

SEQUENCE:

- SEQUENCE IS A DB OBJECT.WHICH IS USED TO GENERATE SEQUENCE NUMBERS ON A PARTICULAR COLUMN AUTOMATICALLY.

SYNTAX:

CREATE SEQUENCE <SEQUENCE NAME>

[START WITH N]

[MINVALUE N]

[INCREMENT BY N]

[MAXVALUE N]

[NO CYCLE / CYCLE]

[NO CACHE / CACHE N];

PARAMETERS OF SEQUENCE OBJECT:

START WITH N:

- IT REPRESENT THE STARTING SEQUENCE NUMBER.HERE "N" IS REPRESENT WITHN NUMBER.

MINVALUE N:

- IT SPECIFY THE MINIMUM VALUE OF THE SEQUENCE.HERE "N" IS REPRESENT WITHN NUMBER.

INCREMENT BY N:

- IT SPECIFY THE INCREMENTAL VALUE IN BETWEEN SEQUENCE NUMBERS.HERE "N" IS REPRESENT WITHN NUMBER.

MAXVALUE N:

- IT SPECIFY THE MAXIMUM VALUE OF THE SEQUENCE.HERE "N" IS REPRESENT WITHN NUMBER.

NO CYCLE:

- IT IS DEFAULT PARAMETER. IF WE CREATED SEQUENCE WITH "NO CYCLE" THEN SEQUENCE STARTS FROM START WITH VALUE AND GENERATE VALUES UPTO MAX VALUE. AFTER REACHING MAX VALUE THEN SEQUENCE IS STOP.

CYCLE:

- IF WE CREATED A SEQUENCE WITH "CYCLE" THEN SEQUENCE STARTS FROM START WITH VALUE AND GENERATE VALUES UPTO MAXVALUE. AFTER REACHING MAX VALUE THEN SEQUENCE WILL STARTS WITH MINVALUE.

NO CACHE:

- IT IS DEFAULT PARAMETER. WHEN WE CREATED A SEQUENCE WITH "NO CACHE"

PARAMETER THEN THE SET OF SEQUENCE VALUES ARE STORING INTO DATABASE

MEMORY. EVERY TIME WE WANT ACCESS SEQUENCE NUMBERS THEN ORACLE SERVER WILL GO TO DATABASE MEMORY AND RETURN TO USER. SO THAT IT WILL DEGRADE THE PERFORMANCE OF AN APPLICATION.

CACHE N:

- WHEN WE CREATED A SEQUENCE WITH "CACHE" PARAMETER THEN SYSTEM IS ALLOCATING TEMP. MEMORY (CACHE) AND IN THIS MEMORY WE WILL STORE THE SET SEQUENCE NUMBERS. WHENEVER USER WANT TO ACCESS SEQUENCE NUMBERS THEN ORACLE SERVER WILL GO TO CACHE MEMORY AND RETURN TO USER.

- ACCESSING DATA FROM CACHE IS MUCH FASTER THAN ACCESSING DATA FROM DATABASE. IT WILL INCREASE THE PERFORMANCE OF AN APPLICATION. HERE "N" IS REPRESENT THE SIZE OF CACHE FILE. MINIMUM SIZE OF CACHE IS 2KB AND MAXIMUM SIZE OF CACHE IS 20KB.

NOTE:

- TO WORK WITH SEQUENCE OBJECT WE SHOULD USE THE FOLLOWING TWO PSEUDO COLUMNS ARE "NEXTVAL" AND "CURRVAL".

NEXTVAL:

- IT IS USED TO GENERATE SEQUENCE NUMBERS ON A PARTICULAR COLUMN.

SYNTAX:

SELECT <SEQUENCE NAME>. <NEXTVAL> FROM DUAL;

CURRVAL:

- IT IS USED TO SHOW THE CURRENT VALUE OF THE SEQUENCE.

SYNTAX:

SELECT <SEQUENCE NAME>. <CURRVAL> FROM DUAL;

EX1:

STEP1:

SQL> CREATE SEQUENCE SQ1

START WITH 1

MINVALUE 1

INCREMENT BY 1

MAXVALUE 3;

SEQUENCE CREATED.

STEP2:

SQL> CREATE TABLE TEST1(SNO INT, NAME VARCHAR2(10));

TABLE CREATED.

TESTING:

=====

SQL> INSERT INTO TEST1 VALUES (SQ1.NEXTVAL,'&NAME');

ENTER VALUE FOR NAME: A

/

ENTER VALUE FOR NAME: B

/

ENTER VALUE FOR NAME: C

/

ENTER VALUE FOR NAME: D

ERROR AT LINE 1:

**ORA-08004: SEQUENCE SQ1.NEXTVAL EXCEEDS MAXVALUE AND
CANNOT BE INSTANTIATED.**

ALTERING A SEQUENCE:

SYNTAX:

ALTER SEQUENCE <SEQUENCE NAME> <PARAMETER NAME> N;

EX:

SQL> ALTER SEQUENCE SQ1 MAXVALUE 5;

SEQUENCE ALTERED.

TESTING:

SQL> INSERT INTO TEST1 VALUES (SQ1.NEXTVAL,'&NAME');

ENTER VALUE FOR NAME: D

/

ENTER VALUE FOR NAME: E

OUTPUT:

SQL> SELECT * FROM TEST1;

SNO	NAME
-----	-----
1	A
2	B
3	C
4	D
5	E

NOTE: WE CAN ALTER ALL PARAMETERS EXCEPT "START WITH " PARAMETER.

EX2:

SQL> CREATE SEQUENCE SQ2

START WITH 1

MINVALUE 1

INCREMENT BY 1

MAXVALUE 3

CYCLE

CACHE 2;

SEQUENCE CREATED.

SQL> CREATE TABLE TEST2(SNO INT, NAME VARCHAR2(10));

TABLE CREATED.

TESTING:

SQL> INSERT INTO TEST2 VALUES (SQ2.NEXTVAL, '&NAME');

ENTER VALUE FOR NAME: A

/

ENTER VALUE FOR NAME: B

/

ENTER VALUE FOR NAME: C

/

.....
.....
OUTPUT:

SQL> SELECT * FROM TEST2;

SNO	NAME
-----	-----
1	A
2	B
3	C
1	D
2	E
3	F

EX3:

SQL> CREATE SEQUENCE SQ3

START WITH 3

MINVALUE 1

INCREMENT BY 1

MAXVALUE 5

CYCLE

CACHE 2;

SEQUENCE CREATED.

SQL> CREATE TABLE TEST3(SNO INT, NAME VARCHAR2(10));

TABLE CREATED.

TESTING:

SQL> INSERT INTO TEST3 VALUES (SQ3.NEXTVAL, '&NAME');

ENTER VALUE FOR NAME: A

/

.....

.....

OUTPUT:

SQL> SELECT * FROM TEST3;

SNO	NAME
-----	-----
3	A
4	B
5	C
1	M
2	N
3	O
4	P
5	Q

NOTE: IF WE WANT TO VIEW ALL SEQUENCES IN ORACLE DATABASE THEN WE USE

"USER_SEQUENCES" DATA DICTIONARY.

EX:

SQL> DESC USER_SEQUENCES;

SQL> SELECT SEQUENCE_NAME FROM USER_SEQUENCES;

SYNTAX TO DROP A SEQUENCE:

=====

SQL> DROP SEQUENCE <SEQUENCE NAME>;

EX:

SQL> DROP SEQUENCE SQ1;