

This chapter introduces you to the concept of sequence, creation of sequence, viewing the sequence, and dropping them.

Introduction

A sequence is a software function that generates integer numbers in either ascending or descending order, within a definite range, to generate primary key and coordinate other keys among the table. You use sequence for availing integer numbers say, for employee_id or transaction_id. A sequence can support SMALLINT, BIGINT, INTEGER, and DECIMAL data types. A sequence can be shared among multiple applications. A sequence is incremented or decremented irrespective of transactions.

A sequence is created by CREATE SEQUENCE statement.

Types of Sequences

There are two type of sequences available:

1. **NEXTVAL**: It returns an incremented value for a sequence number.
2. **PREVIOUS VALUE**: It returns recently generated value

Parameters of sequences

The following parameters are used for sequences:

Data type: This is the data type of the returned incremented value.

SMALLINT, BIGINT, INTEGER, NUMBER, DOUBLE

START WITH: The reference value, with which the sequence starts.

MINVALUE: A minimum value for a sequence to start with.

MAXVALUE: A maximum value for a sequence.

INCREMENT BY: step value by which a sequence is incremented.

Sequence cycling: the **CYCLE** clause causes generation of the sequence repeatedly. The sequence generation is conducted by referring the returned value, which is stored into the database by previous sequence generation.

Creating a sequence

You can create sequence using the following syntax:

Syntax:

```
db2 create sequence <seq_name>
```

Example: [To create a new sequence with the name 'sales1_seq' and increasing values from 1]

```
db2 create sequence sales1_seq as int start  
with 1 increment by 1
```

Viewing the sequences

You can view a sequence using the syntax given below:

Syntax:

```
db2 value <previous/next> value for <seq_name>
```

Example: [To see list of previous updated value in sequence 'sales1_seq']

```
db2 values previous value for sales1_seq
```

Output:

```
1
-----
4
1 record(s) selected.
```

Dropping the sequence

To remove the sequence, you need to use the "DROP SEQUENCE " command. Here is how you do it:

Syntax:

```
db2 drop sequence <seq_name>>
```

Example: [To drop sequence 'sales1_seq' from database]

```
db2 drop sequence sales1_seq
```

Output:

```
DB20000I The SQL command completed successfully.
Loading [Mathjax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js
```