### Introduction To Sequences ====>
=====================================

* A Sequence is a database object which generates integer sequence.

* We generally use it for populating numeric Primary Key columns.

```
CREATE SEQUENCE sequence_name
    [START WITH start_num]
    [INCREMENT BY increment_num]
    [MAXVALUE maximum_num | NOMAXVALUE]
    [MINVALUE minimum_num | NOMINVALUE]
    [CACHE cache_num | NOCACHE]
    [CYCLE | NOCYCLE]
    [ORDER | NOORDER];
```

* Create Sequence is a DDL which is followed by the name of the sequence which is purely user defined i.e. we can give any name of our choice to our sequence.

* Then we have few attributes of the sequence whose description is given:-

# START WITH ==>
================

* Here we have to specify a numeric value from which we want our sequence to start.

* Whatever number we specify will be the first number generated by our sequence.

* The START WITH clause starts the sequence with the number 1.

# INCREMENT BY ==>
==================

* This attribute also takes a numeric value, to increment the sequence by.

* The number that we specify here will serve as the interval between sequence numbers.

* The value for INCREMENT BY cannot be 0 but it can be any positive or negative value.

* If this value is negative, then the sequence descends. If the value is positive, then the sequence ascends. If you omit this clause,then the interval defaults to 1.

# MAXVALUE / NOMAXVALUE ==>
===========================

* Next attribute is MAXVALUE or NOMAXVALUE.

* Using these attributes we can set the maximum upper bound for our sequence.

* Always remember MAXVALUE must be equal to or greater than START WITH and must be greater than the MINVALUE attribute.

* In case we don’t want to set the MAXVALUE for our sequence then we can use NOMAXVALUE attribute.

# MINVALUE / NOMINVALUE ==>
===========================

* Similar to MAXVALUE we use MINVALUE attribute to set the lower bound of our sequence.

* As a value this attribute also accepts the numeric value and should be less than or equal to START WITH as well as less than MAXVALUE.

* In case we don’t want to set the lower bound for our sequence then we can use NOMINVALUE attribute instead.

# CACHE / NOCACHE ==>
=====================

* As the value of cache attribute, we specify the number of integers to keep in memory.

* The default number of integers to cache is 20. The minimum number of integers that may be cached is 2.

* Specify NOCACHE to indicate that values of the sequence are not pre-allocated.

* If you omit both CACHE and NOCACHE, the database caches 20 sequence numbers by default.

# CYCLE / NOCYCLE ==>
=====================

* CYCLE and NOCYCLE are two flags which we have to set.

* If we set the flag on cycle then our sequence continues to generate values after reaching either its maximum or minimum value.

* We specify NOCYCLE flag when we do not want our sequence to generate more values after reaching its maximum or minimum value.

* If in case we omit both these flags then by default oracle engine will set the flag on NOCYCLE.

# ORDER/NOORDER ==>
===================

* At last we have two more flags which are ORDER and NOORDER.

* ORDER Flag guarantees that sequence numbers are generated in order of request.

* Guaranteeing order is usually not important for sequences that are used to generate primary keys.

* Set the flag on ORDER if we want to guarantee that the sequence numbers are generated in order of request.

* NOORDER is the default flag in case we omit either of them.

## Points To Remember ===>
===========================

* We can specify any of these attributes and flags in any order.

* Means order of these flags is not fixed.

* Another point is that all these attributes and flags are optional.

* If we omit all of them then oracle engine will create a default sequence for us.

# Example ==>
=============

* Create Sequence sq_demo;

Since we have not mentioned any attribute , these will be set to default value:
1. START WITH --> 1
2. INCREMENT BY --> 1
3. MINVALUE --> 1 , for asc seq and -10**26 for desc sequence
4. MAXVALUE --> 10**27 for asc seq and -1 for desc sequence
5. CACHE --> default is 20
6. CYCLE/NOCYCLE --> Default is NOCYCLE
7. ORDER/NOORDER --> Default is NOORDER

## How To Use A Sequence ?

* To use a sequence we use NEXTVAL and CURRVAL.

* Both these are pseudo columns of a sequence using which we can retrieve next value and current value of a sequence.

* NEXTVAL column returns the next value of the sequence as well as initializes the sequence whereas CURRVAL column will return the current value of the sequence.

- select s1.nextval from dual;

```
   NEXTVAL
----------
         1
```

* This query will initialize and return the first value of our newly created sequence.

* To get the current value of our sequence we use CURRVAL pseudo column of a Sequence as shown below-

- select s1.currval from dual;

```
   CURRVAL
----------
         1
```

* Example ==>

```
- create table myprod(pid number(3),
  2  pname varchar2(20),
  3  price number(4));
```

```
- desc myprod;
Name                     Null?    Type
----------------------- -------- ----------------------------
PID                               NUMBER(3)
PNAME                             VARCHAR2(20)
PRICE                             NUMBER(4)
```

* Syntax -

```
-  insert into table_name values(s1.nextval, pname, price);
```

```
select * from myprod;

PID PNAME          PRICE
--- ------------- ----------
  4 Mouse            400
  5 Keyboard         800
  6 Pen              200
  7 Marker           500
```

# Point To Remember ==>

=======================

* Before using any sequence it's mandatory to initialize it first.

* If we will try to retrieve current value without initializing it then it will give us an error.

* We use NEXTVAL pseudo column to initialize a sequence as well as to retrieve next value of the sequence.

* This means after creating a sequence we have to execute the NEXTVAL query before the CURRVAL one.

## Altering A Sequence ===>

=============================

* The ALTER SEQUENCE command allows us to change the properties of a sequence, such as the increment value, min and max values, and cache option.

* The syntax of the ALTER SEQUENCE command is as follows.

```
- ALTER SEQUENCE <seq_name>
    [INCREMENT BY increment_num]
    [MAXVALUE maximum_num | NOMAXVALUE]
    [MINVALUE minimum_num | NOMINVALUE]
    [CACHE cache_num | NOCACHE]
    [CYCLE | NOCYCLE];
```

* Example :- Suppose we want to modify the value of INCREMENT BY attribute from 2 to 4, so for that ALTER SEQUENCE command will be:

```
- ALTER SEQUENCE sq_demo INCREMENT BY 4;
```

# Restrictions On Alter Sequence ==>

=====================================

* Following are some limitations on what we can modify in a sequence :
- We cannot change the start value of a sequence.
- 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.

### Dropping A Sequence ====>
==============================

* To delete any sequence from the schema we use DROP SEQUENCE command as shown below :-

```
#   DROP SEQUENCE sq_demo;
```

## Obtaining Details Of Sequence ==>
=====================================

* In order to obtain details about a sequence we can use Oracle’s internal data dictionary called USER_SEQUENCES.

* It has following useful columns :-

```
- desc user_sequences;
Name                                      Null?    Type
----------------------------------------- -------- -------------
SEQUENCE_NAME                             NOT NULL VARCHAR2(30)
MIN_VALUE                                          NUMBER
MAX_VALUE                                          NUMBER
INCREMENT_BY                              NOT NULL NUMBER
CYCLE_FLAG                                         VARCHAR2(1)
ORDER_FLAG                                         VARCHAR2(1)
CACHE_SIZE                                NOT NULL NUMBER
LAST_NUMBER                               NOT NULL NUMBER
```

## Restrictions On Using Sequence ==>
=====================================

* The NEXTVAL and CURRVAL pseudocolumns can be used in the following SQL constructs :-
- VALUES clause of an INSERT statement.
- SET clause of an UPDATE statement.
- SELECT list (unless it is part of a subquery or view).

* Sequence values are not allowed in the following statements :-
- Subquery of a SELECT, UPDATE, or DELETE statement.
- SELECT statement containing DISTINCT, GROUP BY, ORDER BY, UNION, UNION ALL, INTERSECT, or MINUS.
- WHERE clause of a SELECT statement.
- DEFAULT clause of a column in a CREATE or ALTER TABLE statement.
- CHECK constraint condition.