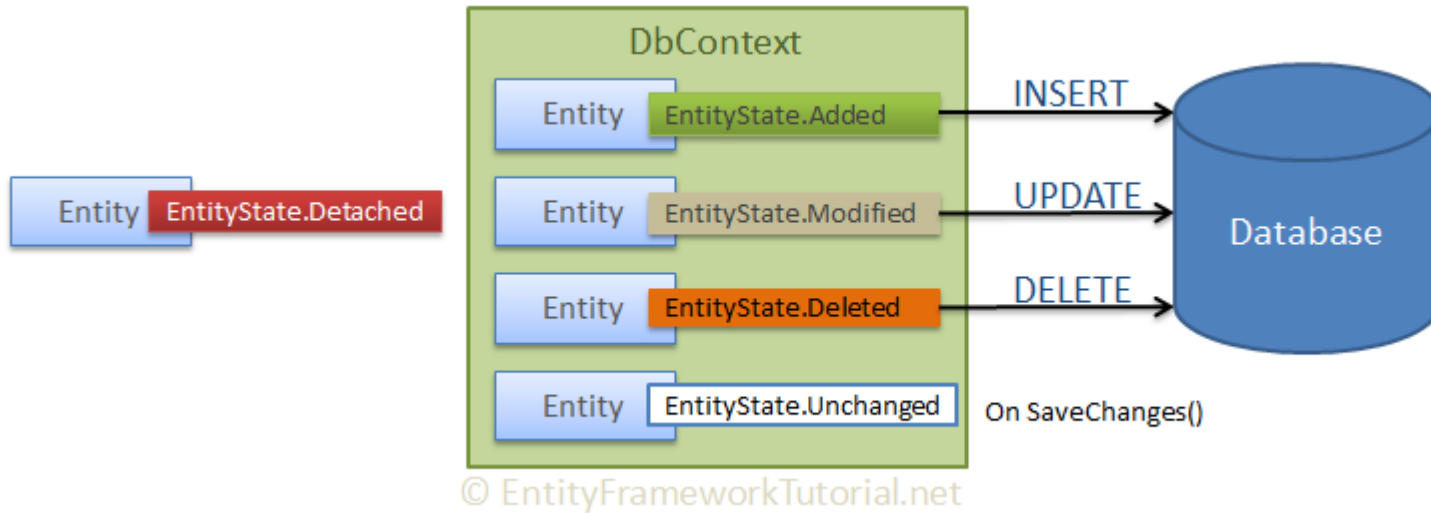
```
public class SchoolContext : DbContext
{
    public SchoolContext()
    {
    }

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

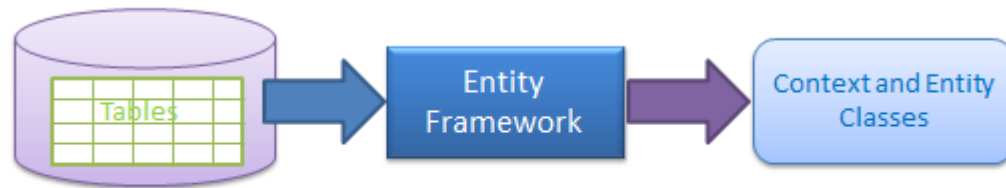
Entity Class



Entity State



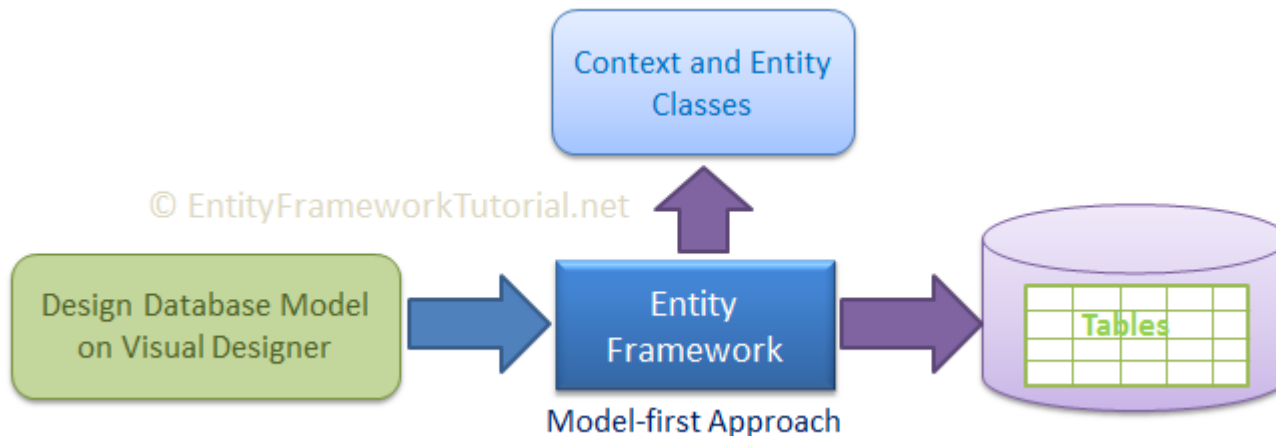
Cách tiếp cận Entity Framework



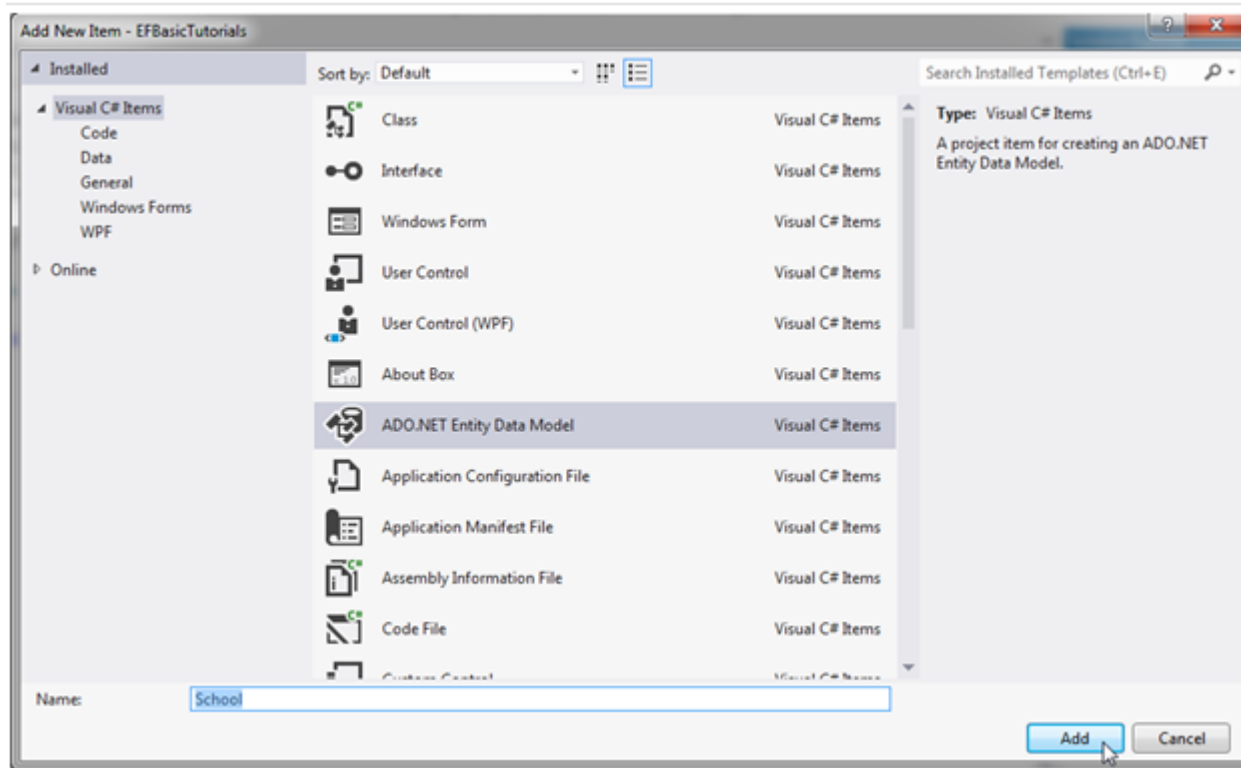
Database-First Approach



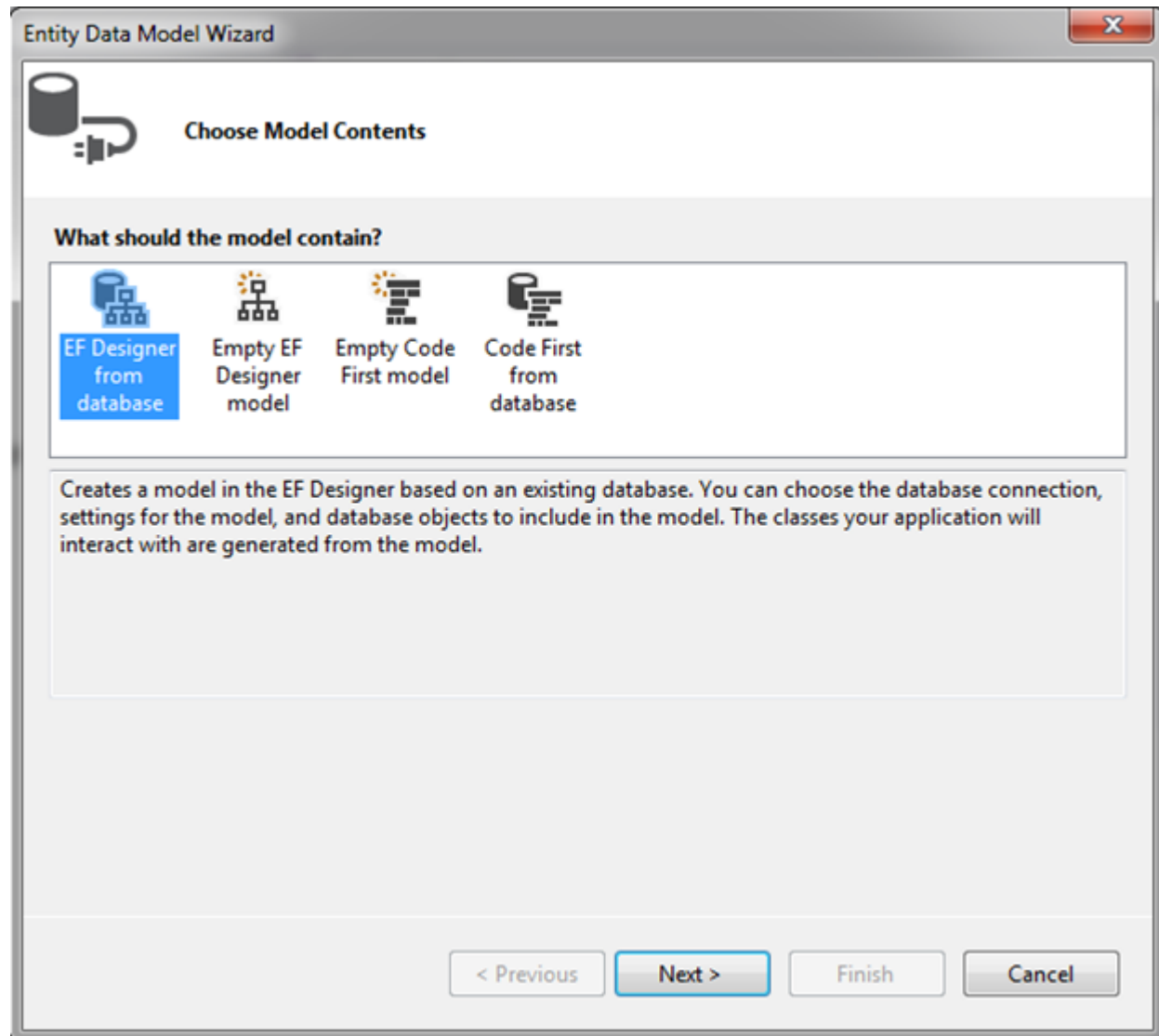
Code-First Approach



Tạo Entity Data Model Database First




Tạo Entity Data Model Database First



Tạo Entity Data Model Database First

Entity Data Model Wizard

 Choose Your Data Connection

Which data connection should your application use to connect to the database?

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☐ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

☒ Save connection settings in App.Config as:

Tạo Entity Data Model Database First

Connection Properties

Enter information to connect to the selected data source or click "Change" to choose a different data source and/or provider.

Data source:
Microsoft SQL Server (SqlClient) Change...

Server name:
. Refresh

Log on to the server

Authentication: Windows Authentication

User name:

Password:

☐ Save my password

Connect to a database

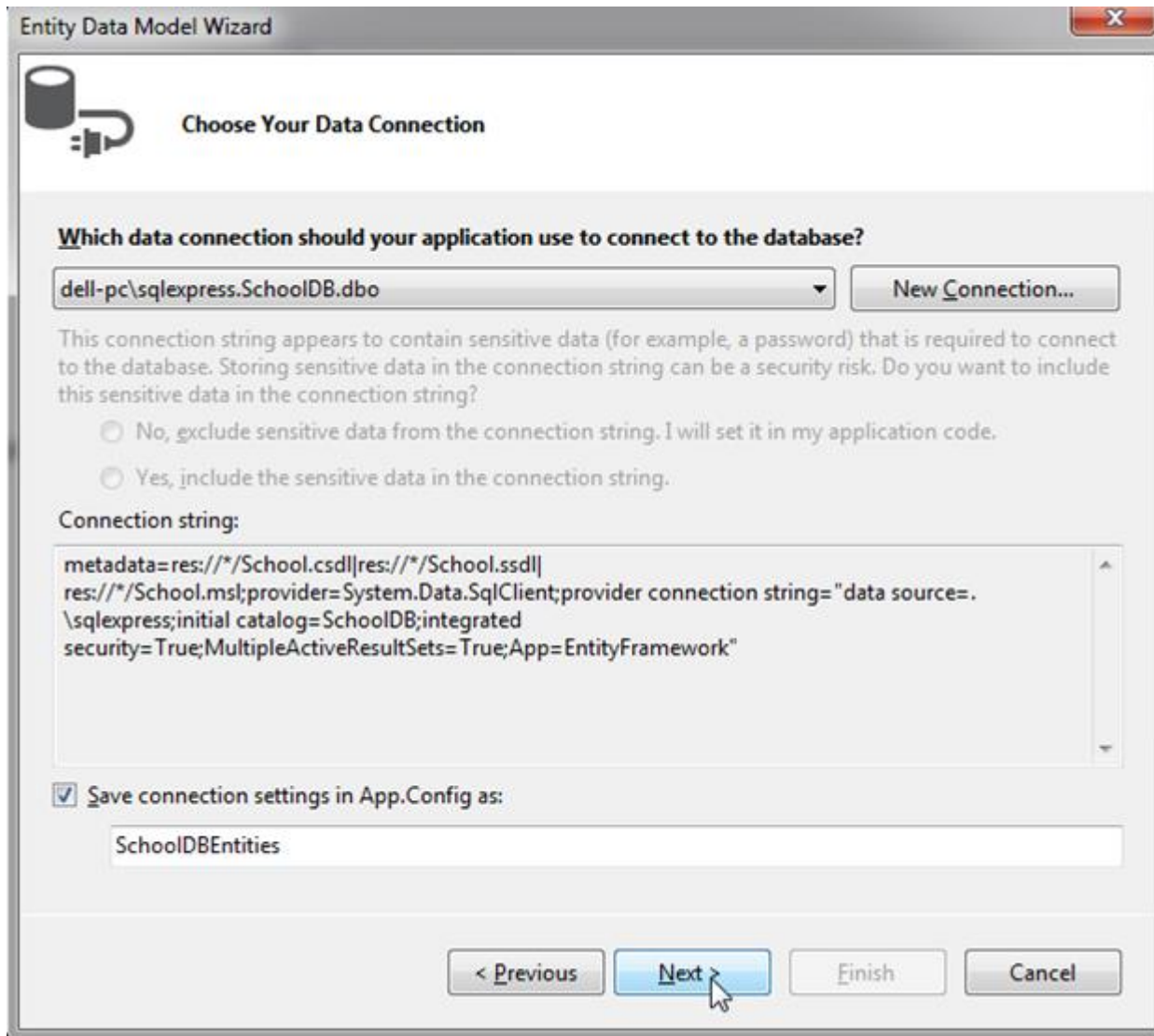
☒ Select or enter a database name:
SchoolDB

☐ Attach a database file:
 Browse...
Logical name:

Advanced...

Test Connection OK Cancel

Tạo Entity Data Model Database First



The image shows a screenshot of the 'Entity Data Model Wizard' window, specifically the 'Choose Your Data Connection' step. The window has a title bar with the text 'Entity Data Model Wizard' and a close button. Below the title bar is a header area with a database icon and the text 'Choose Your Data Connection'. The main content area contains a question: 'Which data connection should your application use to connect to the database?'. Below this question is a dropdown menu showing 'dell-pc\sqlexpress.SchoolDB.dbo' and a 'New Connection...' button. A paragraph of text explains that the connection string might contain sensitive data and asks if the user wants to include it. There are two radio buttons: 'No, exclude sensitive data from the connection string. I will set it in my application code.' (selected) and 'Yes, include the sensitive data in the connection string.' Below this is a 'Connection string:' label and a text box containing a complex connection string. At the bottom, there is a checkbox labeled 'Save connection settings in App.Config as:' which is checked, and a text box containing 'SchoolDBEntities'. At the very bottom are four buttons: '< Previous', 'Next >' (highlighted with a mouse cursor), 'Finish', and 'Cancel'.

Entity Data Model Wizard

Choose Your Data Connection

Which data connection should your application use to connect to the database?

dell-pc\sqlexpress.SchoolDB.dbo

New Connection...

This connection string appears to contain sensitive data (for example, a password) that is required to connect to the database. Storing sensitive data in the connection string can be a security risk. Do you want to include this sensitive data in the connection string?

☒ No, exclude sensitive data from the connection string. I will set it in my application code.

☐ Yes, include the sensitive data in the connection string.

Connection string:

```
metadata=res://*/School.csdl|res://*/School.ssdl|
res://*/School.msl;provider=System.Data.SqlClient;provider connection string="data source=.
\sqlexpress;initial catalog=SchoolDB;integrated
security=True;MultipleActiveResultSets=True;App=EntityFramework"
```

☒ Save connection settings in App.Config as:

SchoolDBEntities

< Previous Next > Finish Cancel

Tạo Entity Data Model Database First

Entity Data Model Wizard

Choose Your Database Objects and Settings

Which database objects do you want to include in your model?

- ☒ Tables
 - ☒ dbo
 - ☒ Course
 - ☒ Standard
 - ☒ Student
 - ☒ StudentAddress
 - ☒ StudentCourse
 - ☒ Teacher
 - ☒ Views
 - ☒ Stored Procedures and Functions

☒ Pluralize or singularize generated object names

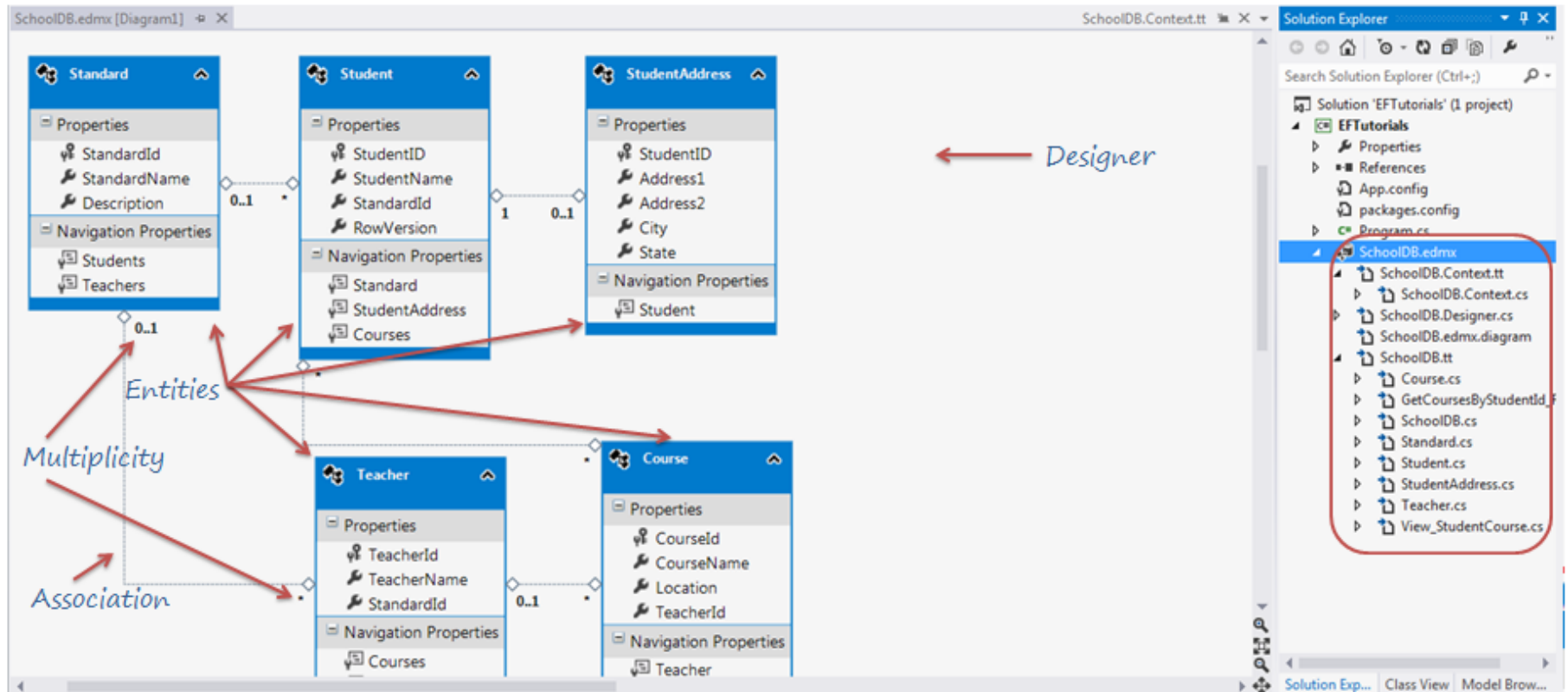
☒ Include foreign key columns in the model

☒ Import selected stored procedures and functions into the entity model

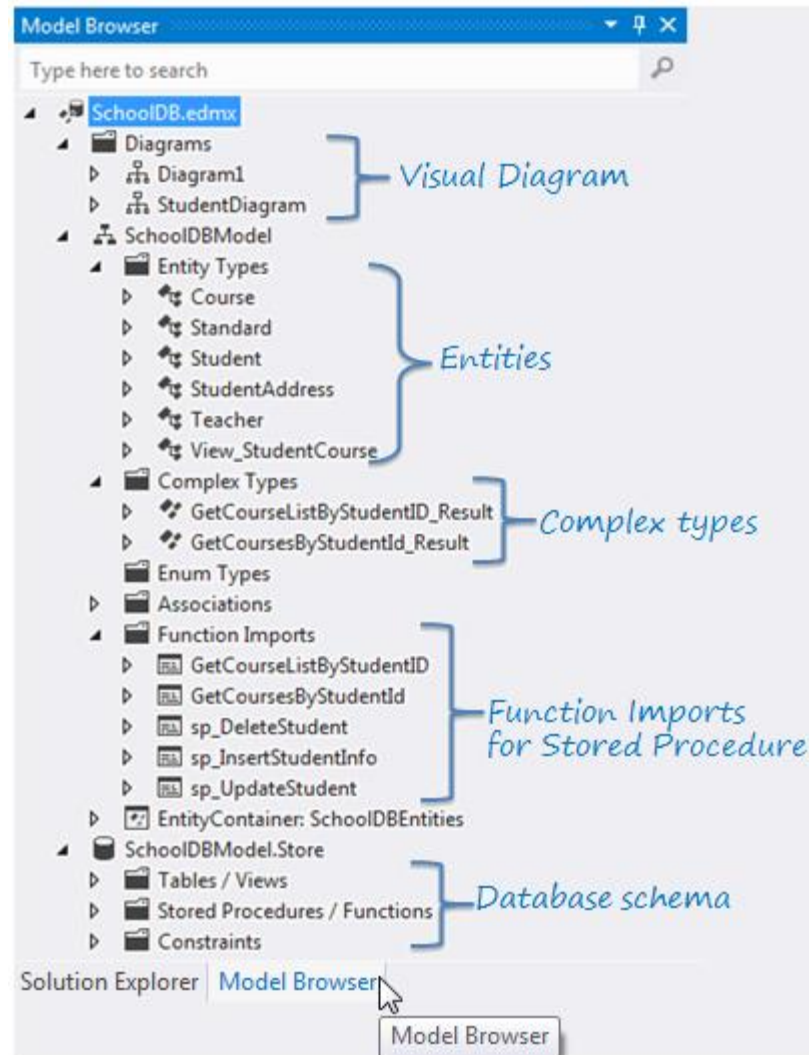
Model Namespace:

SchoolDBModel

< Previous Next > **Finish** Cancel



Model Browser



DbContext

The screenshot displays the Visual Studio IDE with the `SchoolDBEntities` class open in the main editor and the Solution Explorer on the right.

Code in `SchoolDBEntities.cs`:

```
namespace EFTutorials
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;
    using System.Data.Entity.Core.Objects;
    using System.Linq;

    public partial class SchoolDBEntities : DbContext
    {
        public SchoolDBEntities()
            : base("name=SchoolDBEntities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Course> Courses { get; set; }
        public virtual DbSet<Standard> Standards { get; set; }
        public virtual DbSet<Student> Students { get; set; }
        public virtual DbSet<StudentAddress> StudentAddresses { get; set; }
        public virtual DbSet<Teacher> Teachers { get; set; }
    }
}
```

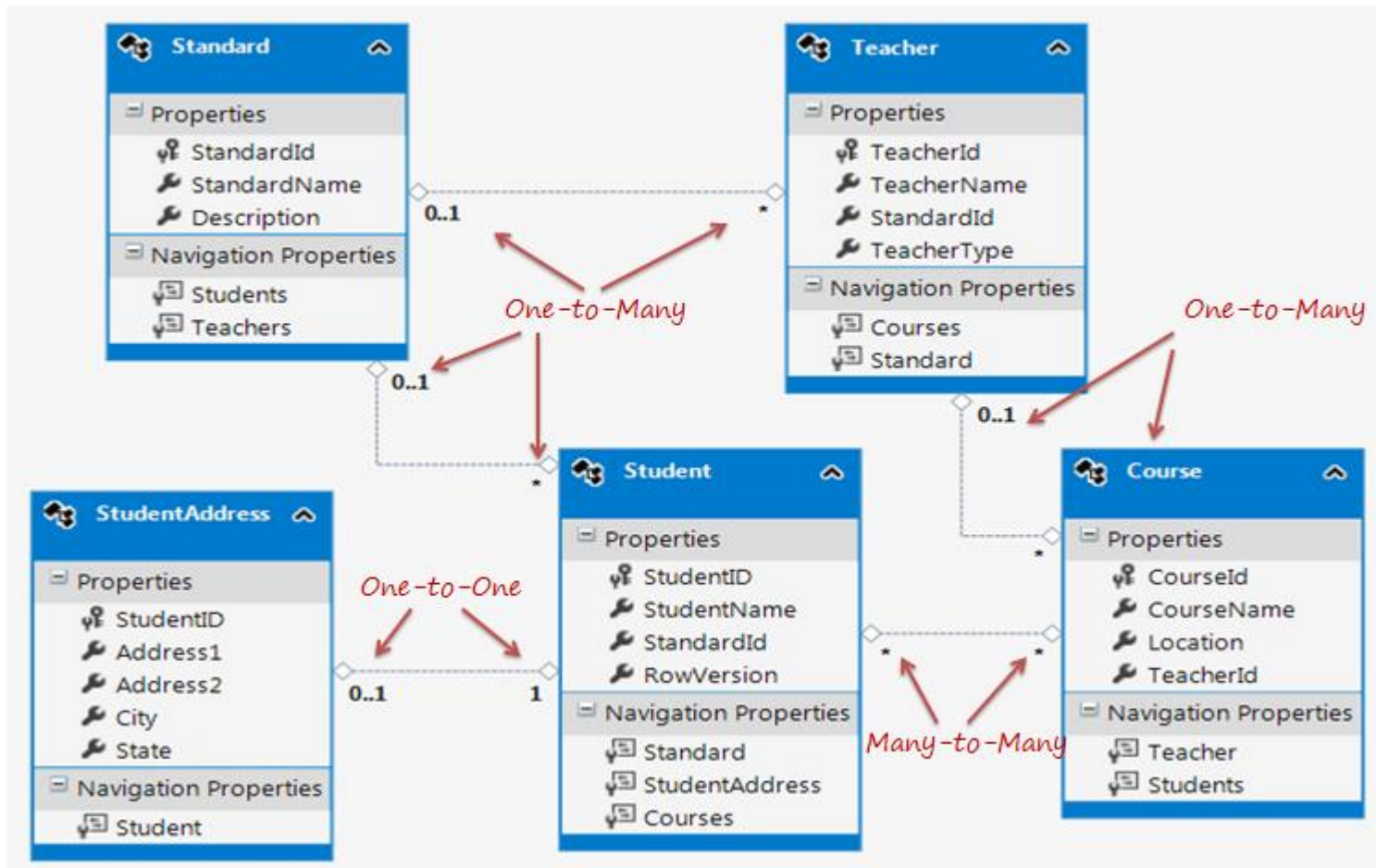
Annotations:

- Fluent API:** An arrow points to the `OnModelCreating` method.
- Entity set:** An arrow points to the `DbSet<Course>` property.

Solution Explorer:

- The project `EFTutorials` is expanded.
- The file `SchoolDB.Context.cs` is selected and highlighted.

Relationship



Query

- Linq to entities
 - Linq Method

```
//Querying with LINQ to Entities
using (var context = new SchoolDBEntities())
{
    var query = context.Students
                        .where(s => s.StudentName == "Bill")
                        .FirstOrDefault<Student>();
}
```

- Linq Syntax

```
using (var context = new SchoolDBEntities())
{
    var query = from st in context.Students
                where st.StudentName == "Bill"
                select st;

    var student = query.FirstOrDefault<Student>();
}
```

Query

- Linq to entities
 - First()
 - FirstOrDefault()
 - Single()
 - SingleOrDefault()
 - ToList()
 - Count()

```
using (var ctx = new SchoolDBEntities())
{
    var student = (from s in ctx.Students
                    where s.StudentName == "Bill"
                    select s).FirstOrDefault<Student>();
}
```

```
var ctx = new SchoolDBEntities();
var student = ctx.Students.Find(1);
```

```
using (var ctx = new SchoolDBEntities())
{
    var students = ctx.Students.OrderBy(s => s.StudentName).ToList();
    // or descending order
    var descStudents = ctx.Students.OrderByDescending(s => s.StudentName).ToList();
}
```

Query

- Entity SQL

```
//Querying with Object Services and Entity SQL
string sqlString = "SELECT VALUE st FROM SchoolDBEntities.Students " +
    "AS st WHERE st.StudentName == 'Bill'";

var objctx = (ctx as IObjectContextAdapter).ObjectContext;

ObjectQuery<Student> student = objctx.CreateQuery<Student>(sqlString);
Student newStudent = student.First<Student>();
```

- Native SQL

```
using (var ctx = new SchoolDBEntities())
{
    var studentName = ctx.Students.SqlQuery("Select studentid, studentname, standardId from Student where studentname='Bill']").FirstOrDefault<Student>();
}
```


Eager Loading

- Là quá trình tải một loại thực thể thì cũng tải cho loại thực thể liên quan như một phần của truy vấn. Được sử dụng bằng hàm Include()

```
using (var ctx = new SchoolDBEntities())
{
    var stud1 = ctx.Students
        .Include("Standard")
        .Where(s => s.StudentName == "Bill")
        .FirstOrDefault<Student>();
}
```



```
SELECT TOP (1)
[Extent1].[StudentID] AS [StudentID],
[Extent1].[StudentName] AS [StudentName],
[Extent2].[StandardId] AS [StandardId],
[Extent2].[StandardName] AS [StandardName],
[Extent2].[Description] AS [Description]
FROM [dbo].[Student] AS [Extent1]
LEFT OUTER JOIN [dbo].[Standard] AS [Extent2] ON [Extent1].[StandardId] = [Extent2].[StandardId]
WHERE 'Bill' = [Extent1].[StudentName]
```


Lazy Loading

- Entity chỉ được load khi có lời gọi.

```
using (var ctx = new SchoolDBEntities())
{
    //Loading students only
    IList<Student> studList = ctx.Students.ToList<Student>();

    Student std = studList[0];

    //Loads Student address for particular Student only (seperate SQL query)
    StudentAddress add = std.StudentAddress;
}
```

- Để sử dụng lazyloading
 - Context.Configuration.ProxyCreationEnable = true
 - Context.Configuration.LazyLoadingEnable = true
 - Navigation property phải để public và virtual

Explicit Loading

- Load tường minh sử dụng hàm Load() để gọi các thực thể quan hệ

```
using (var context = new SchoolContext())
{
    var student = context.Students
        .Where(s => s.FirstName == "Bill")
        .FirstOrDefault<Student>();

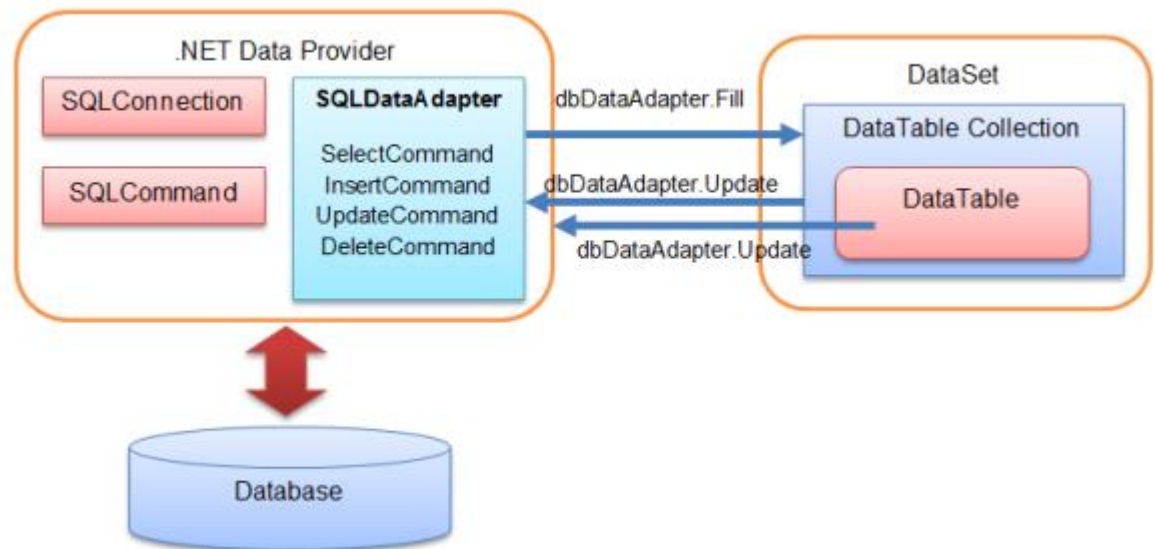
    context.Entry(student).Reference(s => s.StudentAddress).Load(); // loads StudentAddress
    context.Entry(student).Collection(s => s.StudentCourses).Load(); // loads Courses collection
}
```

So sánh performance giữa các query

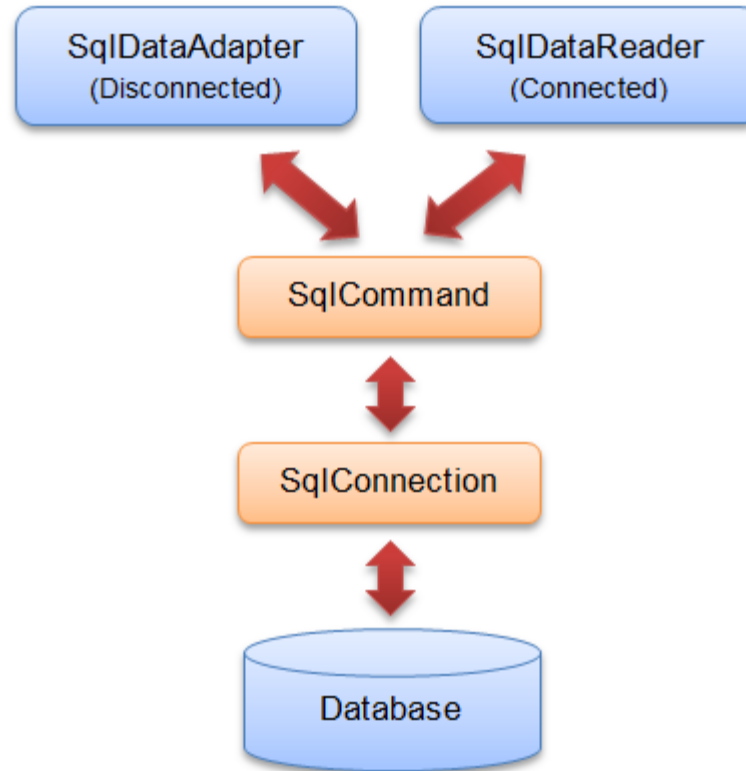
| | Linq to entities | Entity SQL | Raw SQL |
|-------|------------------|------------|---------|
| Lần 1 | 1,402,392,150 ms | 82ms | 7ms |
| Lần 2 | 1,402,392,206 ms | 4ms | 5ms |
| Lần 3 | 1,402,392,197 ms | 4ms | 5ms |
| | | | |

Giới thiệu về ADO.NET

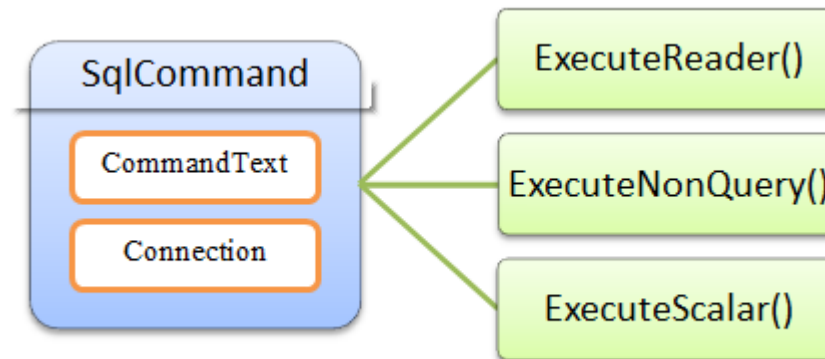
- Thư viện hỗ trợ thao tác với database từ Microsoft
- Các đối tượng
 - SqlConnection
 - SqlCommand
 - SqlDataReader
 - DataSet
 - SqlDataAdapter



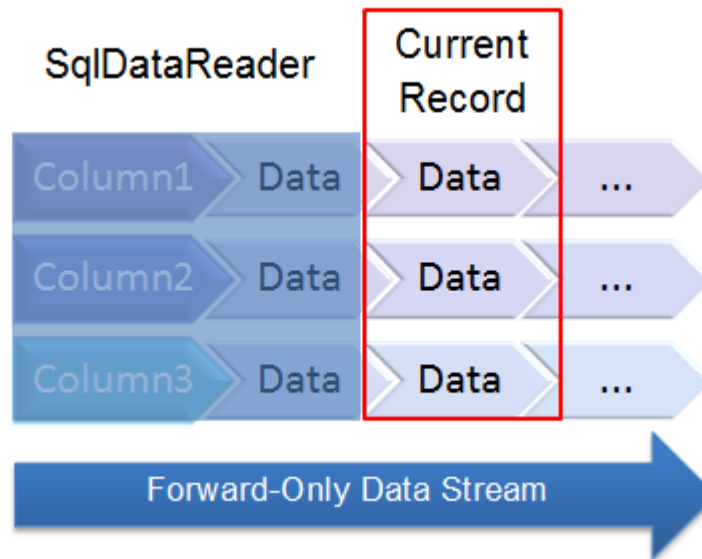
ADO.NET



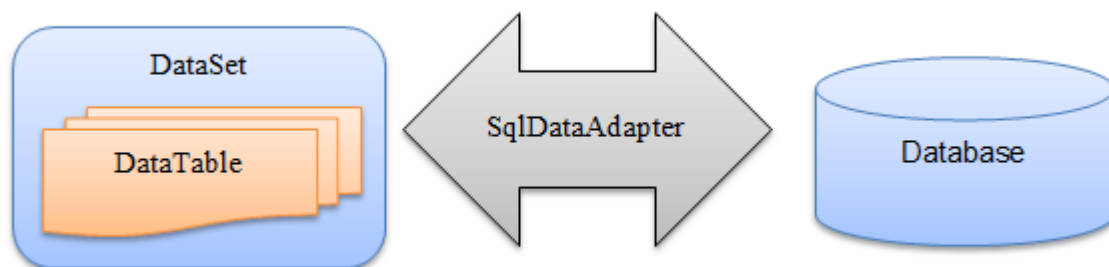
SqlCommand



SqlDataReader



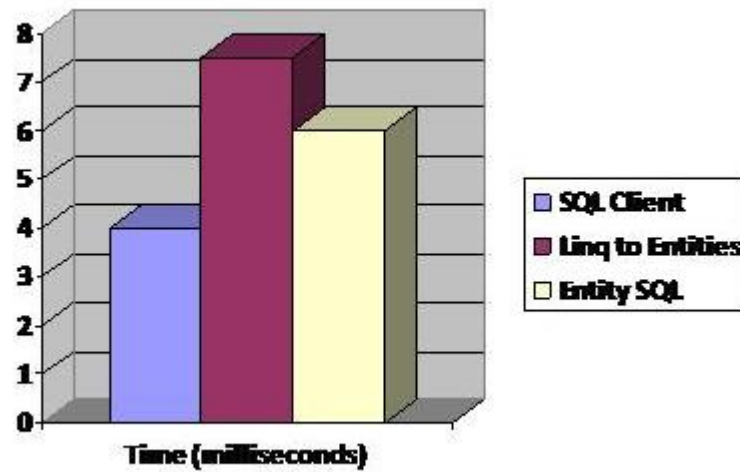
DataSet



So sánh giữa Entity Framework và ADO.NET

| | Entity Framework | ADO.NET |
|------------------|--|---|
| Performance | Chậm hơn ADO.NET | Rất Nhanh |
| Số lượng code | Code ít hơn, đã hỗ trợ code ở tầng Data Layer => tiết kiệm thời gian | Code nhiều hơn, cấu hình phức tạp hơn => mất nhiều thời gian code |
| Khả năng mở rộng | phụ thuộc vào framework và khó maintain | Có thể mở rộng dễ maintain hơn |
| | | |

Performance



THANK YOU