

شروع کار با SQL Server

درس دهم: توابع تعریف شده توسط کاربر

سید کاوه احمدی

User Defined Functions

- Scalar Functions
- Table-valued Functions
 - Inline
 - Multi-statement

Scalar Functions

```
CREATE FUNCTION function_name(parameters)
RETURNS output_type
AS
    BEGIN
        SQL Statement
    RETURN Output
END
```

مثال ۱

- پرسجوئی بنویسید که عنوان اخباری را بازگرداند که تعداد نظرات آنها بیشتر از متوسط تعداد نظرات ارسال شده برای یک خبر باشد.

```

SELECT
    n.Title, COUNT(c.Id)
FROM
    News n
LEFT OUTER JOIN
    Comments c
ON
    n.Id = c.NewsId
GROUP BY
    n.Title
HAVING
    COUNT(c.Id) > (
        SELECT
            AVG(cnt)
        FROM (
            SELECT
                COUNT(*) AS cnt
            FROM
                Comments
            GROUP BY
                NewsId
        ) AS NewsCount
    )

```

■ روش اول از درس‌های گذشته:

در محاسبه میانگین فقط اخباری را در نظر می‌گیرد که دارای نظر هستند

مثال ۱

■ روش دوم

```
SELECT
    n.Title, COUNT(c.Id)
FROM
    News n
LEFT OUTER JOIN
    Comments c
ON
    n.Id = c.NewsId
GROUP BY
    n.Title
HAVING
    COUNT(c.Id) > (
        SELECT
            AVG(cnt)
        FROM (
            SELECT
                CAST(COUNT(c2.NewsId) AS float) AS cnt
            FROM
                News n2
            LEFT OUTER JOIN
                Comments c2
            ON
                C2.NewsID = n2.Id
            GROUP BY
                n2.Id
        ) AS NewsCount
    )
)
```

مثال ۱

■ روش سوم

— تابعی که متوسط تعداد نظرات را باز می‌گرداند

```
CREATE FUNCTION avgComments()  
RETURNS float  
AS  
BEGIN  
    DECLARE @comments float  
    SELECT @comments = count(*) FROM Comments  
  
    DECLARE @news float  
    SELECT @news = count(*) FROM News  
  
    RETURN(@comments / @news)  
END
```

مثال ۱

■ روش دوم

– فراخوانی تابع در پرسجو برای بررسی شرط مورد نظر

```
SELECT
    n.Title
FROM
    News n
LEFT OUTER JOIN
    Comments c
ON
    n.ID = c.NewsID
GROUP BY
    n.Title
HAVING
    COUNT(c.ID) > dbo.avgComments()
```



```
SELECT dbo.avgComments()
```

مثال ۲: تبدیل تاریخ شمسی به میلادی

```
CREATE FUNCTION ShamsiDate (@intDate DATETIME)
RETURNS NVARCHAR(max)
BEGIN
    ...

    RETURN @DayDate
END
```

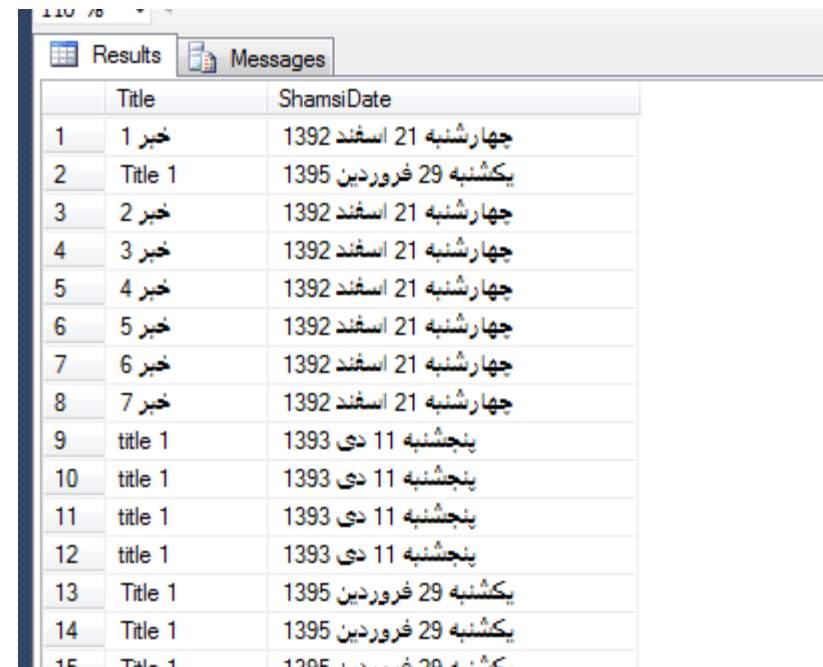
مثال ۲: تبدیل تاریخ میلادی به شمسی

SELECT

Title, dbo.ShamsiDate ([Date]) AS ShamsiDate

FROM

News



	Title	ShamsiDate
1	خبر 1	چهارشنبه 21 اسفند 1392
2	Title 1	یکشنبه 29 فروردین 1395
3	خبر 2	چهارشنبه 21 اسفند 1392
4	خبر 3	چهارشنبه 21 اسفند 1392
5	خبر 4	چهارشنبه 21 اسفند 1392
6	خبر 5	چهارشنبه 21 اسفند 1392
7	خبر 6	چهارشنبه 21 اسفند 1392
8	خبر 7	چهارشنبه 21 اسفند 1392
9	title 1	پنجشنبه 11 دی 1393
10	title 1	پنجشنبه 11 دی 1393
11	title 1	پنجشنبه 11 دی 1393
12	title 1	پنجشنبه 11 دی 1393
13	Title 1	یکشنبه 29 فروردین 1395
14	Title 1	یکشنبه 29 فروردین 1395
15	Title 1	یکشنبه 29 فروردین 1395

- پرسجویی بنویسید که عنوان، متن و تعداد نظرات خبرها را بازگرداند.

```
SELECT
    N.Title,
    N.Body,
    (
        SELECT
            COUNT(*)
        FROM
            Comments C
        WHERE
            N.Id = C.NewsId
    )
FROM
    News N
```

■ روش دوم

— تابعی که تعداد نظرات را بشمرد:

```
CREATE FUNCTION CommentCount(@NewsId int)
RETURNS int
BEGIN
    RETURN (
        SELECT
            COUNT(*)
        FROM
            Comments
        WHERE
            NewsID = @NewsId
    )
END
```

```
SELECT
    Title,
    Body,
    dbo.CommentCount(Id)
FROM
    News
```

- چنانچه در تابع Scalar از پرسجویی استفاده کنید، پرسجوهای درون تابع به دلیل وجود دستورات مختلف T-SQL در تابع، با پرسجوی اصلی ادغام نمی شود.
- در صورتی که از این دسته توابع در بند SELECT استفاده کنیم، به ازای هر رکوردی که پرسش اصلی واکنشی می کند، تابع و پرسجوهای داخل آن یکبار اجرا می شود.
- در حالتی که تابع بتواند به شکل ادغام شدنی با پرسش اصلی نوشته شود، استفاده از تابع می تواند کارایی را کاهش دهد.
- توجه کنید که هنگام بررسی IO یا STATISTICS یا Execution Plan، توابع فراخوانی شده در نظر گرفته نمی شود.


```
SELECT
    n.Title,
    c.Name
FROM
    News n
LEFT OUTER JOIN
    Categories c
ON
    c.Id = n.CategoryID
```

```
CREATE FUNCTION GetCategory(@CatId int)
RETURNS nvarchar(100)
BEGIN
    RETURN (
        SELECT
            Name
        FROM
            Categories
        WHERE
            Id = @CatId
    )
END
```

```
SELECT
    n.Title,
    dbo.GetCategory(CategoryID)
FROM
    News n
```

Inline Table-Valued Functions

```
CREATE FUNCTION function_name (parameters)  
RETURNS TABLE  
AS  
RETURN  
    SELECT Statement
```

Inline Table-Valued Functions

```
CREATE FUNCTION TopNewsFn(@num int)
RETURNS TABLE
AS
RETURN
    SELECT TOP (@num)
        *
    FROM
        News
```

```
SELECT * FROM dbo.TopNewsFn(5)
```

Inline Table-Valued Functions

- در این نوع توابع همانند دیدها فقط می‌توان یک پرسجوی `SELECT` نوشت با این تفاوت که در دیدها نمی‌توان از پارامتری برای کنترل بخش‌های مختلف دستور `SELECT` استفاده کرد. از این رو به این توابع `Parameterized View` نیز گفته می‌شود.
- از آنجایی که این توابع فقط یک دستور `SELECT` را اجرا می‌کنند، می‌توانند با پرسش اصلی ادغام شوند.

```
CREATE FUNCTION PagingNews(@pageNumber int, @pageSize int)
RETURNS TABLE
AS
RETURN
    SELECT
        *
    FROM
        News
    ORDER BY
        [Date]
    OFFSET
        (@pageSize * (@pageNumber - 1)) ROWS
    FETCH NEXT
        @pageSize ROWS ONLY
```

} SQL Server 2012

```
SELECT
    n.Title,
    n.Body,
    c.Name,
    a.Name
FROM
    dbo.PagingNews(1, 10) n ←
LEFT OUTER JOIN
    Categories c
ON
    n.CategoryId = c.Id
INNER JOIN
    Authors a
ON
    n.AuthorId = a.Id
```

Multi-Statement Table-Valued Functions

```
CREATE FUNCTION function_name(parameters)  
RETURNS table_name TABLE table_definition  
AS  
RETURN  
    BEGIN  
        SQL Statement  
    RETURN  
END
```

Multi-Statement Table-Valued Functions

- **Comparing Inline and Multi-Statement Table-Valued Functions**
 - <http://www.sqlservercentral.com/blogs/discussionofsqlserver/2012/02/15/comparing-inline-and-multistatement-table-valued-functions/>

[WITH ENCRYPTION]

- باعث می‌شود که محتوای تریگر دیگر قابل نمایش و تغییر نباشد.