

قبل از ایجاد Sequence در SQL Server 2012، توضیح مختصری را درباره آن می‌دهم. در واقع Sequence روشی برای تولید اعداد ترتیبی با قابلیت افزایش یا کاهش عددهای دلخواه می‌باشد که توسط کاربر یا برنامه نویس ایجاد می‌شود. بنابراین Sequence ها User-Defined می‌باشند. در اینجا ممکن است سئوالی پیش بیاید که اینکار توسط Identity هم قابل انجام است، اما چرا استفاده از Sequence توسط مایکروسافت پیشنهاد می‌شود. بدلائل زیر استفاده از Sequence بهتر می‌باشد:

ممکن است Application شما قبل از درج رکورد، درون یک جدول نیاز به عدد منحصر بفردی داشته باشد. عدد تولید شده بوسیله Sequence را می‌توانید بین جداول یا ستونهای مختلف یک جدول به اشتراک بگذارید. می‌توانید روند تولید اعداد ترتیبی را Restart نمایید. به عبارت دیگر قابلیت Restart نمودن Sequence وجود دارد. می‌توانید Sequence خود را براساس Sort یک یا چند فیلد، تنظیم نمایید.

Syntax آن به شرح ذیل می‌باشد:

```
CREATE SEQUENCE [schema_name . ] sequence_name
[ AS [ built_in_integer_type | user-defined_integer_type ] ]
[ START WITH <constant> ]
[ INCREMENT BY <constant> ]
[ { MINVALUE [ <constant> ] } | { NO MINVALUE } ]
[ { MAXVALUE [ <constant> ] } | { NO MAXVALUE } ]
[ CYCLE | { NO CYCLE } ]
[ { CACHE [ <constant> ] } | { NO CACHE } ]
[ ; ]
```

شرح Syntax :

در زمان ایجاد Sequence، نوع آن می‌بایست عددی باشد، چنانچه Type آن را مشخص ننمایید، SQL Server، نوع آن را bigint در نظر می‌گیرد.

Start With: بدین مفهوم می‌باشد، که Sequence ایجاد شده از چه عددی آغاز شود.

INCREMENT BY: مفهومی این است که Sequence به چه مقداری افزایش یا کاهش یابد. به عبارت دیگری عدد تولید شده براساس مقدار Increment by تولید می‌شود.

Minvalue: کمترین مقداری که Sequence می‌تواند ایجاد نماید.

Maxvalue: بیشترین مقداری که Sequence می‌تواند ایجاد نماید.

Cycle: مقداری را که برای Cycle تعیین می‌نماییم، بدین مفهوم است که Sequence پس از چه عددی می‌بایست Restart شود.

Cache: عددی که برای Cache در نظر می‌گیریم، مفهومی این است که چه تعداد از اعداد تولید شده توسط Sequence، قبل از استفاده، می‌تواند در Cache قرار گیرد.

در ادامه با یک مثال ساده، یک Sequence ایجاد می‌نماییم:

```
CREATE SEQUENCE [dbo].[SequenceTest]
AS [int]
START WITH 1
INCREMENT BY 1
MINVALUE 1
MAXVALUE 30
CYCLE
CACHE
GO
```

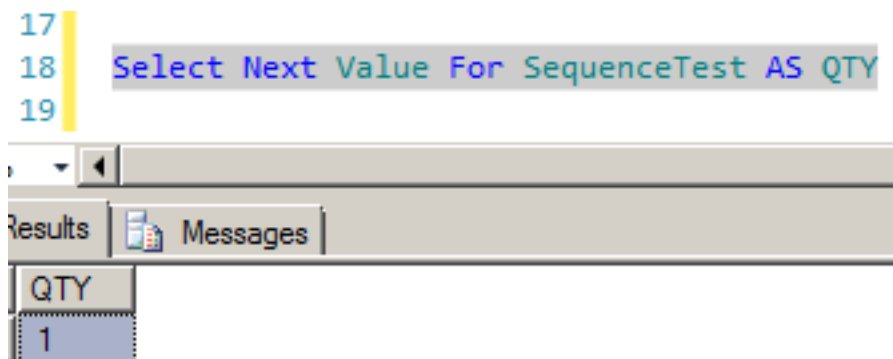
در مثال بالا Start with برابر یک است، یعنی اولین عددی که تولید می‌شود، برابر یک است، INCREMENT BY برابر یک است یعنی

در هر بار فراخوانی Sequence یک عدد به عدد تولید شده قبلی افزوده می‌شود. مقدار Minvalue برابر یک است، یعنی کمترین مقداری که Sequence می‌تواند تولید نماید برابر یک است. مقدار Maxvalue برابر 30 است، یعنی بیشترین مقداری که Sequence می‌تواند تولید نماید برابر 30 می‌باشد. و Cycle هم برای Sequence فوق در نظر گرفته شده است، بدین مفهوم، که Sequence به مقدار 30 برسد، Restart شده و از مقدار یک شروع به تولید اعداد می‌نماید.

برای اینکه بتوانیم مقدار Sequence را بدست آوریم، کفایت از Syntax زیر استفاده نماییم:

```
NEXT VALUE FOR [ database_name . ] [ schema_name . ] sequence_name
[ OVER ( <over_order_by_clause> ) ]
```

به عنوان مثال داریم:



اگر Select بالا را تا 30 بار انجام دهید، برای دفعه 31 مقدار آن یک می‌شود، چون در زمان تعریف Sequence، Cycle را انتخاب کرده بودیم. در غیر اینصورت برای دفعه 31 با خطا زیر مواجه می‌شوید.

```
Msg 11728, Level 16, State 1, Line 1
The sequence object 'SequenceTest' has reached its minimum or maximum value. Restart the sequence
object to allow new values to be generated.
```

یکی از امکانات جالب Sequence این است که شما می‌توانید Sequence را روی یک فیلد Sort شده تنظیم نمایید، برای روش شدن مطلب به مثال زیر توجه نمایید:

در ابتدا مطابق Script زیر جدولی را ایجاد و مقادیری را درون آن درج می‌نماییم:

```
create table Kids
( ID int,
  Name varchar(50)
);
Go
insert Kids
values
(1,'Emma')
, (1,'Tabitha')
, (2,'Kendall')
, (3,'Delaney')
, (4,'Kyle')
, (5,'Jessica')
, (6,'Josh')
, (7,'Kirsten')
, (8,'Amanda')
, (9,'Jimmy')
;
```

سپس یک Schema به نام Sample ایجاد می‌کنیم:

```
CREATE SCHEMA Samples ;  
GO
```

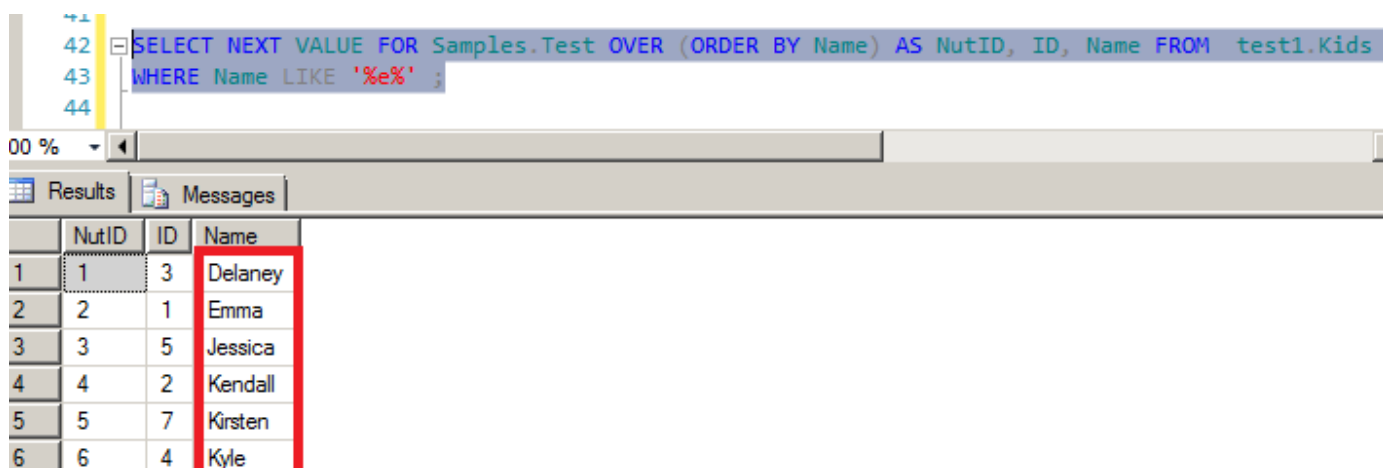
در ادامه یک Sequence به نام Test ایجاد می‌کنیم:

```
CREATE SEQUENCE Samples.Test  
AS tinyint  
START WITH 1  
INCREMENT BY 1 ;  
GO
```

حال Query زیر را اجرا می‌نماییم:

```
SELECT NEXT VALUE FOR Samples.Test OVER (ORDER BY Name) AS NutID, ID, Name FROM test1.Kids  
WHERE Name LIKE '%e%' ;
```

در Query بالا روی فیلد Name به صورت صعودی عملیات Sort انجام می‌شود، و سپس Sequence روی آن اعمال می‌گردد، برای روشن‌تر شدن مطلب خروجی را مشاهده نمایید که تعداد رکورد آن 6 می‌باشد.



	NutID	ID	Name
1	1	3	Delaney
2	2	1	Emma
3	3	5	Jessica
4	4	2	Kendall
5	5	7	Kirsten
6	6	4	Kyle

امیدوارم مطلب فوق مفید واقع شده باشد.

## نظرات خوانندگان

نویسنده:

کاربر

تاریخ:

۱۷:۵۳ ۱۳۹۲/۰۴/۲۹

سلام

مشکلی بر IDENTITY وارد هست بوجود آمدن Gap بین داده هاست مثلاً با حذف پی در پی سطرهای جدول بین مقادیر تولید شده Gap بوجود می‌آید ولی ما این را نمی‌خواهیم، نیاز داریم که همیشه مقادیر identity پشت سر هم و متوالی باشند یعنی با حذف یک سطر میانی تمام مقادیر شیفت پیدا کنند. و برای رفع این مساله هیچ راهی وجود ندارد. حتی اگر بخواهیم با تابع row\_number مقادیر IDENTITY را بعد متوالی کردن بروز رسانی کنیم باز دستور set insert\_identity table\_name on این اجازه را نمی‌دهد یعنی این دستور

```
set insert_identity tbl on
update t
set id = new_id
from (select new_id = row_number() over(order by id), *) t
set insert_identity tbl off
```

در کتابی خواندم که این مشکل با sequence حل میشود.  
میخوام بدانم چگونه؟

نویسنده:

فرهاد فرهمندخواه

تاریخ:

۱۶:۳۰ ۱۳۹۲/۰۵/۰۱

سلام

شما قادر نیستید یک فیلد Identity را بروز رسانی نمایید، دستور set insert\_identity Tablename on به شما اجازه Insert به جدول بدون Identity را می‌دهد، برای اینکه بتوانید Gap مرتبط به فیلد Identity را در جدول برطرف کنید، در ابتدا از جدول مورد نظر خود یک کپی تهیه و جدول اصلی را Truncate کنید، سپس یک Sequencer ایجاد و محتویات جدول کپی را بوسیله Sequencer در جدول اصلی کپی نمایید.  
فرض کنیم جدول اصلی Table\_3 باشد، ابتدا آن را کپی می‌کنیم در جدولی به نام T

```
Select * into T from table_3
```

سپس دستور Truncate را روی جدول Table\_3 اجرا کنید:

```
truncate table dbo.table_3
```

حال یک Sequence ایجاد کنید:

```
CREATE SEQUENCE testEventCounter
AS int
START WITH 1
INCREMENT BY 1 ;
```

در ادامه محتویات جدول کپی را به جدول اصلی منتقل نمایید:

```
SET IDENTITY_INSERT table_3 on
INSERT INTO table_3 (ID, Descriptp)
SELECT
NEXT VALUE FOR testEventCounter AS id
, Descriptp
```

```
FROM T
```

راه دیگر این است که به جای استفاده از Identity از Sequence در فیلد خود استفاده نمایید، بصورت زیر :

```
CREATE TABLE Table3
(
    ID int PRIMARY KEY CLUSTERED
        DEFAULT (NEXT VALUE FOR SequenceTest),
    De nvarchar(300) NULL
);
GO
```

در هنگام ایجاد جدول Sequence را به فیلد ID ست کردیم.  
حال هر زمانی که بخواهید می‌توانید فیلد ID را مطابق Sequence خود بروز رسانی کنید:

```
Update table3 set id=(NEXT VALUE FOR testEventCounter )
```

موفق باشید و امیدوارم مفید واقع شده باشد

نویسنده: کاربر  
تاریخ: ۱۳۹۲/۰۵/۰۸ ۲۰:۸

سلام

خیلی ممنون از پاسخی که دادین. مفید واقع شد.  
البته فکر کنم برای گارانتی کردن ترتیب sequence بر اساس ستون id می‌تونیم از OVER(ORDER BY نیز استفاده کنیم.  
به این شکل:

```
SET IDENTITY_INSERT table_3 on
INSERT INTO table_3 (ID, Descrip)
SELECT
    NEXT VALUE FOR testEventCounter (OVER ORDER BY ID) AS id
    , Descrip
FROM T
```

به هر شکل با مفهوم جدیدی نیز آشنا شدیم (: