T-SQL function for generating slugs?

Ask Question



Quick check to see if anyone has or knows of a T-SQL function capable of generating slugs from a given nvarchar input. i.e;





"Hello World" > "hello-world" "This is a test" > "this-is-a-test"



I have a C# function that I normally use for these purposes, but in this case I have a large amount of data to parse and turn into slugs, so it makes more sense to do it on the SQL Server rather than have to transfer data over the wire.

As an aside, I don't have Remote Desktop access to the box so I can't run code (.net, Powershell etc) against it

Thanks in advance.

EDIT: As per request, here's the function I generally use to generate slugs:

```
public static string GenerateSlug(string n, int maxLength)
   string s = n.ToLower();
```

```
s = Regex.Replace(s, @"s", "-");
return s;
}

tsql slug

edited Jun 23 '10 at 2:07

asked Jun 21 '10 at 6:44

Jeremy Cade
1,003 1 16 26

Can you post your C# slug-generator function? — Constantin
Jun 22 '10 at 7:16
```

7 Answers



You can use LOWER and REPLACE to do this:

13

SELECT REPLACE(LOWER(origString), ' ', '-')
FROM myTable



For wholesale update of the column (the code sets the slug column according to the value of the origString column:

```
UPDATE myTable
SET slug = REPLACE(LOWER(origString), ' ', '-')
```

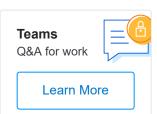
Home

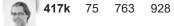
DI IRI IC

Tags

Users

Jobs





4 To properly slugify a unicode string you would need much more than this. At least all non-ascii characters should be dealt with. – Constantin Jun 22 '10 at 7:14



This is what I've come up with as a solution. Feel free to fix / modify where needed.





I should mention that the database I'm currently developing against is case insensitive hence the LOWER(@str).



```
CREATE FUNCTION [dbo].[UDF_GenerateSlug]
(
    @str VARCHAR(100)
)
RETURNS VARCHAR(100)
AS
BEGIN
DECLARE @IncorrectCharLoc SMALLINT
SET @str = LOWER(@str)
SET @IncorrectCharLoc = PATINDEX('%[^0-9a-z ]%',@str)
WHILE @IncorrectCharLoc > 0
BEGIN
SET @str = STUFF(@str,@incorrectCharLoc,1,'')
SET @str = STUFF(@str,@incorrectCharLoc,1,'')
SET @IncorrectCharLoc = PATINDEX('%[^0-9a-z ]%',@str)
END
SET @str = REPLACE(@str,' ','-')
RETURN @str
```

Mention to: http://blog.sqlauthority.com/2007/05/13/sql-server-udf-function-to-parse-alphanumeric-characters-from-string/ for the original code.





Firoze Lafeer **16.9k** 3 46 48

answered Jun 23 '10 at 2:49



Jeremy Cade 1,003 1 16 26

1 It should not be '%[^0-9a-z] %' but should be '%[^0-9a-z-]%' — Aivan Monceller Feb 6 '12 at 21:57 ✓





Here's a variation of Jeremy's response. This might not technically be slugifying since I'm doing a couple of custom things like replacing "." with "-dot-", and stripping out apostrophes. Main improvement is this one also strips out all consecutive spaces, and doesn't strip out preexisting dashes.

```
create function dbo.Slugify(@str nvarchar(max)) returns nva
as
begin
    declare @IncorrectCharLoc int
    set @str = replace(replace(lower(@str),'.',' dot '),''

    -- remove non alphanumerics:
    set @IncorrectCharLoc = patindex('%[^0-9a-z -]%',@str)
    while @IncorrectCharLoc > 0
    begin
        set @str = stuff(@str,@incorrectCharLoc,1,' ')
        set @IncorrectCharLoc = patindex('%[^0-9a-z -]%',@str)
```

```
begin
  set @str = replace(@str, ' ', ' ')
  end
  set @str = replace(@str,' ','-')
return @str
end
```

answered Mar 7 '12 at 2:49



Giscard Biamby 3,761 18 23

Very helpful. Two things I noticed this script didn't handle were trailing (probably leading?) spaces inputs that ended with a closing paren left a trailing hyphen. For the first, trim the input. For the second, I'm not sure because a trailing hyphen might be on purpose... – Carl G Oct 9 '12 at 21:11



3

I took Jeremy's response a couple steps further by removing all consecutive dashes even after spaces are replaced, and removed leading and trailing dashes.



```
create function dbo.Slugify(@str nvarchar(max)) returns nva
begin
    declare @IncorrectCharLoc int
    set @str = replace(replace(lower(@str),'.','-'),'''','

    -- remove non alphanumerics:
    set @IncorrectCharLoc = patindex('%[^0-9a-z -]%',@str)
    while @IncorrectCharLoc > 0
    begin
        set @str = stuff(@str,@incorrectCharLoc,1,' ')
        set @IncorrectCharLoc = patindex('%[^0-9a-z -]%',@end

    -- replace all spaces with dashes
    set @str = replace(@str,' ','-')
```

```
begin
        set @str = replace(@str, '--', '-')
    end
    -- remove leading dashes
   while charindex('-', @str) = 1
    begin
        set @str = RIGHT(@str, len(@str) - 1)
    end
    -- remove trailing dashes
   while len(@str) > 0 AND substring(@str, len(@str), 1) :
   begin
        set @str = LEFT(@str, len(@str) - 1)
    end
return @str
end
                             answered Aug 30 '12 at 13:39
                                    Jason
                                    1,476 2 18 30
```

I know this is an old thread, but for future generation, I found one function that deals even with accents here:

3



```
set @out = ltrim(rtrim(@out))

-- replace space to hyphen
set @out = replace(@out, ' ', '-')

-- remove double hyphen
while CHARINDEX('--', @out) > 0 set @out = replace(@out
return (@out)
END
```

answered Mar 25 '15 at 15:50



Juliana

Using COLLATE SQL_Latin1_General_CP1251_CS_AS to strips accents only works with varchar variables. If you try doing the same with nvarchar variable, nothing happens. If the input is nvarchar it will have to be explicetly converted to varchar at some point using cast(@string as varchar) . If you don't, the accents will stay where they are. — jahu Nov 9 '15 at 15:38

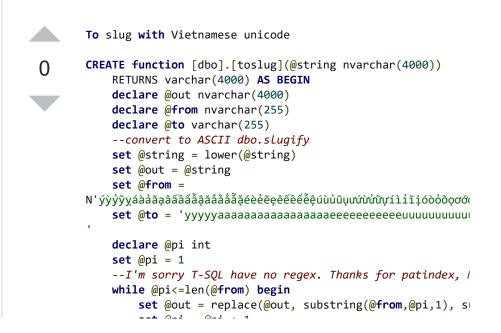
Additionally using the cast with specific length might be required as SQL Server seems to shorten your string if it's converted to varchar inside a SELECT statement. For example cast(@string as varchar(500)) . — jahu Nov 10 '15 at 9:01

```
-- Converts a title such as "This is a Test" to an all lower as "this-is-a-test" for use as the slug in a URL. All I -- (whitespace, underscore, or hyphen) are converted to a second to a
```

```
-- built based on the transitions from the current state to
-- State 0 skips any initial whitespace. State 1 includes
-- characters. State 2 converts multiple separators into (
-- and skips trailing whitespace. State 3 skips any punctu
-- between characters and, if no additional whitespace is e
-- then the punctuation is not treated as a word separator
CREATE FUNCTION ToSlug(@title AS NVARCHAR(MAX))
RETURNS VARCHAR (MAX)
AS
BEGIN
   DECLARE @retval AS VARCHAR(MAX) = ''; -- return value
   DECLARE @i AS INT = 1;
                                         -- title index
   DECLARE @c AS CHAR(1);
                                          -- current charac
   DECLARE @state AS INT = 0;
DECLARE @nextState AS INT;
                                         -- current state
                                          -- next state
   DECLARE @tab AS CHAR(1) = CHAR(9); -- tab
                                          -- line feed
   DECLARE @lf AS CHAR(1) = CHAR(10);
   DECLARE @cr AS CHAR(1) = CHAR(13); -- carriage retui
   DECLARE @separators AS CHAR(8) = '[' + @tab + @lf + @ci
   DECLARE @validchars AS CHAR(11) = '[a-zA-Z0-9]';
   WHILE (@i <= LEN(@title))</pre>
    BEGIN
        SELECT @c = SUBSTRING(@title, @i, 1),
        @nextState = CASE
            WHEN @c LIKE @validchars THEN 1
            WHEN @state = 0 THEN 0
            WHEN @state = 1 THEN CASE
                WHEN @c LIKE @separators THEN 2
                ELSE 3 -- unknown character
                END
            WHEN @state = 2 THEN 2
            WHEN @state = 3 THEN CASE
                WHEN @c LIKE @separators THEN 2
                ELSE 3 -- stay in state 3
                END
            END,
        @retval = @retval + CASE
            WHEN @nextState != 1 THEN ''
            WHEN @state = \emptyset THEN LOWER(@c)
            WHEN @state = 1 THEN LOWER(@c)
            WHEN @state = 2 THEN '-' + LOWER(@c)
```

```
@state = @nextState,
         @i = @i + 1
    END
    RETURN @retval;
END
                                edited Mar 26 '13 at 3:44
                                answered Mar 26 '13 at 3:37
                                       Frank Hellwig
```

could you add more comments? - Aleš Mar 26 '13 at 3:55



```
-- replace space to hyphen
set @out = replace(@out, ' ', '-')

-- remove double hyphen
while CHARINDEX('--', @out) > 0 set @out = replace(@out
return (@out)
END
```

edited Mar 8 '17 at 9:53

answered Mar 8 '17 at 9:26



Nguyen Quy

- 1 Please take a look at how your post has rendered, and edit it to fix the formatting. You get a handy little preview in the postediting area so you can see what it's going to look like before you submit it. AakashM Mar 8 '17 at 9:37
- 2 This link may be useful to you <u>How to Answer</u> Tom Mar 8 '17 at 9:48

thanks @AakashM I got it - Nguyen Quy Mar 8 '17 at 9:55 🖍