

## استفاده شده از SQL 2008

## روش کار : 1- دریافت پارامتر ورودی به صورت رشته

2- درج عناوین اعداد، ارزش مکانی اعداد صحیح و اعشاری هرکدام در یک جدول

3- جدا کردن ارقام صحیح و اعشاری

4- جداکردن سه رقم سه رقم اعداد صحیح و انتقال آنها به جدول مربوطه

5- Join جداول عناوین و ارقام جدا شده

6- ارسال ارقام اعشاری به همین تابع

7- مشخص کردن ارزش مکانی رقم اعشار

8- اتصال رشته حروف صحیح و اعشاری

در آخر این مطلب کد این تابع را به صورت کامل، برای دانلود قرار داده ام.

## بررسی قسمت‌های مختلف کد

برای اینکه محدودیتی در تعداد ارقام صحیح و اعشاری نداشته باشیم، پارامتر ورودی را از نوع VARCHAR می‌گیریم. پس باید ورودی را بررسی کنیم تا رشته عددی باشد.

## بررسی رشته ورودی:

```
-- @pNumber پارامتر ورودی
IF LEN(ISNULL(@pNumber, '')) = 0 RETURN NULL

IF (PATINDEX('%[^0-9.-]%', @pNumber) > 0)
  OR (LEN(@pNumber) - LEN(REPLACE(@pNumber, '-', '')) > 1)
  OR (LEN(@pNumber) - LEN(REPLACE(@pNumber, '.', '')) > 1)
  OR (CHARINDEX('-', @pNumber) > 1)
RETURN 'خطا'

IF PATINDEX('%[^0]%', @pNumber) = 0 RETURN 'صفر'
IF (CHARINDEX('.', @pNumber) = 1) SET @pNumber='0'+@pNumber

DECLARE @Negative AS VARCHAR(5) = '';
IF LEFT(@pNumber, 1) = '-';
BEGIN
  SET @pNumber = SUBSTRING(@pNumber, 2, 100)
  SET @Negative = 'منفی'
END
```

- بررسی NULL، خالی بودن و یا داشتن فاصله در رشته، با دانستن اینکه تابع LEN فاصله‌های آخر یک رشته را در نظر نمی‌گیرد.

- بررسی رشته ورودی برای پیدا کردن کاراکتر غیر عددی، نقطه و منفی. بررسی تعداد علامت منفی و نقطه که بیشتر از یک مورد نباشند، و در نهایت بررسی اینکه علامت منفی در ابتدای رشته ورودی باشد.

- بررسی صفر بودن ورودی (0)، مقدار ورودی شروع شونده با ممیز (0.213) و مقدار عددی منفی (-21210.0021). چیز دیگری به ذهنم نرسید!

## درج عناوین در جداول مربوطه:

فکر کنم اینجا به علت وجود کاراکترهای فارسی و انگلیسی کد کمی بهم ریخته نمایش داده می‌شود.

```
DECLARE @NumberTitle TABLE (val INT, Title NVARCHAR(100));
```

```

INSERT INTO @NumberTitle (val,Title)
VALUES(0, ''),(1, 'چهار',4),(2, 'سه',3),(3, 'دو',2),(4, 'یک',1),
(5, 'هشت',8),(6, 'شش',6),(7, 'هفت',7),(8, 'هشت',8),(9, 'نه',9),
(10, 'ده',10),(11, 'یازده',11),(12, 'دوازده',12),(13, 'شانزده',13),
(14, 'چهارده',14),(15, 'پانزده',15),(16, 'شانزده',16),(17, 'هفده',17),
(18, 'هجده',18),(19, 'نوزده',19),(20, 'بیست',20),(30, 'سی',30),
(40, 'چهل',40),(50, 'پنجاه',50),(60, 'شصت',60),(70, 'هفتاد',70),
(80, 'هشتاد',80),(90, 'نود',90),(100, 'صد',100),(200, 'دویست',200),
(300, 'سیصد',300),(400, 'چهارصد',400),(500, 'پانصد',500),(600, 'ششصد',600),
(700, 'هفتصد',700),(800, 'هشتصد',800),(900, 'نهمصد',900)

DECLARE @PositionTitle TABLE (id INT,Title NVARCHAR(100));
INSERT INTO @PositionTitle (id,title)
VALUES (1, ''),(2, 'هزار',2),(3, 'میلیون',3),(4, 'میلیارد',4),(5, 'تریلیون',5),
(6, 'کوادریلیون',6),(7, 'کوینتیلیون',7),(8, 'سیکستیلیون',8),(9, 'سپتیلیون',9),
(10, 'اکتیلیون',10),(11, 'نونیلیون',11),(12, 'دسیلیون',12),(13, 'تریسیلیون',13),
(14, 'دودسیلیون',14),(15, 'تریسیلیون',15),(16, 'کوادرتریسیلیون',16),
(17, 'کوینتدسیلیون',17),(18, 'سیکسدسیلیون',18),(19, 'سپتندسیلیون',19),
(20, 'اکتودسیلیوم',20),(21, 'نومدسیلیون',21)

DECLARE @DecimalTitle TABLE (id INT,Title NVARCHAR(100));
INSERT INTO @DecimalTitle (id,Title)
VALUES(1, 'دهم',1),(2, 'صدم',2),(3, 'هزارم',3),(4, 'ده-هزارم',4),
(5, 'صد-هزارم',5),(6, 'میلیون ام',6),(7, 'ده-میلیون ام',7),
(8, 'صد-میلیون ام',8),(9, 'میلیارد ام',9),(10, 'ده-میلیارد ام',10)

```

**جداسازی رقم صحیح و اعشاری:** عدد ورودی ممکن است حالت‌های مختلفی داشته باشد مثل: 0.000000 , 00002. , 123 , 234.4344000000000

بنابراین براساس ممیز، قسمت صحیح را از اعشاری جدا می‌کنیم. برای ورودی که با ممیز شروع شود، در ابتدا تابع بررسی می‌کنیم و عدد صفر را به رشته اضافه می‌کنیم.

بعد از ممیز و اعداد بزرگتر از یک، با **صفرهای بی ارزش چه کنیم؟** شاید اولین چیزی که به ذهن برسد استفاده از حلقه (WHILE) برای حذف صفرهای بی ارزش قسمت ممیز باشد؛ ولی من ترجیح می‌دهم که از روش دیگری استفاده کنم :

برعکس کردن رشته قسمت اعشاری، پیدا کردن مکان اولین عدد غیر صفر منهای یک ، و کم کردن عدد بدست آمده از طول رشته اعشاری، قسمت مورد نظر ما را برخواهد گرداند:

```
SUBSTRING(@DecimalNumber,1, len(@DecimalNumber )-PATINDEX('%[^0]%', REVERSE (@DecimalNumber))-1)
```

اما اگر عدد ورودی **20.0** باشد همچنان صفر بی ارزش بعد از ممیز را خواهیم داشت. برای رفع این مشکل کافی است که **کاراکتری غیر از صفر** را به اول رشته اعشاری اضافه کنیم. من از علامت **'?'** استفاده کردم. پس به علت اضافه کردن کاراکتر، استارت را از 2 شروع کرده و دیگر نیازی به 1- نخواهیم داشت. با کد زیر قسمت صحیح و اعشاری را بدست می‌آوریم:

```

DECLARE @IntegerNumber NVARCHAR(100),
@DecimalNumber NVARCHAR(100),
@PointPosition INT =case CHARINDEX('.', @pNumber) WHEN 0 THEN LEN(@pNumber)+1 ELSE CHARINDEX('.',
@pNumber) END

SET @IntegerNumber= LEFT(@pNumber, @PointPosition - 1)
SET @DecimalNumber= '?' + SUBSTRING(@pNumber, @PointPosition + 1, LEN(@pNumber))
SET @DecimalNumber= SUBSTRING(@DecimalNumber,2, len(@DecimalNumber )-PATINDEX('%[^0]%', REVERSE
(@DecimalNumber)))

SET @pNumber= @IntegerNumber

```

**جداد کردن سه رقم سه رقم :** - بدست آوردن یکان، دهگان و صدگان

- برای قسمت دهگان، اگر عددی بین 10 تا 19 باشد به صورت کامل (مثلا 15) و در غیر این صورت فقط رقم دهگان. برای بدست آوردن یکان اگر دو رقم آخر بین 10 و 19 بود صفر و در غیر این صورت یکان برگردانده می‌شود و در جدول MyNumbers درج می‌گردد.

```

DECLARE @Number AS INT
DECLARE @MyNumbers TABLE (id INT IDENTITY(1, 1), Val1 INT, Val2 INT, Val3 INT)

WHILE (@pNumber) <> '0'
BEGIN
    SET @number = CAST(SUBSTRING(@pNumber, LEN(@pNumber) - 2, 3) AS INT)

    INSERT INTO @MyNumbers
    SELECT (@Number % 1000) - (@Number % 100),
    CASE
    WHEN @Number % 100 BETWEEN 10 AND 19 THEN @Number % 100
    ELSE (@Number % 100) - (@Number % 10)
    END,
    CASE
    WHEN @Number % 100 BETWEEN 10 AND 19 THEN 0
    ELSE @Number % 10
    END
    END

    IF LEN(@pNumber) > 2
        SET @pNumber = LEFT(@pNumber, LEN(@pNumber) - 3)
    ELSE
        SET @pNumber = '0'
    END
END

```

سطری که تمام مقادیر آن صفر باشد برای ما بی ارزش محسوب می‌شود، مانند سطر یک در عکس زیر (جدول MyNumbers) برای عدد **1200955000** :

	id	Val1	Val2	Val3
1	1	0	0	0
2	2	900	50	5
3	3	200	0	0
4	4	0	0	1

**استفاده از JOIN : JOIN** کردن جدول اعداد با عناوین عددی براساس ارزش آن‌ها و JOIN جدول اعداد با جدول ارزش مکانی براساس ID به صورت نزولی (شماره سطر).

```

DECLARE @Str AS NVARCHAR(2000) = '';
SELECT @Str += REPLACE(REPLACE(LTRIM(RTRIM(nt1.Title + ' ' + nt2.Title + ' ' + nt3.title)), ' ', ','), ', ', ' و ')
+ ' ' + pt.title + ' و '
FROM @MyNumbers AS mn
INNER JOIN @PositionTitle pt
ON pt.id = mn.id
INNER JOIN @NumberTitle nt1
ON nt1.val = mn.Val1
INNER JOIN @NumberTitle nt2
ON nt2.val = mn.Val2
INNER JOIN @NumberTitle nt3
ON nt3.val = mn.Val3
WHERE (nt1.val + nt2.val + nt3.val > 0)
ORDER BY pt.id DESC

```

Replace داخلی: جایگزین کردن "دو فاصله‌ی خالی" با "یک فاصله‌ی خالی"

Replace بیرونی: جایگزینی فاصله‌های خالی با ' و '

همانطور که در بالا اشاره کردم سطرهایی که Val3, Val2, Val1 آن صفر باشد برای ما بی ارزش هستند، پس آنها را با شرط نوشته شده حذف می‌کنیم.

**بدست آوردن مقدار اعشاری:** خوب! حالا نوبت به عدد اعشاری می‌رسد. برای بدست آوردن حروف، مقدار اعشاری بدست آمده

را به همین تابع ارسال می‌کنیم و برای بدست آوردن عنوان ارزش مکانی، براساس طول اعشار (ID) آن را در جدول مربوطه پیدا می‌کنیم.

اگر عدد ورودی مثلاً 0.355 باشد، تابع باید صفر اول را شناسایی و قسمت عناوین اعشاری را به آن اضافه کند، که این کار با شرط ذیل انجام می‌شود.

اگر رشته اعشار بدون مقدار باشد، تابع مقدار NULL بر می‌گرداند (قسمت بررسی رشته ورودی) و هر رشته ای که با NULL جمع شود برابر با NULL خواهد بود. در این صورت با توجه به کد زیر مقداری به رشته Str به عنوان قسمت اعشاری، اضافه نمی‌گردد.

```
IF @IntegerNumber='0'  
SET @Str=CASE WHEN PATINDEX('%[^0]%', @DecimalNumber) > 0 THEN @Negative ELSE '' END + 'صفر'  
ELSE  
SET @Str = @Negative + LEFT (@Str, LEN(@Str) -2)  
  
DECLARE @PTitle NVARCHAR(100)=ISNULL((SELECT Title FROM @DecimalTitle WHERE id=LEN(@DecimalNumber)), '')  
SET @Str += ISNULL(' ممیز '+[dbo].[fnNumberToWord_Persian](@DecimalNumber) + ' '+@PTitle, '')  
RETURN @str
```

**مثال: رشته ' 5445789240.54678000000000 ' رشتۀ**

پنج میلیارد و چهارصد و چهل و پنج میلیون و هفتصد و هشتاد و نه هزار و دویست و چهل و چهار پنجاه و چهار هزار و ششصد و هفتاد و هشت صد-هزارم

[دانلود فایل](#)

## نظرات خوانندگان

نویسنده: میثم چگینی  
تاریخ: ۱۱:۴۱ ۱۳۹۲/۰۸/۰۱

با تشکر

یک مورد کوچک داره که برای نمایش درست نیاز به اضافه کردن N قبل از تمام کلمات فارسی می‌باشد  
'صفر' N

نویسنده: حمیدرضا عابدینی  
تاریخ: ۱۰:۵۳ ۱۳۹۲/۰۸/۰۳

حق با شماست!

من چون از Collation نوع Persian\_100\_CI\_AI استفاده کردم همه‌ی Nها را حذف کردم. ولی برای دیگر موارد برای اینکه به صورت علامت ؟ ظاهر نشود، همانطور که شما گفتید باید از N استفاده کرد.

ممنون

نویسنده: مهیار  
تاریخ: ۲۰:۳۸ ۱۳۹۲/۱۱/۱۵

سلام،

ممنون از راه حلی که ارائه دادید.

برای حذف صفرهای غیرضروری موجود در اعشار پیشنهاد من کد زیر است که ساده‌تر و خواناتر است:

```
declare @value varchar(50) = '0.1010000000'  
select replace(rtrim(replace(@value, '0', ' ')), ' ', '0');
```

توضیح کد: ابتدا تمام صفرهای موجود در رشته را تبدیل به کاراکتر Space میکنیم توسط تابع Replace سپس با تابع Rtrim تمام کاراکترهای Space آخر رشته را Remove می‌کنیم و مجدد کاراکترهای Space را برمیگردانیم به صفر.

نویسنده: مهیار  
تاریخ: ۲۱:۳۷ ۱۳۹۲/۱۱/۱۵

کد مربوط به حلقه While اتان هم کمی ساده‌تر کردم مخصوصاً اولین دستور حلقه که مربوط میشه به انتخاب سه رقم آخر رشته. ضمناً نیازی نیست که صراحتاً متغیر را به integer تبدیل کنید. با صفر جمع یا با یک ضرب کنید تا بصورت Implicit تبدیل صورت بگیره:

```
WHILE (@pNumber) <> '0'  
BEGIN  
    SET @number = RIGHT(@pNumber, 3) + 0  
  
    INSERT INTO @MyNumbers  
    SELECT  
        @Number / 100 * 100,  
        CASE  
            WHEN nbr BETWEEN 10 AND 19 THEN nbr  
            ELSE nbr / 10 * 10  
        END,  
        CASE  
            WHEN nbr BETWEEN 10 AND 19 THEN 0  
            ELSE nbr % 10  
        END  
    FROM (SELECT @Number % 100)S(nbr);  
  
    IF LEN(@pNumber) > 2
```

```
        SET @pNumber = LEFT(@pNumber, LEN(@pNumber) - 3)
    ELSE
        SET @pNumber = '0'
END
```

نویسنده: حمیدرضا عابدینی  
تاریخ: ۲۳:۳۲ ۱۳۹۲/۱۱/۲۰

سلام مهیار جان!  
کد بسیار خوب و خوانایی گذاشتید، ممنونم!  
در نتیجه:

```
DECLARE @IntegerNumber NVARCHAR(100),
        @DecimalNumber NVARCHAR(100),
        @PointPosition INT =case CHARINDEX('.', @pNumber) WHEN 0 THEN LEN(@pNumber)+1 ELSE CHARINDEX('.',
        @pNumber) END

SET @pNumber=replace(rtrim(replace(@pNumber,'0',' ')), ' ','0');
SET @IntegerNumber= LEFT(@pNumber, @PointPosition - 1)
SET @DecimalNumber= SUBSTRING(@pNumber, @PointPosition+1 , LEN(@pNumber))

SET @pNumber= @IntegerNumber
```