# 概述

在数据库中，ID往往作为数据表的主键。ID的创建规则又往往使用自增的整数。

## MySQL/SQL Server

在SQL Server和MySQL中提供了自增的字段类型，但是Oracle中并未提供该用法。

## PostgreSQL

PostgreSQL 使用序列来标识字段的自增长，数据类型有 smallserial、serial 和 bigserial。

MySQL与PG：

1、mysql在字段上加AUTO\_INCREMENT属性来实现自增，pg使用serial类型，序列号类型其实不是真正的类型，当声明一个字段为序列号类型时其实是创建了一个序列。

2、mysql插入的是字段的值，pg执行插入的时候变化的是字段值和序列的值，只不过在缺省插入的时候二者相等。

所以，INSERT时如果没有给该字段赋值会默认获取下一个值或者对应序列的下一个值。

但是在该字段有赋值插入之后，就有区别了，pg赋值插入的时候是改变了字段的值但是没有改变序列的值，所以再次缺省赋值插入时依旧是顺序获取对应序列的下一个值。

3、mysql和pg在delete之后插入是同样的，但是truncate操作在pg上是清空了表数据，但是没有清空对应的序列，所以在执行truncate之后，再次插入会获取者对应序列的下一个值，而在mysql上是从1开始。

4、达到规定最大值之后都不能继续插入，所以不能循环使用。

## Oracle

在oracle中sequence就是序号，是用来在多用户环境下产生唯一整数的数据库对象。每次取的时候它会自动增加，sequence与表没有关系。

序列产生器顺序生成数字，它可用于自动生成主键值，并能协调多行或者多表的主键操作。没有sequence，顺序的值只能靠编程来生成，先找出最近产生的值然后自增长。

这种方法要求一个事务级别的锁，这将导致在多用户并发操作的环境下，必须有人等待下一个主键值的产生，而且此方法很容易产生主键冲突的错误。

# 功能

Sequence的功能就是产生分布式系统单调变化的序列，为了分布式系统也能像单机系统一样产生可靠的全局单调序列，因此选择系统中控制全局的网元承担数据库表的创建、维护和消耗功能，但是不维护sequence元数据。

其功能主要体现在：

1. 作为代理主键，唯一识别；
2. 用于记录数据库最新动作的语句，只要语句有动作（增删改），sequence号就会随着更新，所以我们可以根据sequence号来select出更新的语句。

# 操作

序列（SEQUENCE）像其他数据库对象（表、约束、视图、触发器等）一样，是实实在在的数据库对象。一旦创建，即可存在于数据库中，并可在适用场合进行调用。序列总是从指定整数开始，并按照特定步长进行累加，以获得新的整数。

## 创建序列

创建序列，应该使用create sequence命令。

## 使用序列

对于序列，有两个重要的属性——currval和nextval。其中currval用于获得序列的当前值，而nextval则用于获得序列的下一个值。每次调用nextval，都会使序列的当前值增加单位步长（默认步长为1）。

序列的调用方法为seq.currval和seq.nextval。但是，需要注意的是，在序列创建之后，应该首先使用seq.nextval，然后才能够使用seq.currval。

## 序列初始值start with

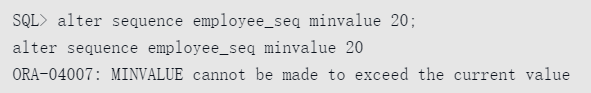
为了解决employee\_seq的nextval属性不适合表t\_employee的问题。可以首先删除序列，然后重建该序列，并在重建时指定初始值。

## 修改序列属性

像其他数据库对象一样，可以通过alter命令修改序列属性。可修改的属性包括minvalue、maxvalue、increment\_by、cycle和cache。

### 修改minvalue和maxvalue

minvalue和maxvalue用于指定序列的最小值和最大值。序列最小值的意义在于限定start with和循环取值时的起始值；而最大值则用于限制序列所能达到的最大值。序列最小值不能大于序列的当前值。例如，尝试将序列employee\_seq的最小值设置为20，Oracle将会抛出错误提示。



### 修改increment by

increment by相当于编程语言for循环中的步长。即每次使用nextval时，在当前值累加该步长来获得新值。序列的默认步长为1，可以通过alter命令和increment by选项来修改序列步长。

### 修改cycle

cycle选项用于指定序列在获得最大值的下一个值时，从头开始获取。这里的“头”即为minvalue指定的值。为了说明cycle的功能及start with与minvalue的区别，首先创建新的序列，并为各选项指定特定值。

### 修改cache

顾名思义，cache是序列缓存，其实际意义为，每次利用nextval，并非直接操作序列，而是一次性获取多个值的列表到缓存。使用nextval获得的值，实际是从缓存抓取。抓取的值，依赖于序列的currval和步长increment by。

默认缓存的大小为20，可以通过alter命令修改缓存大小。

# 分布式数据库实践

## GoldenDB

在分布式数据库GoldenDB中，采用sequence实现全局唯一ID（自增列）的功能，统一由GTM（全局事务管理模块）负责，可以保证全局唯一，但是不能保证严格递增。

### 创建

create sequence [schema.]sequence

[increment by num]

[start with num]

[maxvalue num | nomaxvalue]

[minvalue num | nominvalue]

[cycle | nocycle]

[cache| nocache]

说明：

INCREMENT BY用于定义序列的步长，如果省略，则默认为1，如果出现负值，则代表序列的值是按照此步长递减的。

START WITH定义序列的初始值（即产生的第一值），默认为1。

MAXVALUE定义序列生成器能产生的最大值。选项NOMAXVALUE是默认选项，代表没有最大值定义，这时对于递增序列，系统能够产生的最大值是10的27次方，对于递减序列，最大值是-1。

MINVALUE定义序列生成器能够产生的最小值。选项NOMINVALUE是默认选项，代表没有最小值定义，这时对于递减序列，系统能够产生的最小值是-10的26次方，对于递增序列，最大值为1。

CYCLE和NOCYCLE表示当序列生成器的值达到限制后是否循环。CYCLE代表循环，NOCYCLE代表不循环。如果循环，则当递增序列达到最大值时，循环到最小值；对于递减序列达到最小值时，循环到最大值。如果不循环，达到限制后，继续产生新值就会发生错误。

CACHE（缓存）定义存放序列的内存块的大小。NOCACHE表示不对序列进行内存缓冲。对序列进行内存缓冲，可以改善序列的性能。

### 修改

alter sequence [schema.]sequence increment by num

### 删除

drop sequence [schema.]sequence

### 限制

1. sequence默认cache为100
2. Sequence前缀含义与oracle不一致，oracle是用户概念，GoldenDB是类似db概念
3. Sequence目前不支持ORDER|NOORDER属性
4. Sequence修改步长时会导致当前cache丢失
5. Sequence默认最大值与最小值与oracle不一致
6. Sequence负的最小值应该比理论int64最小值大1

## TDSQL

## OceanBase

## TiDB