# Sequence Numbers

 sequence is a user-defined schema-bound object that generates a sequence of numeric values. The sequence of numeric values is generated in an ascending or descending order at a defined interval and may cycle (repeat) as requested. Sequences, unlike identity columns, are not associated with tables. An application refers to a sequence object to receive its next value. The relationship between sequences and tables is controlled by the application.

## Using Sequences

Use sequences instead of identity columns in the following scenarios:

* The application requires a number before the insert into the table is made.
* The application requires sharing a single series of numbers between multiple tables or multiple columns within a table.
* The application must restart the number series when a specified number is reached. For example, after assigning values 1 through 10, the application starts assigning values 1 through 10 again.
* The application requires sequence values to be sorted by another field. The NEXT VALUE FOR function can apply the OVER clause to the function call. The OVER clause guarantees that the values returned are generated in the order of the OVER clause's ORDER BY clause.
* An application requires multiple numbers to be assigned at the same time. For example, an application needs to reserve five sequential numbers. Requesting identity values could result in gaps in the series if other processes were simultaneously issued numbers. Calling sp\_sequence\_get\_range can retrieve several numbers in the sequence at once.
* You need to change the specification of the sequence, such as the increment value.

## Limitations

Unlike identity columns, whose values cannot be changed, sequence values are not automatically protected after insertion into the table. To prevent sequence values from being changed, use an update trigger on the table to roll back changes.

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 } ]

[ ; ]

CREATE SEQUENCE Test.CountByNeg1 START WITH 0 INCREMENT BY -1 ;

CREATE SEQUENCE Test.CountBy1 START WITH 1 INCREMENT BY 1 ;

CREATE SEQUENCE Test.CountBy1 START WITH 5 INCREMENT BY 5 ;

SELECT \* FROM sys.sequences WHERE name = 'TestSequence' ;

CREATE SEQUENCE Test.DecSeq AS decimal(3,0) START WITH 125

INCREMENT BY 25 MINVALUE 100 MAXVALUE 200 CYCLE CACHE 3

CREATE SEQUENCE CustomerSequence AS int

START WITH 1

INCREMENT BY 1

MINVALUE 1

MAXVALUE 100

CYCLE

UPDATE Customers SET PartitionNumber = NEXT VALUE FOR CustomerSequence

DROP SEQUENCE CustomerSequence

# NEXT VALUE FOR

Generates a sequence number from the specified sequence object.

NEXT VALUE FOR [ database\_name . ] [ schema\_name . ] sequence\_name

[ OVER (<over\_order\_by\_clause>) ]