Understanding Character datatypes

MADHIVANAN

cssug@googlegroups.com or www.facebook.com/groups/cssug/
http://www.facebook.com/Madhivanantsql

- Char
- Varchar
- Nchar
- Nvarchar
- Text
- Ntext
- Varchar(max)
- Nvarchar(max)

Char vs varchar

- Stores character data
- Maximum size is 8000 bytes

- Char – mostly used to store data of fixed length

- Varchar - mostly used to store data of varying length

CHAR and trailing spaces

```
declare @c char(10), @vc varchar(10)
select @c='test',@vc='test'
select @c,@vc

Result
 ----------    ----------
test          test

select @c+'a',@vc+'a‘
Result
-----------     -----------
test      a     testa
```

```
select @c='test',@vc='test     '
select @c+'a',@vc+'a'

---------- ----------
test      a test      a

set @vc=@c
select @c+'a',@vc+'a‘

---------- ----------
test      a test      a
```

Data truncation

```
select @c='this is test',@vc='this is test'

select @c,@vc

Result
----------     ----------
this is te    this is te
```

```
declare @t table (c char(10), vc varchar(10))
insert into @t(c,vc)
select 'test','test'
select c,vc,c+'a',vc+'a' from @t
```

Result

```
c          vc
---------- ---------- ---------- ----------
test       test       test     a    testa
```

```
declare @t table (c char(10), vc varchar(10))
insert into @t(c,vc)
select 'this is test','this is test'
select c,vc,c+'a',vc+'a' from @t
```

Result

```
Msg 8152, Level 16, State 14, Line 2
String or binary data would be truncated.
The statement has been terminated.
```

Trailing spaces are omitted during comparison

```
declare @t table(v char(10))
insert into @t (v) values (' '),('')
select 'a'+v+'a' from @t
where v='                                    '
select 'a'+v+'a' from @t
where v=''
```

Result

```
-----------
a         a
a         a
```

Storage comparison

- Never use varchar(1)

```
create table #t1(c char(1))

insert into #t1 (c)
select top 1000000 left(cast(newid() as varchar(36)),1) from
sys.columns as c1 cross join sys.columns as c2 cross join sys.columns as c3

create table #t2(vc varchar(1))

insert into #t2 (vc)
select top 1000000 left(cast(newid() as varchar(36)),1) from
sys.columns as c1 cross join sys.columns as c2 cross join sys.columns as c3
```

```
exec tempdb..sp_spaceused '#t1'
exec tempdb..sp_spaceused '#t2'
```

	name	rows	reserved	data	index_size	unused
1	#t1_____...	1000000	10888 KB	10872 KB	8 KB	8 KB

	name	rows	reserved	data	index_size	unused
1	#t2_____...	1000000	13896 KB	13848 KB	8 KB	40 KB

```
create table #t3(c char(10))

insert into #t3 (c)
select top 1000000 left(cast(newid() as varchar(36)),5) from
sys.columns as c1 cross join sys.columns as c2 cross join sys.columns as c3

create table #t4(vc varchar(10))

insert into #t4 (vc)
select top 1000000 left(cast(newid() as varchar(36)),5) from
sys.columns as c1 cross join sys.columns as c2 cross join sys.columns as c3
```

```
exec tempdb..sp_spaceused '#t3'
exec tempdb..sp_spaceused '#t4'
```

	name	rows	reserved	data	index_size	unused
1	#t3_____...	1000000	18824 KB	18784 KB	8 KB	32 KB

	name	rows	reserved	data	index_size	unused
1	#t4_____...	1000000	17864 KB	17824 KB	8 KB	32 KB

- Char(N) – needs N bytes
- Varchar(N) – needs Actual Data Length+1 extra byte
- Extra byte stores the information about actual data length

- ISO synonym

```
declare @c character(10)
declare @vc char varying (10)
declare @vc character varying (10)
```

ANSI Padding OFF

- Always use CHAR datatype with NOT NULL constraint
- CHAR(N) NULL will be internally treated as VARCHAR(N)

Convertion

```
select cast('test' as char(3)),cast('test' as varchar(3))

select cast('1001' as char(3)),cast('1001' as varchar(3))
```

Result

```
---- ----
tes  tes

---- ----
100  100
```

```
select cast(1001 as char(3)),cast(1001 as varchar(3))

Result

---- ----
*    *
```

- Always specify the length

```
declare @vc varchar
set @vc='test'
select @vc

Result
----
t
```

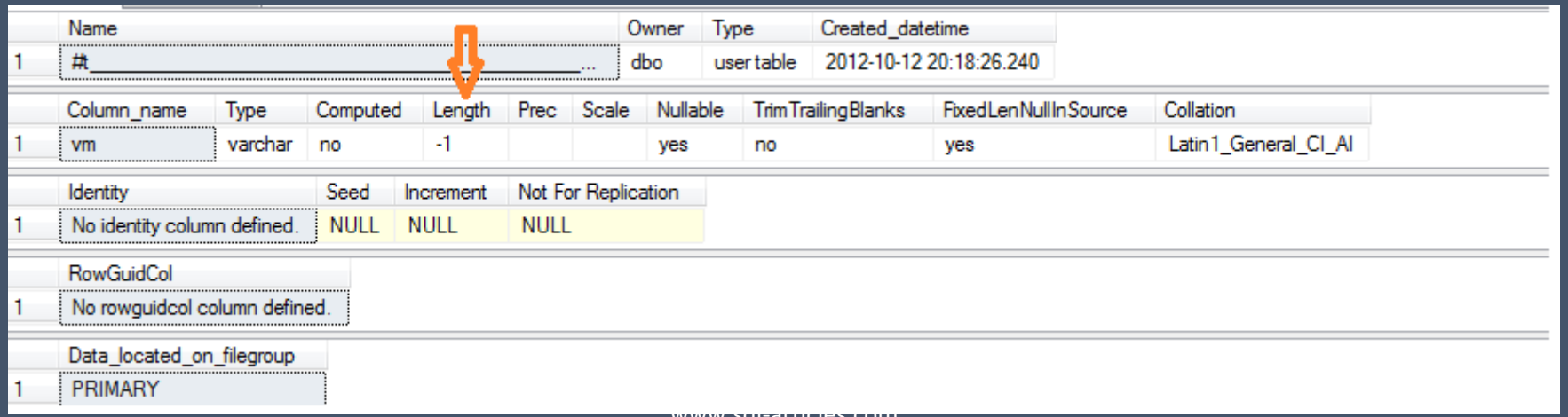
```
select cast('Do you see everything after this
  convertion?' as varchar)

Result

----------------------------

Do you see everything after th
```

VARCHAR(MAX)

- Stores 2GB of data
- No datatype for char(max)

- Length will be marked as -1

```
create table #t(vm varchar(max))
EXEC tempdb..sp_help '#t'
```



	Name	Owner	Type	Created_datetime
1	#t_____...	dbo	user table	2012-10-12 20:18:26.240

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	vm	varchar	no	-1			yes	no	yes	Latin1_General_CI_AI

	Identity	Seed	Increment	Not For Replication
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

PRINT statement

- Return type of PRINT statement is varchar(8000) or nvarchar(4000)

```
declare @date datetime
set @date ='20120101'
--  Select @date – returns 2012-01-01 00:00:00.000
print @date
```

Result

Jan 1 2012 12:00AM

- Anything that can not be implicitly converted to varchar will throw error

```
declare @n xml
set @n ='4'
--  select @n returns 4
Print @n
Result

Msg 257, Level 16, State 3, Line 4
Implicit conversion from data type xml to nvarchar is not allowed. Use the CONVERT function to run this query.
```

Print Statement can return unicode strings too

print 'சாதனை'

Returns

?????

print N'சாதனை‘

Returns

சாதனை

Nchar Vs nvarchar

- Stores unicode data
- Maximum size to be specified is 4000
- Data should be added with prefix N
- Unicode prefix N stands for **National**

- Return type of FORMAT function is NVARCHAR

```
declare @d datetime
set @d='20101219 22:10:23'
select format(@d ,'yyyy-MM-dd') as date_only into
  #t
EXEC tempdb..sp_help '#t'
```

	Name	Owner	Type	Created_datetime
1	#t_____...	dbo	user table	2012-10-12 19:59:50.863

	Column_name	Type	Computed	Length	Prec	Scale	Nullable	TrimTrailingBlanks	FixedLenNullInSource	Collation
1	date_only	nvarchar	no	8000			yes	(n/a)	(n/a)	Latin1_General_CI_AI

	Identity	Seed	Increment	Not For Replication
1	No identity column defined.	NULL	NULL	NULL

	RowGuidCol
1	No rowguidcol column defined.

	Data_located_on_filegroup
1	PRIMARY

Nvarchar(max)

- Stores unicode data
- Stores maximum of 2GB data
- Data should be added with prefix N
- Nchar(max) is not supported

Text Vs Ntext

- Text
  - stores maximum of 1GB of data
  - stores Non unicode data
- Ntext
  - stores maximum of 1GB of data
  - stores unicode data

Text Vs Ntext

- Limitations
  - String functions can not be directly used

```
declare @t table(t text)
insert into @t (t)
select 'testing'
select left(t,3) from @t
```

Result

```
Msg 8116, Level 16, State 1, Line 4
Argument data type text is invalid for argument 1 of left function.
```

- Local variables can not be declared with these datatypes

```
declare @t text
set @t='testing'
```

Result

```
Msg 2739, Level 16, State 1, Line 3
The text, ntext, and image data types are invalid for local variables.
```

- Can not be part of ORDER BY Clause
- Can not concatenate using + operator

```
declare @t table(t text)
insert into @t (t)
select 'testing'
select t+'a' from @t
```

Result

```
Msg 402, Level 16, State 1, Line 4
The data types text and varchar are incompatible in the add operator.
```

Text Vs Ntext

- Will not be supported in future versions of SQL Server

- Alternates
- varchar(max) for Text
- Nvarchar(max) from ntext

Thank you

- http://beyondrelational.com/blogs/madhivanan
- Twitter - https://twitter.com/madhivanan2001