bite-sized.sql

SQL SERVER: REPLACE VS. TRANSLATE

bite-sized.sql

SQL Server REPLACE vs. TRANSLATE

Suppose you have some text littered with characters you need to remove

☹ You want to turn this:

```
'the [TRANSLATE] function can help you with
(cleaning up) {dirty} text strings'
```

😀 Into this:

```
'the TRANSLATE function can help you with
cleaning up dirty text strings'
```

bite-sized.sql

SQL Server REPLACE vs. TRANSLATE

1. You can nest REPLACE several times – once for each character you want to remove

```
DECLARE @str nvarchar(100) = 'the [TRANSLATE]
function can help you with (cleaning up) {dirty}
text strings';

SELECT REPLACE(
            REPLACE(
                REPLACE(
                    REPLACE(
                        REPLACE(
                            REPLACE(@str,'(',''),
                        ')',''),
                    '[',''),
                ']',''),
            '{',''),
        '}','');
```

Each character needs exactly one call to REPLACE

bite-sized.sql

SQL Server REPLACE vs. TRANSLATE

2 Or you can use TRANSLATE

```
DECLARE @str nvarchar(100) = 'the [TRANSLATE]
function can help you with (cleaning up) {dirty}
text strings';
DECLARE @chars nvarchar(6) = '()[]{}';

SELECT REPLACE(
            TRANSLATE(
                @str,
                @chars,
                REPLICATE('~',LEN(@chars))
            )
        ,'~','');
```

REPLICATE repeats its 1st param the number of times specified in the 2nd

Because the 3rd param of TRANSLATE must be the same length as the 2nd param, it can't contain an empty string. So we use REPLACE once to replace the tildes with an empty string

Use TRANSLATE to replace the chars with a tilde (or some other character)

TRANSLATE takes three parameters:

1. The string where you want to make replacements
2. The characters you want to remove
3. The characters you want to put in their place

bite-sized.sql

SQL Server REPLACE vs. TRANSLATE

```sql
DECLARE @str nvarchar(100)
    = 'the [TRANSLATE] function can help you with (cleaning up) {dirty} text strings';

SELECT REPLACE(
            REPLACE(
                REPLACE(
                    REPLACE(
                        REPLACE(
                            REPLACE(@str,'(',''),
                        ')',''),
                    '[',''),
                ']',''),
            '{',''),
        '}','') AS nested_replace;

DECLARE @chars nvarchar(6) = '()[]{}';
SELECT REPLACE(
            TRANSLATE(
                @str,
                @chars,
                REPLICATE('~',LEN(@chars))
            )
        ,'~','') AS replace_translate;
```

RESULTS

	nested_replace
1	the TRANSLATE function can help you with cleaning up dirty text strings

	replace_translate
1	the TRANSLATE function can help you with cleaning up dirty text strings