



SQL Server and T-SQL Development Tutorials

Development resources, articles, tutorials, code samples and tools and downloads for ASP.Net, SQL Server, R Script, Windows, Windows Phone, AWS, SAP HANA and ABAP, like SAP UI5, Screen Personas, etc.

[Sign in](#) | [Join](#) |

[Home](#)[Articles](#)[News](#)[IT Jobs](#)[Tools](#)[Sample Chapters](#)[Trainers](#)[Blogs](#)[Forums](#)[Photos](#)[Files](#)

[SQL Server 2019 Installation](#)
[download SQL Server 2019](#)
[download SQL Server 2017](#)
[download SQL Server 2016](#)
[download SQL Server 2014](#)
[download SQL Server 2012](#)

Custom Sequence String as SQL Identity Column in SQL Server

Tweet

Related SQL Resources

[SQL Server Articles](#)

[SQL Server 2012](#)

[SQL Server Tools](#)

[SQL Blog](#)

[MS SQL Server Forums](#)

SQL developers want to create sql identity column not as a numeric value but as a alpha-numeric identity column or as a custom identity column in some cases. In such a cases generally the requirement is a special formatted identity column field with values like 'A0001','A0002', etc.

Unfortunately the build in SQL identity column generation does not support to generate custom identity in SQL Server.

In SQL Server, sql column identity can be managed or created by using SQL Server Management Studio (SSMS). For the easiest way to set a column as an sql identity column in SQL Server database table, administrators or transact-sql developers should modify table column properties "Identity Specification", Identity Increment and Identity Seed properties.

But SQL Server does not allow any alpha-numeric values to be specified for sql identity column specifications which will lead to generate custom format identity column values.

Here is the message that you'll get when you try to play around on SQL column identity in SQL Server table:

Identity seed must be a integral number containing 5 digits or less.

SQL developers can create **SQL Computed Column** instead of SQL Identity Column and provide custom sequence solution for their applications. First of all I'll create a user function in SQL Server in order to return a special formatted string value when I pass an integer number to this sql function. The main parameter of T-SQL function will be an integer input parameter which is representing the counter part of the sequence string. Identity function will add leading zeros in front of the number to have a fixed length of number like '000007' or '000012', etc. So I need to pass how many characters will be the numeric part of the identity to my sql identity function. Then I'll format this sql identity by concatenating a string prefix (like 'SQL-', 'NO-' or 'ID-', etc) to the beginning of this number to end up like an output 'ID-002011' or 'SQL-002012', etc.

The following user function will help Transact-SQL developers to create auto-generated sequences including characters and numbers in it where numbers are increasing sequentially. Please note that this custom identity function will return the input integer value in the desired format as a character value.

```
CREATE FUNCTION udf_ZeroPaddingAndConcat (
    @Prefix NVARCHAR(10),
    @Id INT,
    @Length INT,
    @PaddingChar CHAR(1) = '0'
)
RETURNS NVARCHAR(MAX)
AS
BEGIN

    RETURN (
```

```

SELECT @Prefix + RIGHT(REPLICATE(@PaddingChar, @Length) + CAST(@Id as nvarchar(10)), @Length)
)

END
GO

```

Since the SQL identity column of the source table will generate auto increasing numbers like 1,2,3,4,5,... in a sequence, we need to convert those identity values into structures like 'NO-000001','NO-000002','NO-000003', etc. using udf_ZeroPaddingAndConcat() SQL function.

Let's call SQL Server function udf_ZeroPaddingAndConcat for zero padding and string concatenation with custom prefix

```

SELECT
dbo.udf_ZeroPaddingAndConcat('NO-',1,6,'0'),
dbo.udf_ZeroPaddingAndConcat('NO-',2,6,'0'),
dbo.udf_ZeroPaddingAndConcat('NO-',3,6,'0'),
dbo.udf_ZeroPaddingAndConcat('NO-',4,6,'0')

```

As you can see in the below output of the above sql statement with user function udf_ZeroPaddingAndConcat(), the returned values are all formatted to reflect a string sequence.

	(No column name)	(No column name)	(No column name)	(No column name)
1	NO-000001	NO-000002	NO-000003	NO-000004

Now SQL developers and database administrators can use this user defined SQL Identity function on a database table.

Let's create a sample SQL database table with an identity column named 'Id'

Then create an other column within the table as a computed column (or calculated column) as taking the Id Identity column in SQL table and assigning it to the new column value. The computed column in our sql tutorial is "StringSequence" column.

```

CREATE TABLE CustomIdentityTable
(
    Id int IDENTITY(1,1) NOT NULL,
    StringSequence as dbo.udf_ZeroPaddingAndConcat('ID-', CAST(Id as nvarchar(10)),6,'0'),
    BookName nvarchar(250),
    BookDescription nvarchar(max)
)

```

Please refer to SQL tutorial for more information on [SQL Computed Columns](#)

Now let's insert new rows into our sample database table "CustomIdentityTable" and query the table data if we could generate custom identity values, which are increasing by 1 but in a special string format

```

insert into CustomIdentityTable select 'T-SQL Recipes',NULL
insert into CustomIdentityTable select 'SQL Server 2008 for Developers',NULL
insert into CustomIdentityTable select 'Transact-SQL',NULL
insert into CustomIdentityTable select 'The Guru's Guide to Transact-SQL',NULL
insert into CustomIdentityTable select 'Inside T-SQL 2008',NULL
select * from CustomIdentityTable

```

As you can see in the following screenshot of the SQL Server Management Studio (SSMS) Query Editor screen, we successfully create sql identity column as a sequence of string values.

	Id	StringSequence	BookTitle	BookDescription
1	1	ID-000001	T-SQL Recipes	NULL
2	2	ID-000002	SQL Server 2008 for Developers	NULL
3	3	ID-000003	Transact-SQL	NULL
4	4	ID-000004	The Guru's Guide to Transact-SQL	NULL
5	5	ID-000005	Inside T-SQL 2008	NULL

Copyright © 2004 - 2019 Eralper YILMAZ. All rights reserved.

