# 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。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。

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

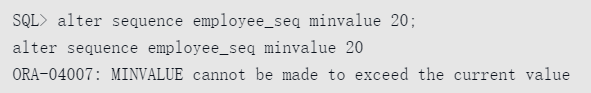
为了解决employee\_seq的nextval属性不适合表t\_employee的问题。可以首先删除序列，然后重建该序列，并在重建时指定初始值。

## 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

在分布式数据中，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。