

## RE: Best Way to Auto Increment with Oracle

**Source:** <http://coding.derkeiler.com/Archive/Perl/perl.dbi.users/2004-06/0147.html>

---

**From:** Ron Reidy (*Ron.Reidy\_at\_arraybiopharma.com*)

**Date:** 06/16/04

Date: Wed, 16 Jun 2004 10:30:17 -0600

To: "David Goodman" <dtzgdman@yahoo.com>, <dbi-users@perl.org>

Set event 10046 and trace the execution.

I am reasonably sure any overhead associated with a trigger will not be as expensive as client side SQL\*Net round trips to get sequences and the associated SQL\*Net wait events while the client is waiting for the server, or while the server is waiting for the client.

-----  
Ron Reidy  
Senior DBA  
Array BioPharma, Inc.

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

From: David Goodman [mailto:dtzgdman@yahoo.com]

Sent: Wednesday, June 16, 2004 9:59 AM

To: dbi-users@perl.org

Subject: Re: Best Way to Auto Increment with Oracle

Doesn't an Oracle before insert trigger carry extra performance overhead?

--- "Peter J. Holzer" <hjp@wsr.ac.at> wrote:

> On 2004-06-16 09:14:25 -0600, Reidy, Ron wrote:

> > Better? Maybe. Consider.

> >

> > 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.  
>  
> You can get the next value from a sequence in the  
> insert statement (see  
> my previous mail in this thread), so with or without  
> the trigger, you  
> have to parse (once) and execute (possibly many  
> times) only the insert  
> statement.  
>  
> The difference is:  
>  
> With a trigger, you hide the details of the sequence  
> from the insert  
> statement.  
> This makes the insert statement simpler and more  
> portable. OTOH, you  
> don't know which value you just inserted (you can  
> get at it with "select  
> \$sequence.currval from dual" but then you lose these  
> advantages).  
>  
> hp  
>  
> --  
> \_/ Peter J. Holzer | Shooting the users in  
> the foot is bad.  
> |\_) | Sysadmin WSR / LUGA | Giving them a gun  
> isn't.  
> ||| hjp@wsr.ac.at | -- Gordon Schumacher,  
> \_/ | <http://www.hjp.at/> | mozilla bug  
> #84128  
>  
> ATTACHMENT part 2 application/pgp-signature

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.