**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**مبانی توابع در SQL**

وقتی خطوط برنامه ما زیاد می شود درک، پیگیری و رصد، خطایابی یا اشکال زدایی و دیگر اعمال بر روی برنامه بسیار ملال آور و دشوار خواهد شد. توابع ابزاری هستند که به ما در بهبود برنامه کمک شایانی می کنند و برنامه نویسی ساخت یافته را ارائه می دهند، بدین معنا که برنامه اصلی به تکه های منطقی و مجزای کوچکتری تقسیم می شود که توابع نام دارند. یک تابع وظیفه ای شبیه به یک ماشین دارد که یک سری ورودی را می گیرد و با انجام عملیات بر روی ورودی های دریافتی (پارامتر یا آرگومان)، خروجی یا خروجی هایی را برمی گرداند.

تابع یا Function یک تکه کد است که برای اجرای عملیات مجزا بکار می رود. پس از اجرای تکلیف و عملیات محول شده به آن، می توان به نتایج یا خروجی تابع دسترسی داشت. در SQL نیز توابع کاربردی مشابه به توابع در دیگر زبان ها دارند، بدین معنی که می توان از آن ها برای اجرای عملیات مختلف بهره گرفت.

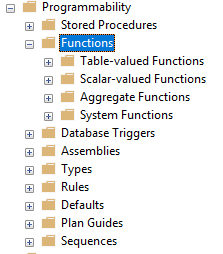
در زبان Transact-SQL ، تابع یک شی محسوب می شود. پس از ایجاد یا تعریف آن، تابع در پایگاه داده ذخیره می شود، سپس می توان آن را در صورت نیاز و هر زمان که لازم بود صدا زده و اجرا کرد.

**انواع توابع**

SQL Server توابع توکار یا از پیش تعریف شده فراوانی را برای ما فراهم می کند. این توابع را می توان به صورت زیر دسته بندی نمود:

در حالت کلی Function به 3 گروه تقسیم کرد

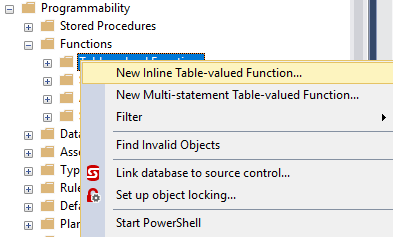
* Table Valued Function
* Scalar Valued Function
* CLR Integration (متدی برای اتصال به دیگر زبان های برنامه نویسی)



**Table Valued Function**

اين توابع خود به 2 دسته تقسيم مي شوند:

* Inline table-Statement
* Multi table – Statement



توابع Inline  نوع خاصي از توابع Multi-table هستند، يعني در حالت کلي شما بايد از توابع Multi-Statement استفاده کنيد .  
شما از توابعي که جدول برمي گردانند ، دقيقا مانند يک جدول حقيقي مي توانيد استفاده کنيد .  
يعني مي توانيد يک يا چند ستون پانرا انتخاب کنيد يا آنرا با يک جدول يا چند جدول ديگر join  کنيد و ...

در inline شما فقط یه Select دارید که نتیجه اونو بر می گردانید اما توی Multi-statement شما می تونید یه جدول با ساختار مشخص بسازید اطلاعات توش بریزید و تابع اون جدول رو برای شما بر می گردونه.  
استفاده صحيح از اين گونه توابع مي تواند تاثير بسيار مناسبي در کارايي و سرعت برنامه هاي شما بگذارد و گاهي مسائل مشکل را با استفاده صحيح از اين توابع به راحتي مي توانيد حل کنيد .

|  |
| --- |
| CREATE FUNCTION [GetTable](@CatID int)  RETURNS TABLE  AS  RETURN  (  SELECT \* FROM Categories  ); |
| CREATE FUNCTION GetTable1()  RETURNS @T1 TABLE  (CatName NVARCHAR(50),ForumsName NVARCHAR(50))  AS  BEGIN  INSERT INTO @T1(CatName,ForumsName)  SELECT Categories.CatName, Forums.ForumsName FROM Categories INNER JOIN Forums ON Categories.ID = Forums.CatID  RETURN;  END |

**طریقه استفاده از GetTable1**

|  |  |
| --- | --- |
| select \* from [GetTable](1) | SELECT \* FROM GetTable1() |

**حذف کردن GetTable1**

|  |
| --- |
| Drop FUNCTION GetTable1 |

**Scalar Function**

خروجي هاياين توابع تک مقداري مي با شدکه معمولا براي انجام محاسبات به کار برده مي شود

|  |
| --- |
| CREATE FUNCTION GetTable1(@CatID int)  RETURNS nvarchar(50)  AS  begin  declare @CategoryName nvarchar(50)  SELECT @CategoryName = CatName from Categories  WHERE ID = @CatID  RETURN @CategoryName  end |

**طریقه استفاده از GetTable1**

|  |
| --- |
| select \*, dbo.GetTable1(1) from Categories |
| select \* from Categories where CatName = dbo.GetTable1(1) |

**حذف کردن GetTable1**

|  |
| --- |
| Drop FUNCTION GetTable1 |

**مثال برای Scalar Valued Function**

|  |  |
| --- | --- |
| INSERT [dbo].[Table\_1] ([Month]) VALUES (1)  INSERT [dbo].[Table\_1] ([Month]) VALUES (2)  INSERT [dbo].[Table\_1] ([Month]) VALUES (3)  INSERT [dbo].[Table\_1] ([Month]) VALUES (4)  INSERT [dbo].[Table\_1] ([Month]) VALUES (5)  INSERT [dbo].[Table\_1] ([Month]) VALUES (6)  INSERT [dbo].[Table\_1] ([Month]) VALUES (7)  INSERT [dbo].[Table\_1] ([Month]) VALUES (8)  INSERT [dbo].[Table\_1] ([Month]) VALUES (9)  INSERT [dbo].[Table\_1] ([Month]) VALUES (10)  INSERT [dbo].[Table\_1] ([Month]) VALUES (11)  INSERT [dbo].[Table\_1] ([Month]) VALUES (12) | CREATE TABLE [Table\_1]([Month] [tinyint] ULL) |

**Scalar برای تبدیل عدد به ماه معادل**

|  |
| --- |
| CREATE FUNCTION [GetMonth] (@Month INT)  RETURNS NVARCHAR(50)  AS  BEGIN  RETURN CASE @Month WHEN 1 THEN N'Farvardin' WHEN 2 THEN N'Ordibehesht'  WHEN 3 THEN N'Khordad' WHEN 4 THEN N'Tir'  WHEN 5 THEN N'Mordad' WHEN 6 THEN N'Shahrivar'  WHEN 7 THEN N'Mehr' WHEN 8 THEN N'Aban'  WHEN 9 THEN N'Azar' WHEN 10 THEN N'Dei'  WHEN 11 THEN N'Bahman' WHEN 12 THEN N'Esfand'  ELSE N'Not' END  END |

**طریقه استفاده از GetTable1**

|  |
| --- |
| SELECT \*,dbo.[GetMonth](Month) FROM Table\_1 |

**حذف کردن GetTable1**

|  |
| --- |
| Drop FUNCTION GetMonth |

**CLR integration**

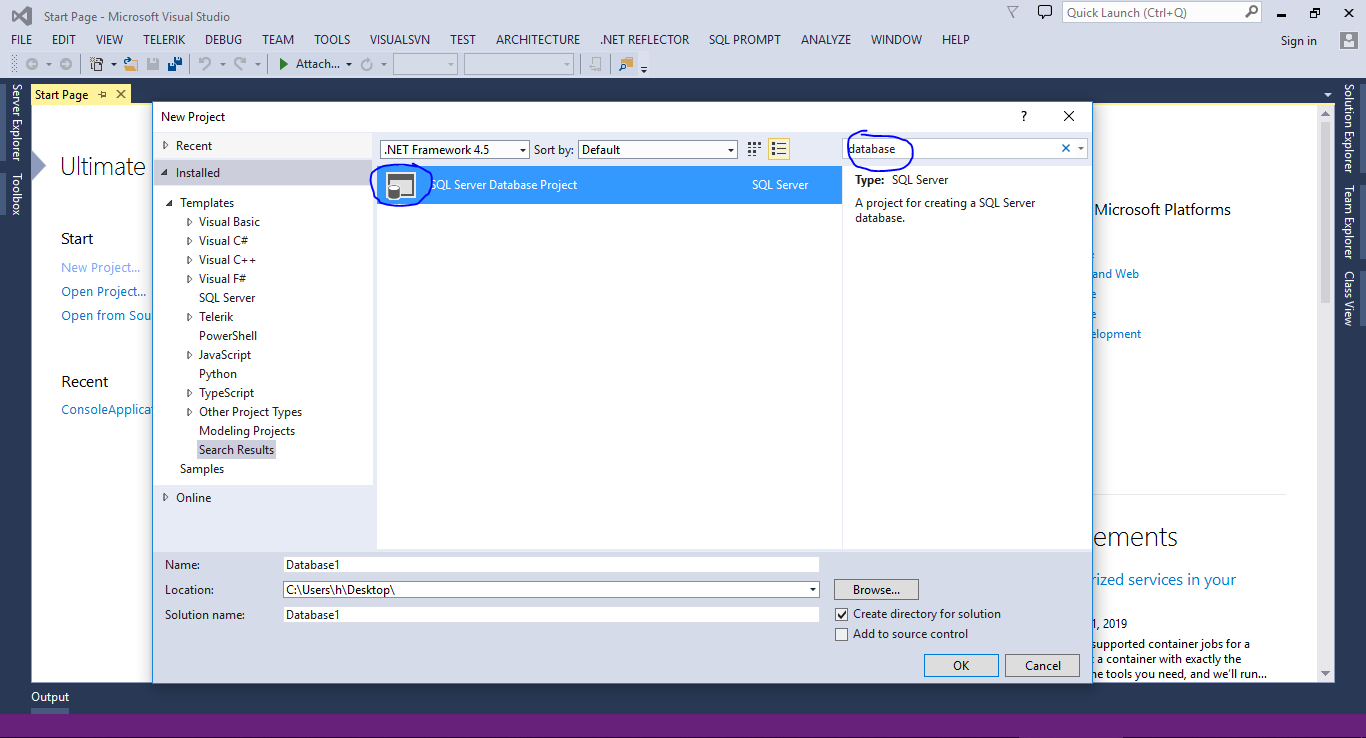
در SQL Server 2005 اين قابليت يه SQL Server  اضافه شد Common Language Runtime که توابعي که توسط  يکي از زبانهايي که .NET Framework  را پشتيباني مي کنند نوشته شده اند را بتوان در SQL Server  استفاده نمود، يعني يه اين معني که شما بوسيله يکي از زبانهاي .NET تابعي يا کدي را مي نويسيد و سپس مي توانيد از اين کد داخل SQL Server استفاده کنيد .

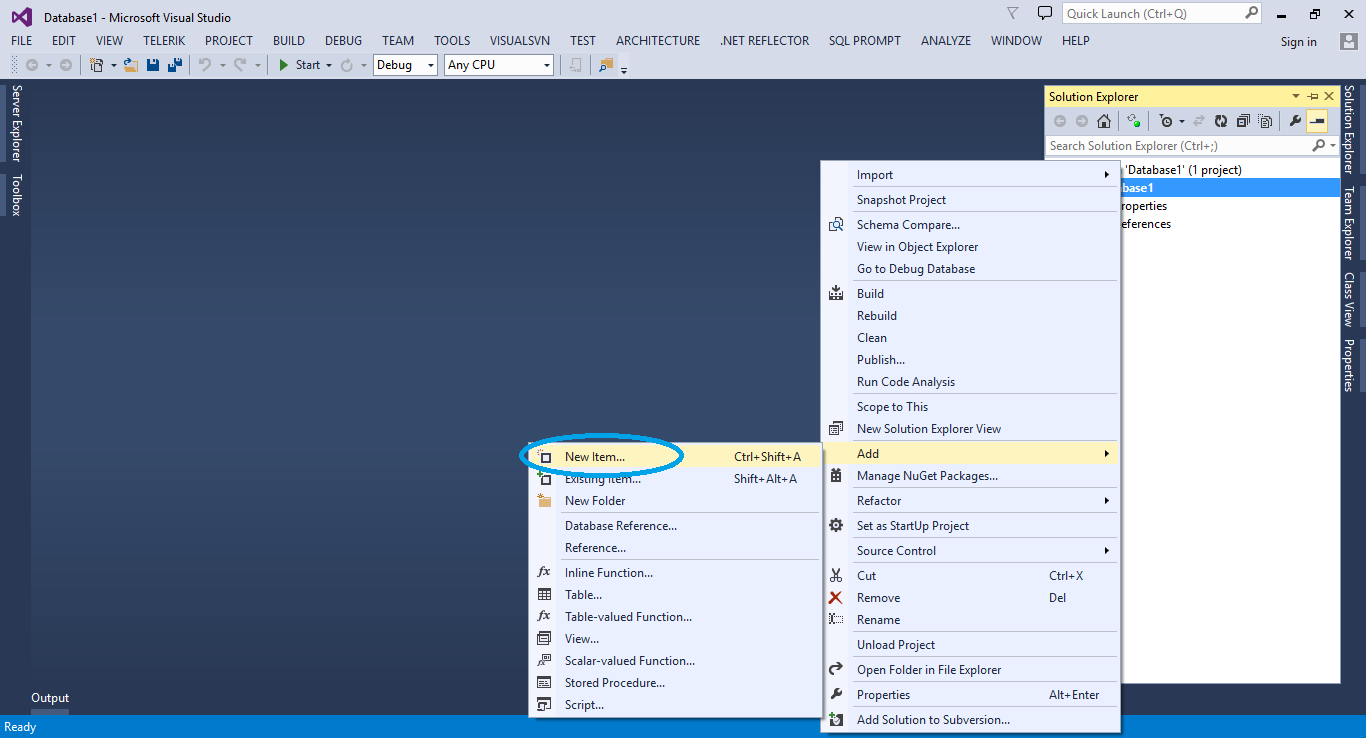
خطوط زیر را جهت فعالسازی CLR در New Query جدید در SQL اجرا نمائید:

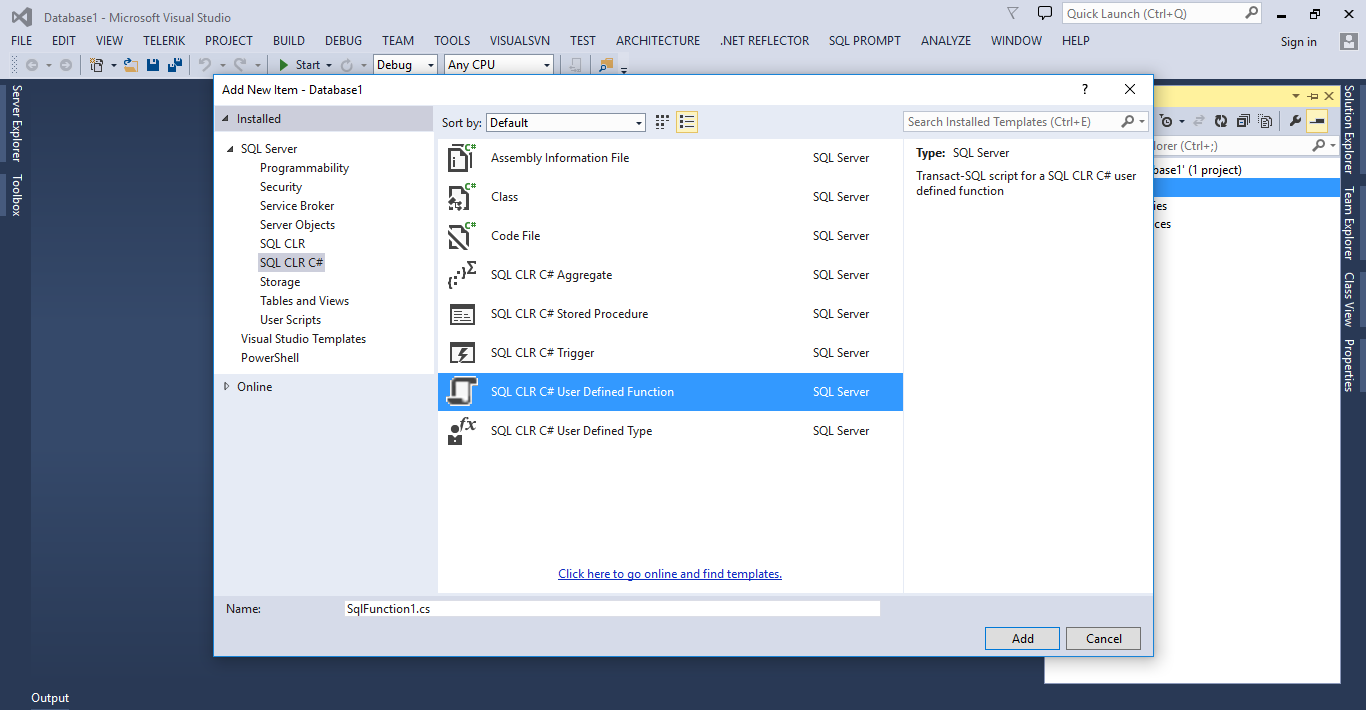
|  |
| --- |
| IF NOT EXISTS ( SELECT value  FROM sys.configurations  WHERE name = 'clr enabled'  AND value = 1)  BEGIN  EXEC sp\_configure @configname = clr\_enabled, @configvalue = 1;  RECONFIGURE;  END |
| EXEC sp\_configure @configname = clr\_enabled, @configvalue = 0;  RECONFIGURE; |

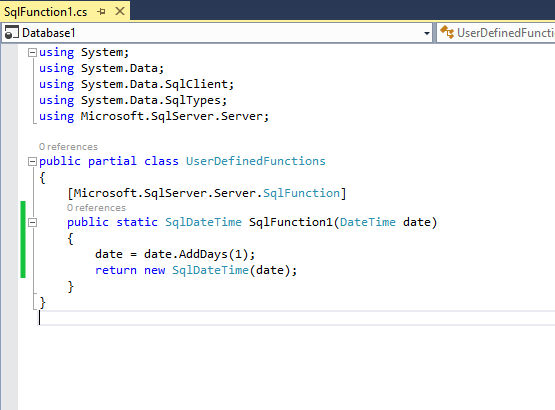
آموزش گام به گام ساخت Stored Procedure CLR

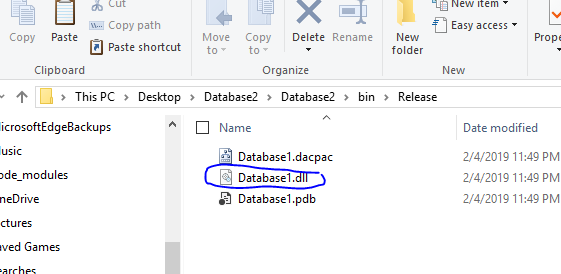
1. ویژوال استودیوی ۲۰۱۰ یا بالاتر را اجرا کنید.
2. یک پروژه ی جدید بسازید.
3. پروژه ی پایگاه داده را انتخاب کنید.
4. پروژه را بسازید که به یک پایگاه داده مرجع نیز اجتیاج دارید.
5. در Solution Explorer بر روی نام پروژه راست کلیک کنید و گزینه ی افزودن را انتخاب کنید.
6. روش ذخیره سازی را انتخاب کنید
7. فایل جدیدی به نام “myTestStoredProcedure” را بیافزایید.

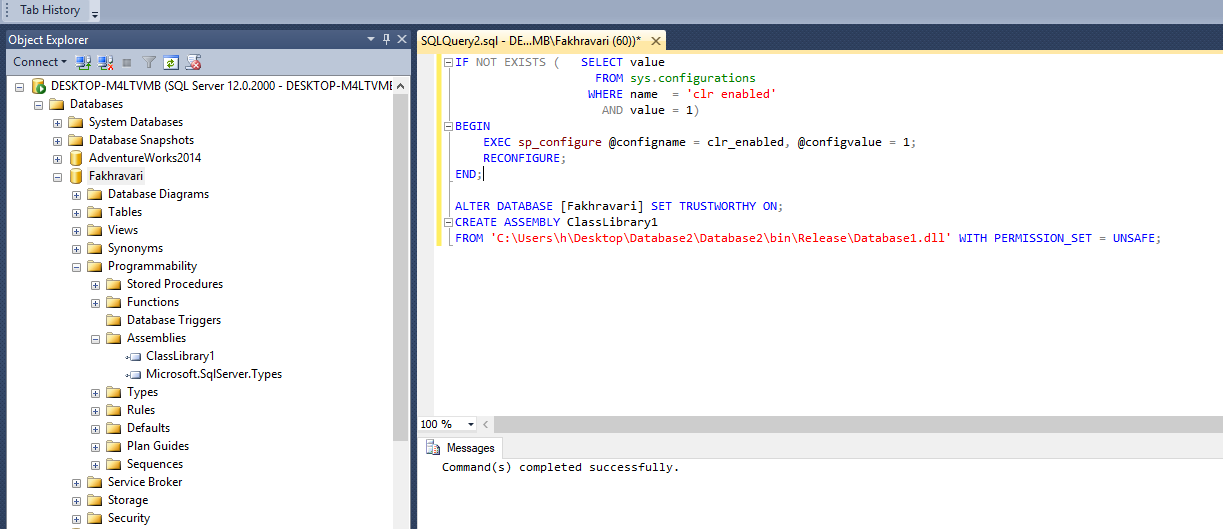


****

****







|  |
| --- |
| ALTER DATABASE [Fakhravari] SET TRUSTWORTHY ON;  CREATE ASSEMBLY ClassLibrary1  FROM 'C:\Users\h\Desktop\Database2\Database2\bin\Release\Database1.dll' WITH PERMISSION\_SET = UNSAFE; |
| DROP ASSEMBLY ClassLibrary1 |

**مپ کردن اسمبلی به پروژه**

|  |
| --- |
| CREATE FUNCTION dbo.GetNextDate (@pDateTime DATETIME NULL)  RETURNS DATETIME  AS  EXTERNAL NAME ClassLibrary1.UserDefinedFunctions.SqlFunction1  GO  SELECT GETDATE(),dbo.GetNextDate(GETDATE()) |

