

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

مثال ساده Code-First

مدرس: مهندس افشین رفوآ

## مثال ساده Code-First

خوب اولین مثال ساده Code-First را بسازیم.

در ویزوال استودیو 2010، Class Library project را ایجاد کنید و دو کلاس ساده Student و Standard همان طور که در زیر نشان داده شده است را بنویسید. (برای این مثال شما می توانید Entity Framework 4.1/4.3/5.0 را ببینید).

Student Class:

```
public class Student
{
    public Student()
    {
    }
    public int StudentID { get; set; }
    public string StudentName { get; set; }
}
```

Standard Class:

```
public class Standard
{
    public Standard()
    {
    }
    public int StandardId { get; set; }
    public string StandardName { get; set; }
    public string Description { get; set; }
}
```

اکنون، یک context class ایجاد کنید که از کلاس DbContext با دو ویژگی DbSet یکی برای Student و دیگری برای Standard مشتق گرفته است.

```
namespace SchoolDataLayer
{
    public class Context: DbContext
    {
        public Context(): base()
        {
        }
    }
}
```

```

        }
        public DbSet <Student> Students {
get; set; }
        public DbSet <Standard> Standards {
get; set; }
    }
}

```

و اکنون هم **Console Project** ایجاد کنید تا این کلاس ها را همانطور که در زیر هم نشان داده شده است تست کنید.

```

using (var ctx = new Context())
{
    Student stud = new Student() { StudentName =
    "New Student" };
    ctx.Students.Add(stud);
    ctx.SaveChanges();
}

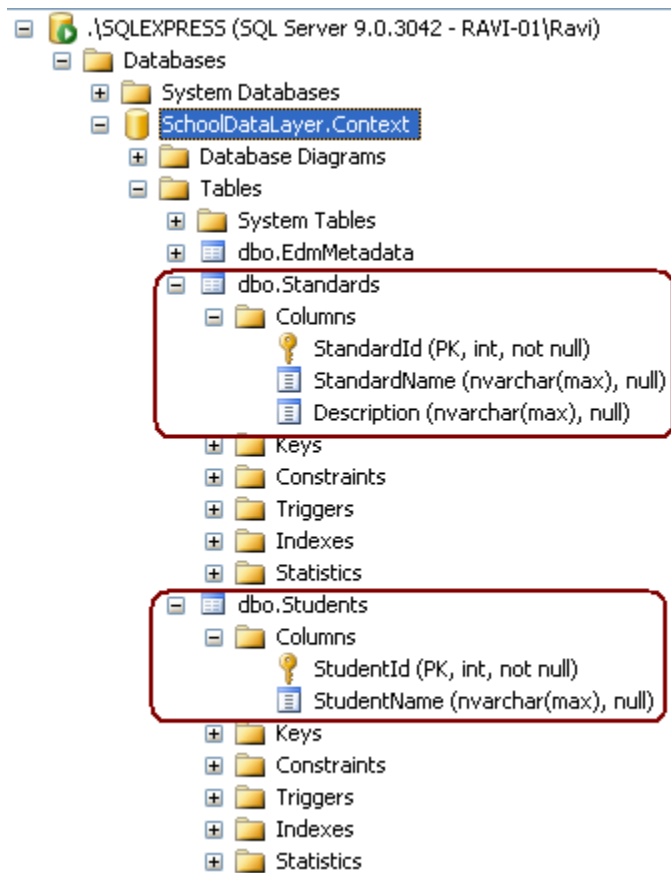
```

اکنون اگر شما **Application** را اجرا کنید، از اینکه **Application** با موفقیت اجرا می شود، شگفت زده خواهید شد.

**Application** با موفقیت اطلاعات **Student** را در پایگاه داده ذخیره کرده است اما پایگاه داده کجاست و جدول ها و ستون های آن چه چیزی هستند؟

این زیبایی **API** های **Code-First**، **Entity Framework** است که پایگاه داده را بر اساس پارمترهایی که به سازنده اصلی کلاس **context** داده شده است، ایجاد می کند. خوب ما هیچ پارامتری را به آن ارسال نکردیم، بنابراین پایگاه داده **"SchoolDataLayer.Context"** را در **SQLEXPRESS** محلی ایجاد می کند. جریان کار پایگاه داده را در قسمت بعدی مشاهده خواهیم کرد.

**Code-first API** دو جدول را نیز در پایگاه داده ایجاد می کند، جدول های **Standards** و **Students** بر اساس کلاس های **Student** و **Standard**. **Code-first API** یک کلید اولیه (**Primary Key**) در جدول ایجاد می کند اگر کلاس **Id** یا ویژگی **Id** را داشته باشد. برای مثال کلاس **Student** ویژگی **StudentId** را دارد، بنابراین **StudentId** را به عنوان کلید اولیه (**PK**) ایجاد می کند. همین طور ستون ها را با همین اسم و نوع داده ایجاد می کند، که اسم **Property** ها و نوع داده در زیر نشان داده شده است.



خوب در این روش بدون ایجاد پایگاه داده، شما می توانید **Application** هایی را بنویسید که در نهایت پایگاه داده را از کلاس های **Domain** تان ایجاد می کند.