

SQL | SEQUENCES

SQL sequences are an essential feature of relational database management systems (RDBMS) used to generate unique numeric values in a sequential order. These values are widely used for generating primary keys, unique keys, and other numeric identifiers in databases. SQL sequences offer flexibility, performance, and ease of use, making them indispensable in managing and organizing data in large-scale applications.

In this article, we will explain SQL sequences in-depth, explaining how they work, their advantages, syntax, and real-world use cases.

What Are SQL Sequences?

SQL sequences are user-defined database objects designed to generate a series of numeric values. Unlike identity columns, which are tightly bound to specific tables, sequences are independent objects and can be used across multiple tables. They allow applications to retrieve the next number in a sequence whenever needed, offering a simple and efficient way to generate unique numbers on demand.

The values in a sequence can be configured to be generated in ascending or descending order, and the sequence can be set to restart (cycle) once the maximum value is exceeded. This makes SQL sequences particularly useful in scenarios where there is a need for continuous, unique values, such as generating primary keys or serial numbers.

Key Features of SQL Sequences

- **Automatic Primary Key Generation:** Sequences automatically generate unique values that can be used for primary or unique keys in database tables.
- **Ascending or Descending Order:** Sequences can be configured to generate numbers in either ascending or descending order.
- **Multiple Table Usage:** A single sequence can be used to generate values for multiple tables, making it flexible and reusable.
- **Independent of Tables:** Unlike identity columns, sequences are independent and can be used across different tables.
- **Efficient:** Sequences reduce the complexity and overhead of manually generating unique values, which saves time and reduces application code.

How SQL Sequences Work

When creating a sequence, we specify the starting point, the increment (how much the sequence increases with each step), and optionally the minimum and maximum values. Sequences can be set to cycle, which means they restart from the beginning when they reach the maximum value.

Example 1: Creating a Sequence in Ascending Order



```
CREATE SEQUENCE sequence_1
start with 1
increment by 1
minvalue 0
maxvalue 100
cycle;
```

Example 2: Creating a Sequence in Descending Order




```
CREATE SEQUENCE sequence_2
start with 100
increment by -1
min value 1
max value 100
cycle;
```

What's New in SEQUENCE – Oracle 19c

1. Scalable Sequences (Scalability Improvement)


- Oracle 19c introduced Scalable Sequences to improve performance in highly concurrent environments.
- These sequences reduce contention and allow parallel inserts to run faster without waiting for the same sequence.



```
CREATE SEQUENCE scalable_seq  
START WITH 1  
INCREMENT BY 1  
SCALE;
```

2. Session Sequences (also introduced in 18c, still relevant)

You can create private session-level sequences, but note this feature was introduced in 18c and continued in



```
CREATE PRIVATE TEMPORARY TABLE t1 (  
  id NUMBER DEFAULT SESSION.SEQ.NEXTVAL  
);
```