

SEQUENCE

Purpose

`CREATE SEQUENCE` creates a database object that generates unique integers, often used for primary keys. Sequences are independent of tables and accessible by multiple users. Each `NEXTVAL` call increments the sequence, regardless of transaction success, which may lead to gaps. Values are unique per user and remain accessible after generation. Use `CURRVAL` for the current value and `NEXTVAL` to get the next.

Prerequisites

To create a sequence in your own schema, you must have the `CREATE SEQUENCE` system privilege.

To create a sequence in another user's schema, you must have the `CREATE ANY SEQUENCE` system privilege.

Key Features of SQL Sequences

- **Auto Key Generation:** Automatically creates unique values for primary or unique keys.
- **Custom Order:** Supports ascending or descending numbering.
- **Reusable:** One sequence can serve multiple tables.
- **Table-Independent:** Works separately from table definitions.
- **Efficient:** Simplifies unique value generation and reduces code overhead.

How They Work

A sequence starts from a defined value and increases by a set increment. You can specify min/max limits and enable cycling to restart after reaching the max.

Create Sequence

```
CREATE SEQUENCE user_id_seq START WITH 1000
```

```
INCREMENT BY 1
```

```
MINVALUE 1000
```

```
MAXVALUE 9999
```

```
CYCLE
```

```
CACHE 20;
```

CREATE SEQUENCE user_id_seq	Creates a new sequence named user_id_seq
START WITH 1000	The sequence starts at 1000 (first value generated).
INCREMENT BY 1	Each new value increases by 1. You can use -1 for descending.
MINVALUE 1000	The lowest number the sequence can generate
MAXVALUE 9999	The highest number allowed in this sequence.
CYCLE / NOCYCLE	Restarts from MINVALUE after reaching MAXVALUE. Without it, the sequence will stop
CACHE 20 / NOCACHE	Pre-generates 20 values in memory for better performance. You can use NOCACHE to disable this

Use the Sequence

```
INSERT INTO users (id, name) VALUES (user_id_seq.NEXTVAL , 'Ahmed');
```

Use CURRVAL to get the **current value** (after a NEXTVAL call in the same session):

```
SELECT user_id_seq.CURRVAL FROM dual;
```