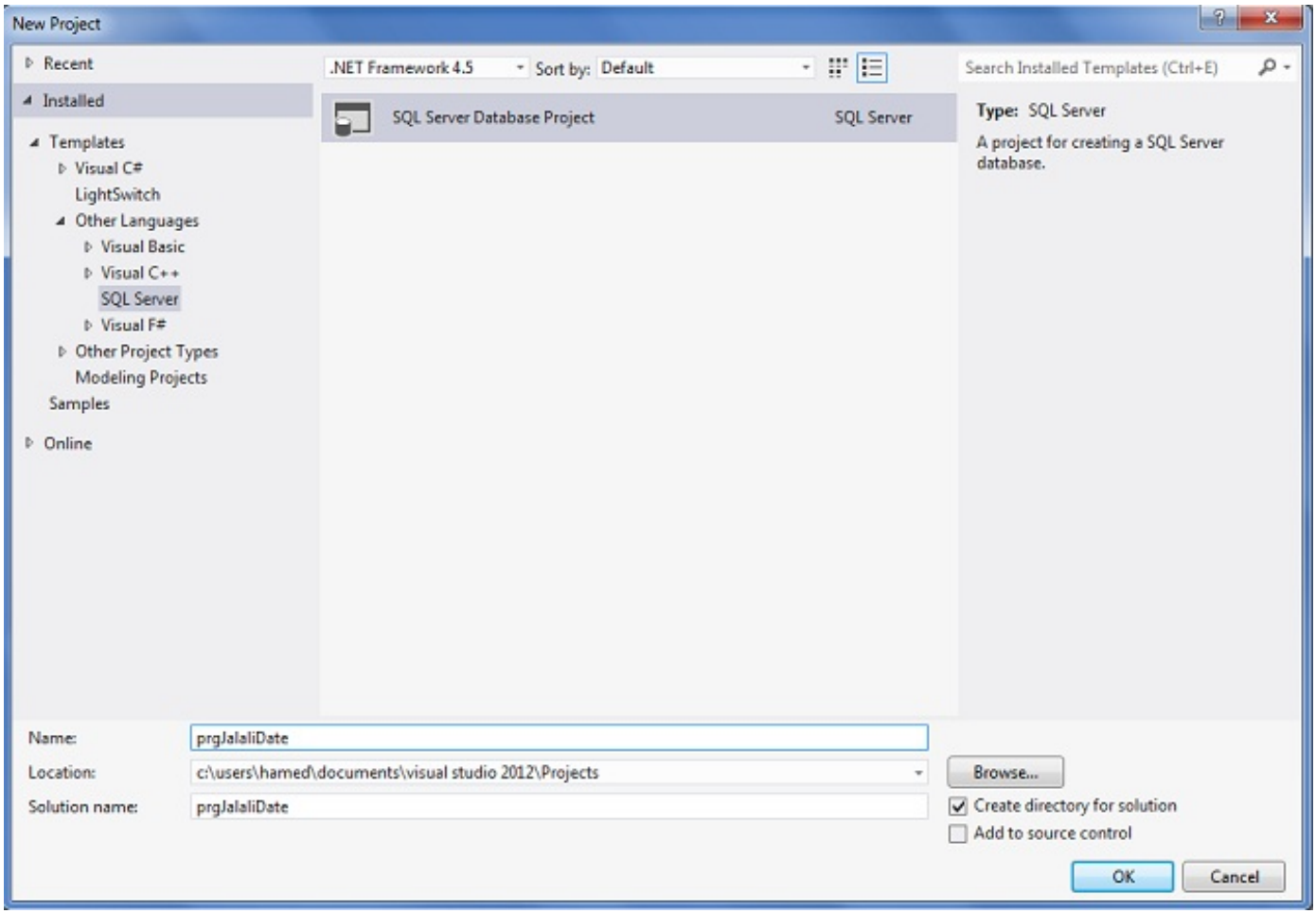
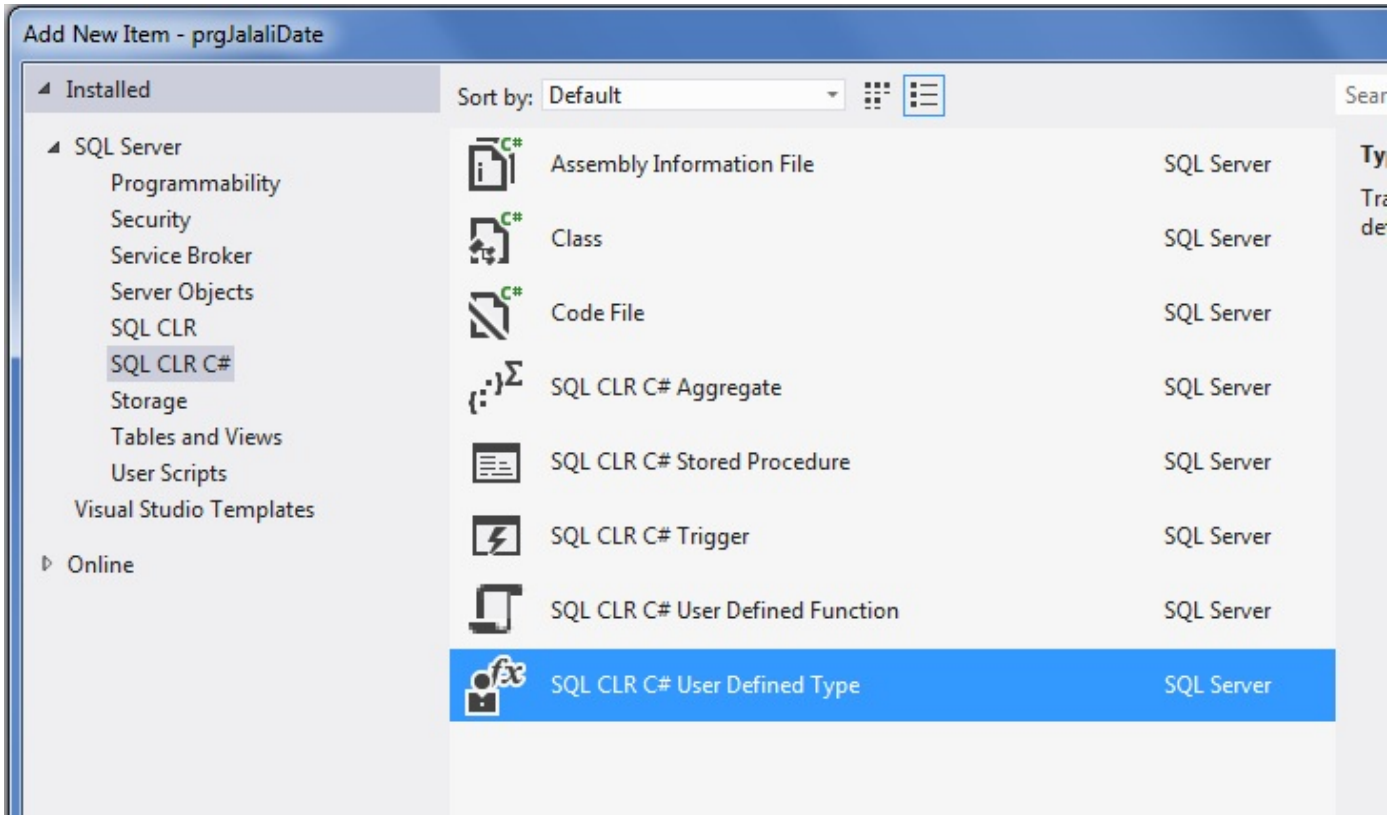
عنوان: **افزودن یک DataType جدید برای نگه‌داری تاریخ خورشیدی - 1**
نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۰۹ ۲۳:۳۰
آدرس: www.dotnettips.info
گروه‌ها: SQL Server, SQL Server 2012, C#.NET, Persian, CLR

ثبت و نگه‌داری تاریخ خورشیدی در SQL Server از دیرباز یکی از نگرانی‌های برنامه‌نویسان و طراحان پایگاه داده‌ها بوده است. در این نوشتار، راه‌کار تعریف یک DataType در SQL Server 2012 به روش CLR آموزش داده خواهد شد.

در ویژوال استودیو یک پروژه‌ی جدید از نوع SQL Server Database Project به شکل زیر ایجاد کنید:



نام پروژه را به یاد تقویم خیام، prgJalaliDate می‌گذارم. در Solution Explorer روی نام پروژه راست‌کلیک کرده، سپس روی Add New Item کلیک کنید. در پنجره‌ی بازشده مطابق شکل SQL CLR C# User Defined Type را برگزینید؛ سپس نام JalaliDateType را برای آن انتخاب کنید.



متن موجود در صفحه‌ی بازشده را کاملاً حذف کرده و با کد زیر جای‌گزین کنید.
(در کد زیر همه‌ی توابع لازم برای مقداردهی به سال، ماه، روز، ساعت، دقیقه و ثانیه و البته گرفتن مقدار از آن‌ها، تبدیل تاریخ خورشیدی به میلادی، گرفتن تاریخ به تنهایی، گرفتن زمان به تنهایی، افزایش یا کاهش زمان برپایه‌ی یکی از متغیرهای زمان و بررسی و اعتبارسنجی انواع بخش‌های زمان گنجانده شده است. در صورت پرسش یا پیشنهاد روی هر کدام در قسمت نظرات، پیام خود را بنویسید.)

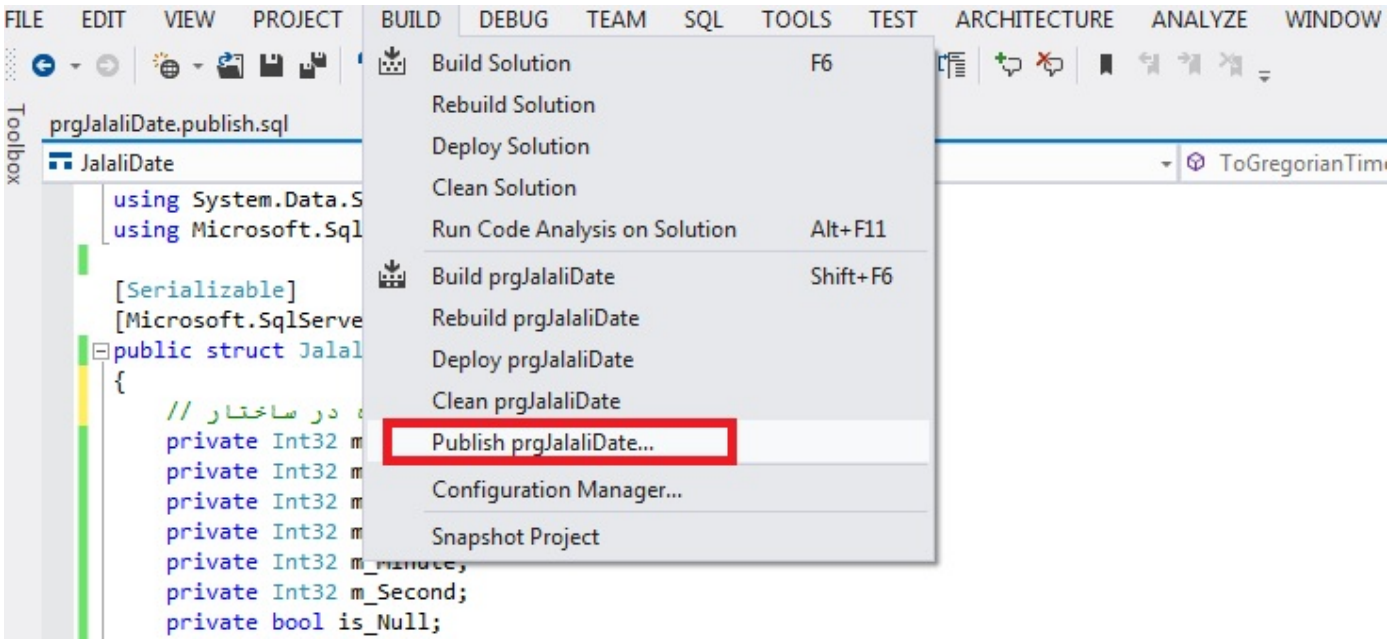
```
using System;
using System.Data.SqlTypes;
using Microsoft.SqlServer.Server;

[Serializable()]
[SqlUserDefinedType(Format.Native)]
public struct JalaliDate : INullable
{
    private Int16 m_Year;
    private byte m_Month;
    private byte m_Day;
    private byte m_Hour;
    private byte m_Minute;
    private byte m_Second;
    private bool is_Null;


    public Int16 Year
    {
        get
        {
            return (this.m_Year);
        }
        set
        {
            m_Year = value;
        }
    }

    public byte Month
    {
        get
        {
            return (this.m_Month);
```

```
        }
        set
        {
            m_Month = value;
        }
    }

    public byte Day
    {
        get
        {
            return (this.m_Day);
        }
        set
        {
            m_Day = value;
        }
    }

    public byte Hour
    {
        get
        {
            return (this.m_Hour);
        }
        set
        {
            m_Hour = value;
        }
    }

    public byte Minute
    {
        get
        {
            return (this.m_Minute);
        }
        set
        {
            m_Minute = value;
        }
    }

    public byte Second
    {
        get
        {
            return (this.m_Second);
        }
        set
        {
            m_Second = value;
        }
    }

    public bool IsNull
    {
        get
        {
            return is_Null;
        }
    }

    public static JalaliDate Null
    {
        get
        {
            JalaliDate jl = new JalaliDate();
            jl.is_Null = true;
            return (jl);
        }
    }


    public override string ToString()
    {
        if (this.IsNull)
        {
            return "NULL";
        }
        else
```

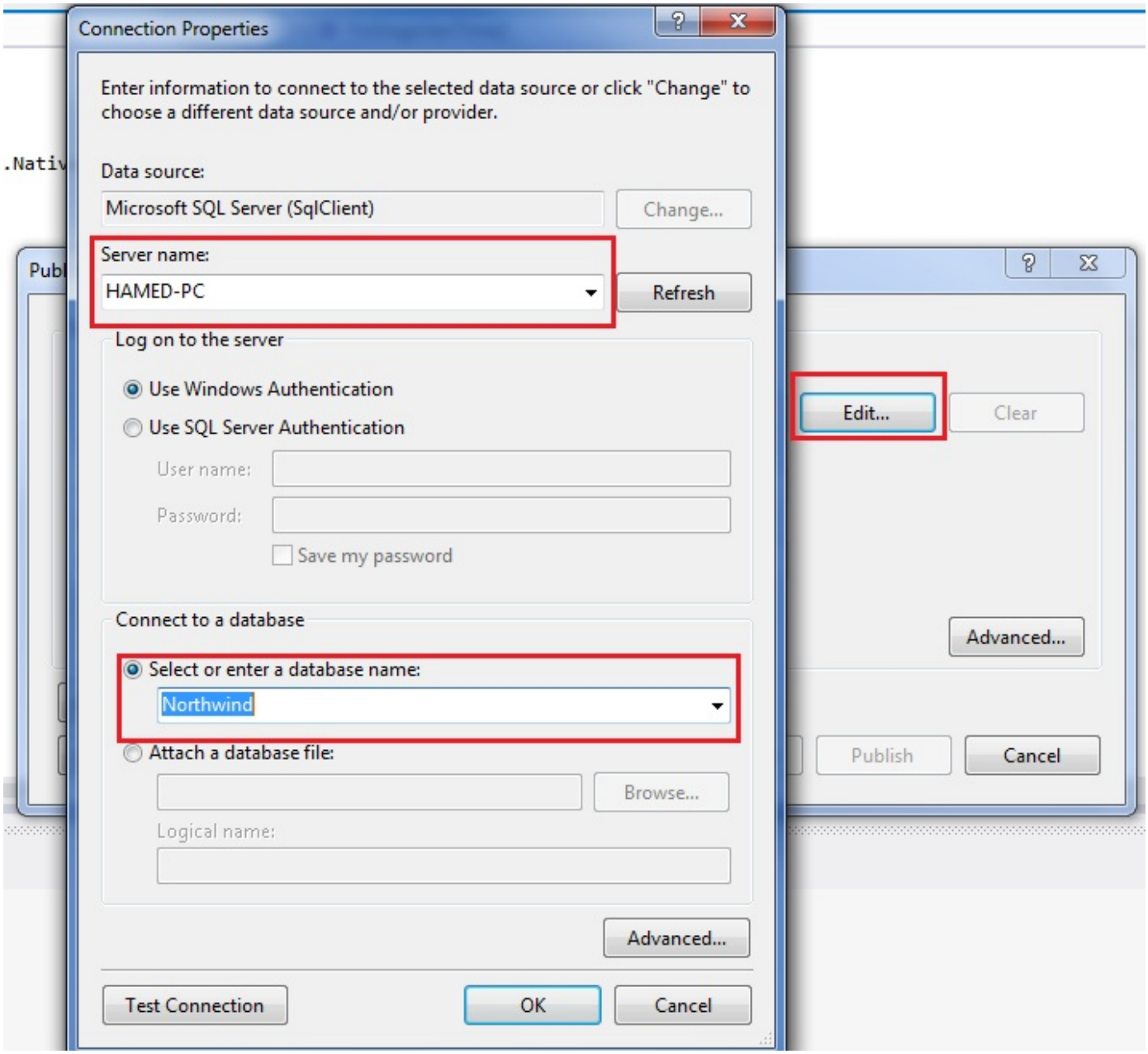
```
        {
            return this.m_Year.ToString("D4") + "/" + this.m_Month.ToString("D2") + "/" +
this.m_Day.ToString("D2") + " " + this.Hour.ToString("D2") + ":" + this.Minute.ToString("D2") + ":" +
this.Second.ToString("D2");
        }
    }


    public static JalaliDate Parse(SqlString s)
    {
        if (s.IsNull)
        {
            return Null;
        }

        System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
        string str = Convert.ToString(s);
        string[] JDate = str.Split(' ')[0].Split('/');

        JalaliDate jl = new JalaliDate();

        jl.Year = Convert.ToInt16(JDate[0]);
        byte MonthsInYear = (byte)pers.GetMonthsInYear(jl.Year);
        jl.Month = (byte.Parse(JDate[1]) <= MonthsInYear ? (byte.Parse(JDate[1]) > 0 ?
byte.Parse(JDate[1]) : (byte)1) : MonthsInYear);
        byte DaysInMonth = (byte)pers.GetDaysInMonth(jl.Year, jl.Month); ;
        jl.Day = (byte.Parse(JDate[2]) <= DaysInMonth ? (byte.Parse(JDate[2]) > 0 ?
byte.Parse(JDate[2]) : (byte)1) : DaysInMonth);
        if (str.Split(' ').Length > 1)
        {
            string[] JTime = str.Split(' ')[1].Split(':');
            jl.Hour = (JTime.Length >= 1 ? (byte.Parse(JTime[0]) < 23 && byte.Parse(JTime[0]) >=
(byte)0 ? byte.Parse(JTime[0]) : (byte)0) : (byte)0);
            jl.Minute = (JTime.Length >= 2 ? (byte.Parse(JTime[1]) < 59 && byte.Parse(JTime[1]) >=
(byte)0 ? byte.Parse(JTime[1]) : (byte)0) : (byte)0);
            jl.Second = (JTime.Length >= 3 ? (byte.Parse(JTime[2]) < 59 && byte.Parse(JTime[2]) >=
(byte)0 ? byte.Parse(JTime[2]) : (byte)0) : (byte)0);
        }
        else { jl.Hour = 0; jl.Minute = 0; jl.Second = 0; }

        return (jl);
    }

    public SqlString GetDate()
    {
        return this.m_Year.ToString("D4") + "/" + this.m_Month.ToString("D2") + "/" +
this.m_Day.ToString("D2");
    }

    public SqlString GetTime()
    {
        return this.Hour.ToString("D2") + ":" + this.Minute.ToString("D2") + ":" +
this.Second.ToString("D2");
    }

    public SqlDateTime ToGregorianTime()
    {
        System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
        return SqlDateTime.Parse(pers.ToDateTime(this.Year, this.Month, this.Day, this.Hour,
this.Minute, this.Second, 0).ToString());
    }

    public SqlString JalaliDateAdd(SqlString interval, int increment)
    {
        System.Globalization.PersianCalendar pers = new System.Globalization.PersianCalendar();
        DateTime dt = pers.ToDateTime(this.Year, this.Month, this.Day, this.Hour, this.Minute,
this.Second, 0);
        string CInterval = interval.ToString();
        bool isConvert = true;
        switch (CInterval)
        {
            case "Year":
                dt = pers.AddYears(dt, increment);
                break;
            case "Month":
                dt = pers.AddMonths(dt, increment);
                break;
            case "Day":
                dt = pers.AddDays(dt, increment);
                break;
            case "Hour":
```

```
                dt = pers.AddHours(dt, increment);
                break;
            case "Minute":
                dt = pers.AddMinutes(dt, increment);
                break;
            case "Second":
                dt = pers.AddSeconds(dt, increment);
                break;
            default:
                isConvert = false;
                break;
        }

        if (isConvert == true)
        {
            this.Year = (Int16)pers.GetYear(dt);
            this.Month = (byte)pers.GetMonth(dt);
            this.Day = (byte)pers.GetDayOfMonth(dt);
            this.Hour = (byte)pers.GetHour(dt);
            this.Minute = (byte)pers.GetMinute(dt);
            this.Second = (byte)pers.GetSecond(dt);
        }


        return this.m_Year.ToString("D4") + "/" + this.m_Month.ToString("D2") + "/" +
this.m_Day.ToString("D2") + " " + this.Hour.ToString("D2") + ":" + this.Minute.ToString("D2") + ":" +
this.Second.ToString("D2");
    }
}
```

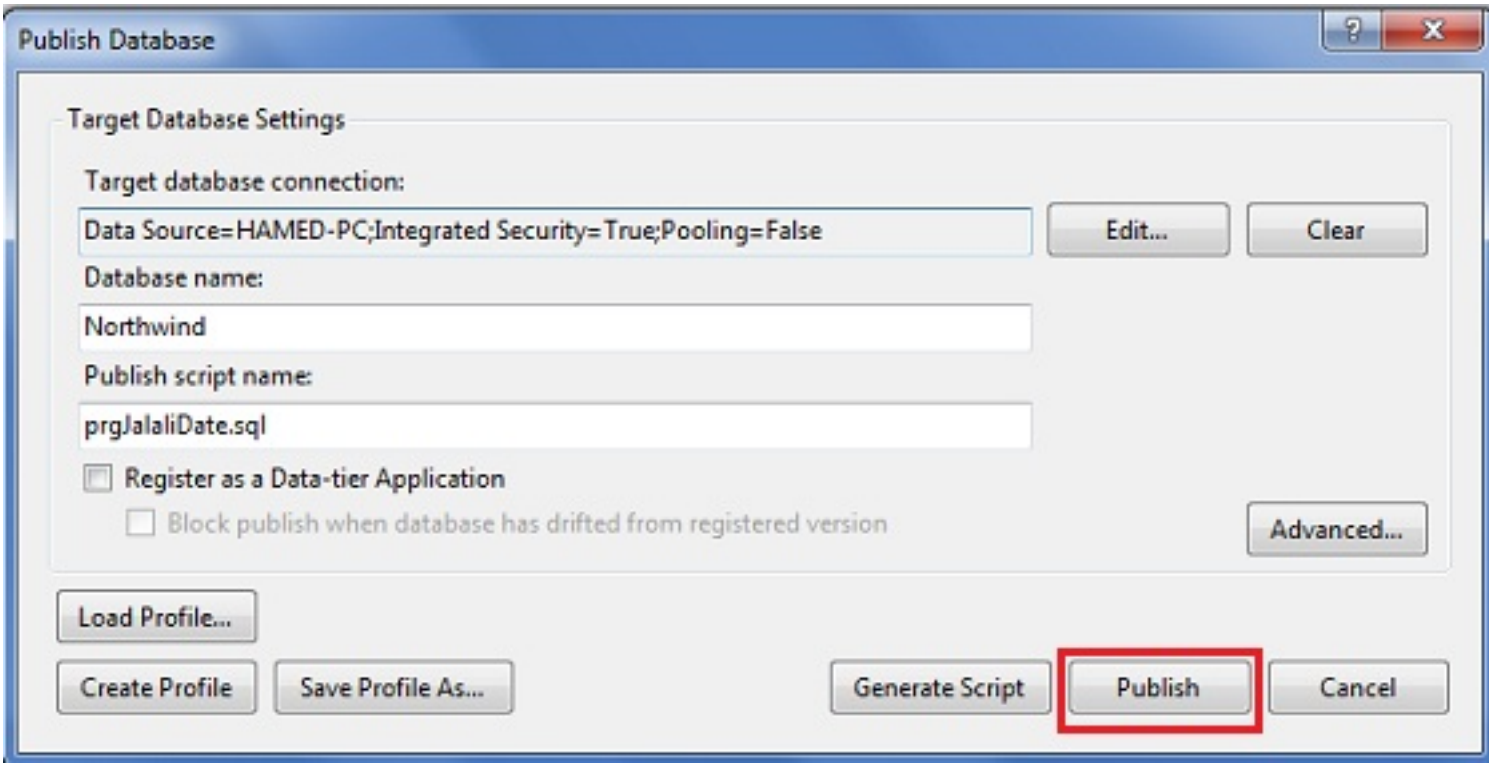
از منوهای بالا روی منوی Bulild و سپس گزینه‌ی Publish prgJalaliDate کلیک کنید:



در پنجره‌ی بازشده روی دکمه‌ی Edit کلیک کنید سپس تنظیمات مربوط به اتصال به پایگاه داده را انجام دهید.



روی دکمه‌ی OK کلیک کنید و سپس در پنجره‌ی اولیه، روی دکمه‌ی Publish کلیک کنید:



به همین سادگی، DataType مربوطه در SQL Server 2012 ساخته می‌شود. خبر خوش این‌که شما می‌توانید با راست‌کلیک روی نام پروژه و انتخاب گزینه‌ی Properties در قسمت Project Setting تنظیمات مربوط به نگارش SQL Server را انجام دهید. (از نگارش 2005 به بعد در VS 2012 پشتیبانی می‌شود.)

اکنون زمان آن رسیده است که DataType ایجادشده را در SQL Server 2012 بیازماییم. SQL Server را باز کنید و دستور زیر را در آن اجرا کتید.

```
USE Northwind

GO

CREATE TABLE dbo.TestTable
(
Id int NOT NULL IDENTITY (1, 1),
TestDate dbo.JalaliDate NULL
)  ON [PRIMARY]
GO
```

همین‌طور که مشاهده می‌کنید؛ امکان به‌کارگیری DataType تعریف‌شده وجود دارد.
اکنون چند رکورد درون این جدول درج می‌کنیم:

```
Insert into TestTable (TestDate) Values ('1392/02/09'),('1392/02/09 22:40'),('1392/12/30 22:40')
```

پس از اجرای این دستور خطای زیر در پایین صفحه‌ی SQL Server نمایان می‌شود:

این خطا به این خاطر است که CLR را در SQL Server فعال نکرده ایم. جهت فعال‌کردن CLR دستور زیر را اجرا کنید:

```
sp_configure 'clr enabled', 1
Reconfigure
```

بار دیگر دستور درج را اجرا می‌کنیم:

```
Insert into TestTable (TestDate) Values ('1392/02/09'),('1392/02/09 22:40'),('1392/12/30 22:40')
```

	Id	TestDate
▶	1	1392/02/09 00:00:00
	2	1392/02/09 22:40:00
	3	1392/12/29 22:40:00
*	NULL	NULL

ملاحظه می‌کنید که داده‌ها در جدول مربوطه ذخیره شده است. در رکورد نخست چون ساعت، دقیقه و ثانیه تعریف نشده است؛ به طور هوشمند صفر درج شده است. در رکورد دوم، ساعت و دقیقه مقدار دارد ولی ثانیه صفر ثبت شده است. و در رکورد سوم چون سال 1392 کبیسه نیست؛ به صورت هوشمند آخرین روز ماه به جای روز ثبت شده است. هرچند می‌توان با دست‌کاری در توابع سی‌شارپ، این قوانین را عوض کرد.

اکنون زمان آن رسیده است که توسط یک پرس‌وجو، همه‌ی توابعی که در سی‌شارپ برای این نوع داده نوشتیم، بیازماییم.
پرس‌وجوی زیر را اجرا کنید:

```
Select TestDate.ToString() as JalaliDateTime,
          TestDate.GetDate() as JalaliDate, TestDate.GetTime() as JalaliTime,
          TestDate.ToGregorianTime() as GregorianTime,
          TestDate.JalaliDateAdd('Day',1) JalaliTomorrow,
          TestDate.Month as JalaliMonth from TestTable
```

خروجی این پرس‌وجو به شکل زیر خواهد بود:

	JalaliDateTime	JalaliDate	JalaliTime	GregorianTime	JalaliTomorrow	JalaliMonth
1	1392/02/09 00:00:00	1392/02/09	00:00:00	2013-04-29 00:00:00.000	1392/02/10 00:00:00	2
2	1392/02/09 22:40:00	1392/02/09	22:40:00	2013-04-29 22:40:00.000	1392/02/10 22:40:00	2
3	1392/12/29 22:40:00	1392/12/29	22:40:00	2014-03-20 22:40:00.000	1393/01/01 22:40:00	12

البته درباره‌ی ستون پنجم و ششم شما می‌توانید روی همه‌ی اجزای تاریخ افزایش و کاهش داشته باشید و هم‌چنین می‌توانید با تابع مربوطه هر کدام از اجزای زمان را جداگانه به دست بیاورید که در این مثال عدد ماه نشان داده شده است.

نیازی به گفتن نیست که می‌توانید به سادگی از توابع مربوط به DateTime در SQL Server بهره ببرید. برای مثال برای به دست

آوردن فاصله‌ی میان دو روز از پرس‌وجوی زیر استفاده کنید:

```
Declare @a JalaliDate  = '1392/02/07 00:00:00'
Declare @b JalaliDate = '1392/02/05 00:00:00'

SELECT DATEDIFF("DAY",@b.ToGregorianTime(),@a.ToGregorianTime()) AS DiffDate
```

شاد و پیروز باشید.

نظرات خوانندگان

نویسنده: ali
تاریخ: ۱۳۹۲/۰۲/۱۰ ۰:۵

Nice article

Thanks

نویسنده: سام ناصری
تاریخ: ۱۳۹۲/۰۲/۱۰ ۴:۲۲

بسیار خوب.
فقط من تو مقاله شما دلیلی برای اینکه چرا زمان و تاریخ را میخواهی به این صورت ذخیره کنی متوجه نشدم؟ چرا به همان شکل استانداردش ذخیره نکنیم؟
دیگر اینکه نوع داده جدید به چه شکل در دیتابیس ذخیره میشود. Sql Server از کجا میداند که باید چگونه لیترالها را پارس کند(چگونه متود Parse برگزیده میشود)؟ عملگرهای مقایسه چگونه کار خواهند کرد؟

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۰ ۷:۸

با درود
به پرسش‌های شما در بخش دوم پاسخ خواهم داد.

نویسنده: قاسم
تاریخ: ۱۳۹۲/۰۲/۱۰ ۹:۴۲

سلام، میشه مراحل انجام کار توی VS2010 هم بنویسید، فقط مراحل کار. خیلی ممنون میشم

نویسنده: فرشید علی اکبری
تاریخ: ۱۳۹۲/۰۲/۱۰ ۹:۵۵

سلام
کدهای شمارو درست همونطوریکه گفتین کپی کردم ولی موقعی که Publish رو میزنم پیغام زیر رو میده :
Publish cannot begin until your project(s) build successfully
و پنجره publish ظاهر نمی‌شه و هیچ جایی هم برای تعریف کانکشن وجود نداره... مشکل از کجاست؟
دوم اینکه clr رو هربار که خواستیم روی یک دیتابیس جدید این نوع رو تعریف کنیم باید فعال کنیم... مثلا توی ایجاد سال مالی جدید که سیستم یک دیتابیس خام ایجاد میکنه تا مانده حسابها رو بهش انتقال بدیم قبلش باید دستور فعال سازی clr رو هم مجددا بدیم یافقط توی زمان طراحی کفایت میکنه ؟
سوم اینکه : در مورد استفاده اون توی EF Code First هم اگه نکته‌ی خاصی وجود داره محبت کنین ممنون میشم.
شاد وپیروز باشید.

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۱:۴

با درود
New Project -> Database -> SQL Server -> Visual C# SQL CLR Database Project
- تنظیمات اتصال به پایگاه داده ها
- انتخاب دکمه Yes

- Add New Item -> User Defined Type
- کپی کدها
- استفاده از Deploy در منوی Build یا استفاده از روشی که در بخش دوم نوشتار آموزش داده خواهد شد.

نویسنده: حامد قنادی
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۱:۲۶

با درود
-1 لطفاً پروژه را پیش از Publish یک بار Rebuild کنید. احتمالاً به خاطر یک ارور خاص Publish نمی‌شود. اگر به راحتی Rebuild شده ولی باز هم Publish کار نمی‌کند؛ می‌توانیم با روش دیگری که در بخش دوم آموزش می‌دهم DLL را به SQL معرفی کنیم.
-2 کافی است یک بار پس ساخت Database این پرس‌وجو را اجرا کنید.
-3 تست نکردم ولی به نظر می‌رسد این نوع داده از سمت EF شبیه به Hierarchy باشد.

نویسنده: ش.د
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۴:۳۵

آیا فقط در sql2012 قابل اجرا می‌باشد؟

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۴:۵۴

در متن نوشتن از نگارش 2005 به بعد اس کیوال سرور این قابلیت استفاده از افزونه‌های CLR اضافه شدن.

نویسنده: سید امیر سجادی
تاریخ: ۱۳۹۲/۰۲/۱۰ ۱۸:۳۸

با تشکر از مطلب مفیدتون.
چند تا سوال برام پیش اومده.
اول اینکه آیا به صورت یک DLL به بانک اضافه میشه؟
دوم اینکه اگه از بانک بک آپ بگیریم و جایی دیگه خواستیم اون رو ریستور کنیم چی میشه؟
آپلود بانک روی هاست (بک آپ یا اتچ) ؟

نویسنده: امیر بختیاری
تاریخ: ۱۳۹۲/۰۲/۱۱ ۹:۴۳

به صورت اسمبلی به بانک اضافه میشه
وقتی بک آپ بگیرید و ریستور کنید همراه بک آپ این اسمبلی هم انقال داده می‌شود
در اتچ هم به همین شکل
فقط زمانی که از این اسمبلی در توابع و پروسیجرها استفاده می‌کنید نمی‌تونید حذفش کنید و در صورت تغیرات باید اسمبلی را به روز کنید.
یک مورد دیگه که من زیاد تو هاست‌های شیر شده بهش برخوردم اینه که این امکان توشون فعال نیست و ادمین سرورها هم به سختی این امکان را فعال می‌کنند پس اگر خواستید از این امکان استفاده کنید ابتدا از فعال بودن آن مطمئن شوید

نویسنده: rahim
تاریخ: ۱۳۹۳/۰۵/۱۱ ۱۲:۱۴

با تشکر از مطلب مفیدتون
من زمانی که از این روش استفاده میکنم در هنگام درج رکورد جدید با پیغام خطای زیر مواجه میشم.
:"A .NET Framework error occurred during execution of user-defined routine or aggregate "JalaliDate

.System.OverflowException: Value was either too large or too small for an unsigned byte

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۸/۲۲ ۰:۲۳

با سلام؛ من زمانی که این کد را در SQl 2012 میکنم یک Error در خصوص ToGregorianTime() نمایش داده میشه. لطفا راهنمایی بفرمایید. با تشکر

```
Msg 6522, Level 16, State 2, Line 1
A .NET Framework error occurred during execution of user-defined routine or aggregate
"SpatialDateTime":
System.FormatException: String was not recognized as a valid DateTime.
System.FormatException:
at System.DateTimeParse.ParseExactMultiple(String s, String[] formats, DateTimeFormatInfo dtfi,
DateTimeStyles style)
at System.DateTime.ParseExact(String s, String[] formats, IFormatProvider provider, DateTimeStyles
style)
at System.Data.SqlTypes.SqlDateTime.Parse(String s)
at SpatialDateTime.ToGregorianTime().
```

نویسنده: محسن خان
تاریخ: ۱۳۹۳/۰۸/۲۲ ۰:۵۵

یک try/catch بذار، تا بتونی تاریخ مشکل دار رو پیدا کنی:

```
var pers = new PersianCalendar();
var date = pers.ToDateTime(this.Year, this.Month, this.Day, this.Hour, this.Minute, this.Second,
0).ToString();
try
{
   return SqlDateTime.Parse(date);
}
catch(Exception ex)
{
  throw new InvalidOperationException("Can't parse "+ date);
}
```