PostgreSQL与Oracle一样， 都有单独的序列， 而不像MySQL中的自增键是绑定在一张表的字段上的。 MySQL数据库中的自增键有如下限制：

1.自增键只能用于表中的其中一个字段。

2.自增键只能被分配给固定表的固定的某一个字段， 不能被多个表共用。

PostgreSQL的序列是基于bigint运算的， 其范围不能超过8字节的整数范围（-9223372036854775808到223372036854775807）

**序列的使用**

**语法格式：**

CREATE [ TEMPORARY | TEMP ] SEQUENCE name [ INCREMENT [ BY ] increment ]

[ MINVALUE minvalue | NO MINVALUE ] [ MAXVALUE maxvalue | NO MAXVALUE ]

[ START [ WITH ] start ] [ CACHE cache ] [ [ NO ] CYCLE ]

[ OWNED BY { table.column | NONE } ]

**参数说明：**

TEMPORARY或TEMP： 是否创建临时序列， 会话结束序列会自动删除。

INCREMENT [BY] increment： 指定序列的步长， 正数为递增序列， 负数为递减序列。 默认值是1。

MINVALUE minvalue|NO MINVALUE： 指定序列的最小值， 对于递增序列， 最小值默认为1； 对于递减序列， 最小值默认为-2^63-1。 NO MINVALUE相当于使用默认值。

MAXVALUE maxvalue|NO MAXVALUE： 指定序列的最大值， 对于递增序列， 最大值默认为2^63-1； 对于递减序列， 最大值默认为-1。 NO MAXVALUE相当于使用默认值。

START [WITH] start： 指定开始的起点值。

[CACHE cache]： 指定cache的数值。 最小值（也是默认值） 是1， 表示一次只能生成一个值， 也就是说没有缓存。

[NO] CYCLE： CYCLE选项可用于使序列到达maxvalue或minvalue时循环并继续下去。 也就是说， 如果达到极限， 生成的下一个数据将分别是minvalue或maxvalue。 如果声明了NO CYCLE， 那么在序列达到其最大值后， 任何对nextval的调用都将返回一个错误。 如果既没有声明CYCLE也没有声明NO CYCLE， 那么默认是NO CYCLE。

OWNED BY {table.column|NONE}： OWNED BY选项将序列关联到一个特定的表字段上。 这样， 在删除该字段或其所在的表时将自动删除绑定的序列。 指定的表和序列必须被同一个用户所拥有， 并且在同一个模式中。 默认的OWNED BY NONE表示无关联。

**序列管理**

cache的用法

create sequence seq1 cache 10;

select nextval('seq1');--1

select nextval('seq1');--2

select nextval('seq1');--11，新开一个会话

所有在会话中分配了却没有使用的数字都将在会话结束时丢失， 从而导致序列里面出现“空洞”，很多时候表中生成的序列不是递增的。

CYCLE的使用

没有加CYCLE选项的序列示例如下：

CREATE SEQUENCE seq2 MINVALUE 1 MAXVALUE 4;

select nextval('seq2');--1

select nextval('seq2');--2

select nextval('seq2');--3

select nextval('seq2');--4

select nextval('seq2');--ERROR

加CYCLE选项的序列示例如下：

CREATE SEQUENCE seq2 MINVALUE 1 MAXVALUE 4 CYCLE;

select nextval('seq2');--1

select nextval('seq2');--2

select nextval('seq2');--3

select nextval('seq2');--4

select nextval('seq2');--1

**重置序列**

ALTER SEQUENCE seq1 RESTART WITH 100;

**删除序列**

DROP SEQUENCE seq2;

**序列函数**

|  |  |
| --- | --- |
| **函数** | **描述** |
| nextval ( regclass ) → bigint | 递增序列并返回最新的值 |
| setval ( regclass, bigint [, boolean ] ) → bigint | 设置序列的当前值，is\_called为可选项项 |
| currval ( regclass ) → bigint | 返回最近一次用 nextval 获取的指定序列的值 |
| lastval () → bigint | 返回最近一次用 nextval 获取的任何序列的值 |

CREATE SEQUENCE seq1;

SELECT nextval('seq1');--1

SELECT currval('seq1');--1

SELECT currval('seq1');--新开一个session执行报错

SELECT nextval('seq1');--1

select lastval();--1

SELECT nextval('seq1');--2

select lastval();--2 ,新开一个session执行不会报错

select setval('seq1', 1);

SELECT nextval('seq1');--2

select setval('seq1', 1, false);

select nextval('seq1');--1

select nextval('seq1');--2

注意：为了避免阻塞，nextval操作不会回滚，也就是说，在事务中使用序列， 当事务回滚后， 序列不会回滚。

begin;

select nextval('seq1');--3

rollback;

select currval('seq1');--3