

Oracle CREATE SEQUENCE

Summary: in this tutorial, you will learn how to use the Oracle CREATE SEQUENCE statement to create a new sequence in Oracle.

Introduction to Oracle CREATE SEQUENCE statement

The CREATE SEQUENCE statement allows you to create a new sequence in the database.

Here is the basic syntax of the CREATE SEQUENCE statement:

```
CREATE SEQUENCE schema_name.sequence_name  
[INCREMENT BY interval]  
[START WITH first_number]  
[MAXVALUE max_value | NOMAXVALUE]  
[MINVALUE min_value | NOMINVALUE]  
[CYCLE | NOCYCLE]  
[CACHE cache_size | NOCACHE]  
[ORDER | NOORDER];
```

Code language: SQL (Structured Query Language) (sql)

CREATE SEQUENCE

Specify the name of the sequence after the CREATE SEQUENCE keywords. If you want to create a sequence in a specific schema, you can specify the schema name in along with the sequence name.

INCREMENT BY

Specify the interval between sequence numbers after the INCREMENT BY keyword.

The interval can have less than 28 digits. It also must be less than MAXVALUE - MINVALUE.

If the interval is positive, the sequence is ascending e.g., 1,2,3,...

If the interval is negative, the sequence is descending e.g., -1, -2, -3 ...

The default value of interval is 1.

START WITH

Specify the first number in the sequence.

The default value of the first number is the minimum value of the sequence for an ascending sequence and maximum value of the sequence for a descending sequence.

MAXVALUE

Specify the maximum value of the sequence.

The `max_value` must be equal to or greater than `first_number` specify after the `START WITH` keywords.

NOMAXVALUE

Use `NOMAXVALUE` to denote a maximum value of 10^{27} for an ascending sequence or -1 for a descending sequence. Oracle uses this option as the default.

MINVALUE

Specify the minimum value of the sequence.

The `min_value` must be less than or equal to the `first_number` and must be less than `max_value`.

NOMINVALUE

Use `NOMINVALUE` to indicate a minimum value of 1 for an ascending sequence or -10^{26} for a descending sequence. This is the default.

CYCLE

Use `CYCLE` to allow the sequence to generate value after it reaches the limit, min value for a descending sequence and max value for an ascending sequence.

When an ascending sequence reaches its maximum value, it generates the minimum value.

On the other hand, when a descending sequence reaches its minimum value, it generates the maximum value.

NOCYCLE

Use `NOCYCLE` if you want the sequence to stop generating the next value when it reaches its limit. This is the default.

CACHE

Specify the number of sequence values that Oracle will preallocate and keep in the memory for faster access.

The minimum of the cache size is 2. The maximum value of the cache size is based on this formula:

```
(CEIL (MAXVALUE - MINVALUE)) / ABS (INCREMENT)
```

Code language: SQL (Structured Query Language) (sql)

In case of a system failure event, you will lose all cached sequence values that have not been used in committed SQL statements.

ORDER

Use ORDER to ensure that Oracle will generate the sequence numbers in order of request.

This option is useful if you are using Oracle Real Application Clusters. When you are using exclusive mode, then Oracle will always generate sequence numbers in order.

NOORDER

Use NOORDER if you do not want to ensure Oracle to generate sequence numbers in order of request. This option is the default.

Oracle CREATE SEQUENCE statement examples

Let's take some example of using sequences.

1) Basic Oracle Sequence example

The following statement creates an ascending sequence called id_seq, starting from 10, incrementing by 10, minimum value 10, maximum value 100. The sequence returns 10 once it reaches 100 because of the CYCLE option.

```
CREATE SEQUENCE id_seq  
  INCREMENT BY 10  
  START WITH 10  
  MINVALUE 10  
  MAXVALUE 100  
  CYCLE  
  CACHE 2;
```

Code language: SQL (Structured Query Language) (sql)

To get the next value of the sequence, you use the NEXTVAL pseudo-column:

```
SELECT
  id_seq.NEXTVAL
FROM
  dual;
```

Code language: SQL (Structured Query Language) (sql)

Here is the output:

```
NEXTVAL
-----
10
```

Code language: SQL (Structured Query Language) (sql)

To get the current value of the sequence, you use the CURRVAL pseudo-column:

```
SELECT
  id_seq.CURRVAL
FROM
  dual;
```

Code language: SQL (Structured Query Language) (sql)

The current value is 10:

```
CURRVAL
-----
10
```

Code language: SQL (Structured Query Language) (sql)

This SELECT statement uses the id_seq.NEXTVAL value repeatedly:

```
SELECT
  id_seq.NEXTVAL
FROM
  dual
CONNECT BY level <= 9;
```

Code language: SQL (Structured Query Language) (sql)

Here is the output:

```
NEXTVAL
-----
20
30
40
50
60
70
80
90
100
```

9 rows selected

Code language: SQL (Structured Query Language) (sql)

Because we set the CYCLE option for the id_seq sequence, the next value of the id_seq will be 10:

```
SELECT id_seq.NEXTVAL FROM dual;
```

Code language: SQL (Structured Query Language) (sql)

And here is the output:

```
NEXTVAL
-----
10
```

Code language: SQL (Structured Query Language) (sql)

2) Using a sequence in a table column example

Prior Oracle 12c, you can associate a sequence indirectly with a table column only at the insert time.

See the following example.

First, [create a new table](#) called tasks:

```
CREATE TABLE tasks(
  id NUMBER PRIMARY KEY,
  title VARCHAR2(255) NOT NULL
);
```

Code language: SQL (Structured Query Language) (sql)

Second, create a sequence for the id column of the tasks table:

```
CREATE SEQUENCE task_id_seq;
```

Code language: SQL (Structured Query Language) (sql)

Third, [insert](#) data into the tasks table:

```
INSERT INTO tasks(id, title)
VALUES(task_id_seq.NEXTVAL, 'Create Sequence in Oracle');
```

```
INSERT INTO tasks(id, title)
VALUES(task_id_seq.NEXTVAL, 'Examine Sequence Values');
```

Code language: SQL (Structured Query Language) (sql)

Finally, [query](#) data from the tasks table:

```
SELECT
  id, title
FROM
  tasks;
```

Code language: SQL (Structured Query Language) (sql)

ID	TITLE
1	Create Sequence in Oracle
2	Examine Sequence Values

In this example, the `tasks` table has no direct association with the `task_id_seq` sequence.

3) Using the sequence via the identity column example

From Oracle 12c, you can associate a sequence with a table column via the [identity column](#).

First, [drop](#) the `tasks` table:

```
DROP TABLE tasks;
```

Code language: SQL (Structured Query Language) (sql)

Second, recreate the `tasks` table using the identity column for the `id` column:

```
CREATE TABLE tasks(  
  id NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,  
  title VARCHAR2(255) NOT NULL  
);
```

Code language: SQL (Structured Query Language) (sql)

Behind the scenes, Oracle creates a sequence that associates with the `id` column of the `tasks` table.

Because Oracle generated the sequence automatically for the `id` column, in your Oracle instance, the name of the sequence may be different.

	❖ COLUMN_NAME	❖ DATA_TYPE	❖ NULLABLE	DATA_DEFAULT	❖ COLUMN_ID	❖ COMMENTS
1	ID	NUMBER	No	"OT"."ISEQ\$\$_74366".nextval	1 (null)	
2	TITLE	VARCHAR2 (255 BYTE)	No	(null)	2 (null)	

Oracle uses the `sys.idnseq$` to store the link between the table and the sequence.

This query returns the association of the `tasks` table and `ISEQ$$_74366` sequence:

```
SELECT  
  a.name AS table_name,  
  b.name AS sequence_name  
FROM  
  sys.idnseq$ c  
  JOIN obj$ a ON c.obj# = a.obj#  
  JOIN obj$ b ON c.seqobj# = b.obj#  
WHERE  
  a.name = 'TASKS';
```

Code language: SQL (Structured Query Language) (sql)

Third, insert some rows into the `tasks` table:

```
INSERT INTO tasks(title)  
VALUES('Learn Oracle identity column in 12c');
```

```
INSERT INTO tasks(title)
VALUES('Verify contents of the tasks table');
Code language: SQL (Structured Query Language) (sql)
```

Finally, query data from the tasks table:

```
SELECT
  id, title
FROM
  tasks;
Code language: SQL (Structured Query Language) (sql)
```

ID	TITLE
1	Learn Oracle identity column in 12c
2	Verify contents of the tasks table

In this tutorial, you have learned how to use the Oracle CREATE SEQUENCE statement to create a new sequence in the database.