SEQUENCE

SEQUENCE

- A Sequence is simply an automatic counter, which generates sequential numbers whenever required.
- The value generated can have a maximum of 38 digits.
- It generates numbers in ascending or descending order.
- Provides intervals between numbers.
- Caching of sequence numbers in memory to speed up their availability.
- It is an independent object and can be used with any table that requires its output.

SEQUENCE

TO CREATE A SEQUENCE:

SYNTAX:

CREATE SEQUENCE <Sequence Name>

[INCREMENT BY <IntegerValue>

START WITH <Integer Value>

MAXVALUE <Integer Value>

MINVALUE <Integer Value>

CYCLE/NOCYCLE

CACHE]

SEQUENCE

INCREMENT BY:

Specifies the interval between sequence numbers. It can be any positive or negative value but not Zero. If this clause is omitted, the default value is 1.

MAXVALUE And MINVALUE:

Specifies the maximum or minimum value that a sequence can generate.

START WITH:

Specifies the first sequence number to be generated. The default for an **ascending** sequence is the sequence **minimum value (1)** and for a **descending** sequence, it is the **maximum value (-1)**.

SEQUENCE

CYCLE:

Specifies that the sequence continues to generate repeat values after reaching either its Maximum value.

NOCYCLE:

No cycle specifies that if sequence exceeds maxvalue an error will be thrown.

CACHE:

Specifies how many values to generate in advance and to keep in memory for faster access. Minimum value is two for this option.

SEQUENCE

```
SQL> CREATE SEQUENCE SEQ_1
  2  INCREMENT BY 1
  3  START WITH 5
  4  MINVALUE 10
  5  MAXVALUE 20
  6  CYCLE;
CREATE SEQUENCE SEQ_1
*
ERROR at line 1:
ORA-04006: START WITH cannot be less than MINVALUE
```

Start value should always be greater than or equal to the minimum value.

```
SQL> CREATE SEQUENCE SEQ_1
  2  INCREMENT BY 1
  3  START WITH 10
  4  MINVALUE 10
  5  MAXVALUE 11
  6  CYCLE;
CREATE SEQUENCE SEQ_1
*
ERROR at line 1:
ORA-04013: number to CACHE must be less than one cycle
```

By default **cache will have 20 values in its memory**. So the max value should start from **21**.

SEQUENCE

```
SQL> CREATE SEQUENCE SEQ_1
  2  INCREMENT BY 1
  3  START WITH 5
  4  MINVALUE 5
  5  MAXVALUE 21
  6  CYCLE
  7  CACHE 5;

Sequence created.
```

TO ACCESS SQUENCES:

Oracle provides two pseudo columns **NEXTVAL** and **CURRVAL** to access the values generated by Sequence.

NEXTVAL – To get the next value from the sequence generated.

CURRVAL – To get the current value from the sequence generated.

SEQUENCE

To view Current value from Sequence:

```
SQL> Select SEQ_1.CURRVAL from DUAL;

   CURRVAL
----------
         5
```

To view Next value from Sequence:

```
SQL> Select SEQ_1.NEXTVAL from DUAL;

   NEXTVAL
----------
         6
```

SEQUENCE

ALTERING A SEQUENCE:

- A sequence once created can be altered.
- It can be done by using the ALTER SEQUENCE statement.

Limitations:

- The minimum value cannot be more than the current value of the sequence.
- The maximum value cannot be less than the current value of the sequence.
- The **START** value of the sequence **CANNOT BE ALTERED**.

SEQUENCE

SYNTAX:

ALTER SEQUENCE <SequenceName>

[INCREMENT BY <IntegerValue>

MINVALUE <IntegerValue>

MAXVALUE <IntegerValue]

EXAMPLE:

```
SQL> ALTER SEQUENCE SEQ_1
  2  INCREMENT BY 2
  3  MAXVALUE 30;

Sequence altered.
```

SEQUENCE

CYCLE IN SEQUENCE:

```
SQL> CREATE SEQUENCE SEQ_2
  2  START WITH 40
  3  INCREMENT BY 3
  4  MAXVALUE 50
  5  CYCLE
  6  CACHE 2;

Sequence created.
```

Note: Once Maximum range is reached, Cycle will generate the sequence of numbers from first. By default it will start from 1.

SEQUENCE

EXAMPLE:

```
SQL> Select SEQ_2.NEXTVAL from DUAL;
   NEXTVAL
----------
        40
SQL> Select SEQ_2.NEXTVAL from DUAL;
   NEXTVAL
----------
        43
SQL> Select SEQ_2.NEXTVAL from DUAL;
   NEXTVAL
----------
        46
SQL> Select SEQ_2.NEXTVAL from DUAL;
   NEXTVAL
----------
        49
SQL> Select SEQ_2.NEXTVAL from DUAL;
   NEXTVAL
----------
         1
SQL> Select SEQ_2.NEXTVAL from DUAL;
   NEXTVAL
----------
         4
```

SEQUENCE

NOCYCLE IN SEQUENCE:

```
SQL> CREATE SEQUENCE SEQ_3
  2  START WITH 40
  3  INCREMENT BY 3
  4  MAXVALUE 50
  5  NOCYCLE
  6  CACHE 2;

Sequence created.
```

Note: Once Maximum range is reached, NOCycle will not generate sequence, instead it will throw error.

SEQUENCE

NOCYCLE IN SEQUENCE:

EXAMPLE:

```
SQL> Select SEQ_3.NEXTVAL from DUAL;

   NEXTVAL
----------
        40

SQL> Select SEQ_3.NEXTVAL from DUAL;

   NEXTVAL
----------
        43

SQL> Select SEQ_3.NEXTVAL from DUAL;

   NEXTVAL
----------
        46

SQL> Select SEQ_3.NEXTVAL from DUAL;

   NEXTVAL
----------
        49

SQL> Select SEQ_3.NEXTVAL from DUAL;
Select SEQ_3.NEXTVAL from DUAL
*
ERROR at line 1:
ORA-08004: sequence SEQ_3.NEXTVAL exceeds MAXVALUE and cannot be instantiated
```

SEQUENCE

TO DROP A SEQUENCE:

SYNTAX:

DROP SEQUENCE SEQUENCE_NAME;

EXAMPLE:

```
SQL> DROP SEQUENCE SEQ_1;

Sequence dropped.
```

SEQUENCE

TO USE SEQUENCE IN RELATION:

```
SQL> CREATE SEQUENCE STUDENT_SEQ
  2  INCREMENT BY 1
  3  START WITH 1
  4  MINVALUE 1
  5  MAXVALUE 100
  6  NOCYCLE
  7  CACHE 5;

Sequence created.
```

INSERTING VALUES FOR THE COLUMN ID USING NEXTVAL:

```
SQL> CREATE TABLE STUDENT_INFO(ID NUMBER, NAME VARCHAR(10));

Table created.

SQL> INSERT INTO STUDENT_INFO VALUES(STUDENT_SEQ.NEXTVAL,'Aanandh');

1 row created.

SQL> INSERT INTO STUDENT_INFO VALUES(STUDENT_SEQ.NEXTVAL,'Aakash');

1 row created.
```

SEQUENCE

INSERTING VALUES FOR THE COLUMN ID USING CURRVAL:

```
SQL> INSERT INTO STUDENT_INFO VALUES(STUDENT_SEQ.CURRVAL,'Chandru');

1 row created.

SQL> INSERT INTO STUDENT_INFO VALUES(STUDENT_SEQ.CURRVAL,'Dinesh');

1 row created.
```

TO DISPLAY STUDENT_INFO:

```
SQL> SELECT *FROM STUDENT_INFO;

        ID NAME
---------- ----------
         1 Aanandh
         2 Aakash
         2 Chandru
         2 Dinesh
```