

RE: Best Way to Auto Increment with Oracle

Source: <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2004-07/0050.html>

From: Ron Reidy (Ron.Reidy_at_arraybiopharma.com)

Date: 07/06/04

Date: Tue, 6 Jul 2004 10:13:26 -0600

To: "Helck, Timothy" <Timothy.Helck@bowker.com>, "Jared Still" <jkstill@cybcon.com>, "Kipp, James"

Yes, I said that, but ... I was mistaken (so sorry and thanks to Ron Kimball for pointing this out to me). According to the Metalink note 108643.1,

Once a sequence number is generated, the sequence number is available only to the session that generated the number.

Ron Reidy
Senior DBA
Array BioPharma, Inc.

-----Original Message-----

From: Helck, Timothy [<mailto:Timothy.Helck@bowker.com>]

Sent: Tuesday, July 06, 2004 8:34 AM

To: Reidy, Ron; Jared Still; Kipp, James

Cc: Jim; DBI List

Subject: RE: Best Way to Auto Increment with Oracle

-- OK, RETURNING is good, I can see where I might use that.

-- Are you saying that currval cannot be relied upon to be the same as nextval without locking? My understanding is that it can be (though how it works is beyond me).

-----Original Message-----

From: Reidy, Ron [<mailto:Ron.Reidy@arraybiopharma.com>]

Sent: Tuesday, July 06, 2004 10:06 AM

To: Helck, Timothy; Jared Still; Kipp, James

Cc: Jim; DBI List

Subject: RE: Best Way to Auto Increment with Oracle

Comments below ...

Ron Reidy
Senior DBA
Array BioPharma, Inc.

perl.dbi.users: RE: Best Way to Auto Increment with Oracle

-----Original Message-----

From: Helck, Timothy [mailto:Timothy.Helck@bowker.com]
Sent: Tuesday, July 06, 2004 7:52 AM
To: Jared Still; Kipp, James
Cc: Reidy, Ron; Jim; DBI List
Subject: RE: Best Way to Auto Increment with Oracle

It's not always convenient to use triggers to assign the primary key. Sometimes we want to know the primary key of a new record so we can use it to insert rows into a dependent table.

[rr] Look at using the 'RETURNING' clause.

Am I right to think that the following is nearly as efficient as a trigger-based approach?

```
insert into customer(customer_id, first, last)
values(customer_id_seq.nextval, 'Homer', 'Simpson');
```

```
insert into address(address_id, customer_id, street, town)
values(address_id_seq.nextval, customer_id_seq.currval, '742 Evergreen Tr.', 'Springfield');
```

[rr] This will not work. What happens when there are multiple processes running an insert into customer at the same time. Sequences are not transactional.

Or is there a better way to do this?

Tim Helck

-----Original Message-----

From: Jared Still [mailto:jkstill@cybcon.com]
Sent: Monday, July 05, 2004 9:45 PM
To: Kipp, James
Cc: 'Reidy, Ron'; Jim; DBI List
Subject: RE: Best Way to Auto Increment with Oracle

On Wed, 2004-06-16 at 08:43, Kipp, James wrote:

```
> >
> > Your program connects via SQL*net to the DB. At insert time,
> > you need to fetch a sequence, so you issue 'select
> > seq.nextval from dual'. Seems harmless enough until ...
> >
> > 1. The select statement must be passed across the SQL*Net connection.
> > 2. The statement must be parsed. Maybe the worst you will
> > have here is a soft parse, but non the less, you are using
> > the CPU for this.
> > 3. The statement is executed and the fetch occurs.
> > 4. The result is passed back to your program.
> > 5. You bind the value (hopefully) to your statement handle.
> > 6. You then execute the insert statement.
> >
> > If you use a BEFORE INSERT trigger to fetch the sequence, you
> > will eliminate steps 1, 2, 3, 4, 5 from the flow.
> >
```

RE: Best Way to Auto Increment with Oracle

>
> *Great point. I think I will stay with the Before insert trigger.*
>
>

I've always liked the elegance of inserting primary key values from a before insert trigger via a sequence. I still do it sometimes, though now I prefer not to.

When done from a trigger, you must use the 'select seq.nextval from dual' statement to get the PK value.

This is a **major** scalability killer. Easy to prove if anyone is interested.

Or just google for it, it has been discussed quite heavily on at least one oracle list.

Jared

This electronic message transmission is a PRIVATE communication which contains information which may be confidential or privileged. The information is intended to be for the use of the individual or entity named above. If you are not the intended recipient, please be aware that any disclosure, copying, distribution or use of the contents of this information is prohibited. Please notify the sender of the delivery error by replying to this message, or notify us by telephone (877-633-2436, ext. 0), and then delete it from your system.

This electronic message transmission is a PRIVATE communication which contains information which may be confidential or privileged. The information is intended to be for the use of the individual or entity named above. If you are not the intended recipient, please be aware that any disclosure, copying, distribution or use of the contents of this information is prohibited. Please notify the sender of the delivery error by replying to this message, or notify us by telephone (877-633-2436, ext. 0), and then delete it from your system.