

Re: MATREADU/MATWRITE statements timing out

Source: <http://newsgroups.derkeiler.com/Archive/Comp/comp.databases.pick/2007-11/msg00181.html>

- *From:* "Chandru Murthi" <cmurth_xyz@xxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Tue, 27 Nov 2007 01:47:05 GMT
-

Ross, I suspect you and I would be in complete agreement. If I understand you right, Visage allows pessimistic locking but also give you other options. And from past posts, I know that Visage is far from being stateless.

The issue in this thread is twofold. Locking of transaction records (as defined in the classical green screen terms) has to be pessimistic or else all hell ensues. Anyone who has had to deal with a pissed-off (rightly so) user whose updates (be it for 1 minute or 60) are rejected because of optimistic locking is either willfully misrepresenting the issue or has a much thicker skin than I.

Abstract the issue. If you logged into your favorite web site and started buying products, only to be told at the end that something was wrong, you would never go that site again. So it's to do with fulfilling user expectations. If you need to build an app that, eventually, updates tens of files in some related fashion, but for reasons of scale, cannot afford to lock all the records, you will find a way to do so without causing the user grief. If nothing else, that's what back ordering is for.

Chandru Murthi

"Ross Ferris" <rossf@xxxxxxxxxxxxxxxx> wrote in message news:ee5c011e-d576-4a34-a8cd-f3ddaf917b36@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

On Nov 26, 10:45 am, "Peter McMurray" <excalibu...@xxxxxxxxxxxxx> wrote:

Hi

This is exactly what I am getting at and what things like continuation servers are trying to address. There are a couple of issues here, I'll deal with the simple one first. It is now my opinion that fast moving items such as the balance that are only ever incremented or decremented should be held in a separate file to the static information such as credit rating, name etc.

However the issue of changing the static information is a much more

Re: MATREADU/MATWRITE statements timing out

important problem. If I take a typical debtor master file I can be looking at say 50 attributes. It is my understanding of what you and others have said that you expect to do the following

- 1 Send a copy of the record to the user
- 2 the user makes alterations
- 3 the user then sends 2 copies back, the original and the alterations.
- 4 the server program reads and locks the record
- 5 the server program compares the original from the user to the file master that is now locked. If it is equal the server updates it otherwise it rejects the changes.

Obviously this can cause some considerable angst to the user and dramatically increase the load on an already overloaded communication link.

However it gets to be a much bigger problem when more than one file is involved as the complete set has to be held and checked before release. Surely the minor occasional inconvenience of a temporarily pessimistically locked record is preferable.

I really would appreciate expansion of this.

Regards

Peter McMurray "Bill H" <some...@xxxxxxxxxxxxxxxx> wrote in message

[news: fmdnbc8K6YnadTanZ2dnUVZ_h2pnZ2d@xxxxxxxxxxxxxxxx](mailto:fmdnbc8K6YnadTanZ2dnUVZ_h2pnZ2d@xxxxxxxxxxxxxxxx)

Peter:

I wouldn't say your're wrong. The locking model you've described (pessimistic locking at the read level) works wonderfully but not in many circumstances using modern technology.

Web connections are stateless and require a replacement of this model.

There's nothing we can do about this. What's required is that the database manage the transaction updating in a batch (transaction) and the

Re: MATREADU/MATWRITE statements timing out

transaction has to be concerned about what's important.

For instance, let's assume we have a customer record that contains the following:

001 Customer Name
002 Customer Address (MVA)
003 Outstanding Invoices
004 Invoice Balance
.
.

In our telnet paradigm, we could read and lock the record, make whatever changes we wanted, update everything, then update the customer record.

All was well. But in a stateless environment, there is no simple locking scheme. Consequently, we have to understand that when we read the record, only attribute#s 001 & 002 are important. Thus, when we call the update program from the web page's [Submit] button, we need to verify the name and address haven't changed since our read. However, the outstanding invoices and invoice balance, from the read, can be ignored and we can accept whatever is read and locked during the update.

This at least overcomes us trying to build locking into our application, with all the attendant complexities and problems.

Bill

Re: MATREADU/MATWRITE statements timing out

"Peter McMurray" <excalibu...@xxxxxxxxxxxx> wrote in message
news:b152j.16674\$CN4.13593@xx

Hi
Joe has made a weird statement that has no
basis in fact. My
statement
however stands as a proven daily event it is
not based on something
you
may or may not have mentioned. If a party is
changing a master file
the
method of communication is irrelevant.
Please have another look at
my
answer and demonstrate where I am wrong.

Peter McMurray
"Kevin Powick" <kpow...@xxxxxxxxxx>
wrote in message
news:7b610a34-8f14-4360-bb27-c57bc5e66dad@xx

On Nov 24, 5:00 pm,
"Excalibur"
<excalibu...@xxxxxxxxxxxx>
wrote:

"Kevin
Powick"
<kpow...@xxxxxxxxxx>
wrote in
message

I
won't
go
into
all
the
details,
which
inevitably
lead
to
religious
debates

Re: MATREADU/MATWRITE statements timing out

(Been
there.
Done
that)

Hi
Anything
can be done
but some
things
should
never be
done. This
is
one
of
them.

My original quote stands. I
won't be drawn into what
ultimately ends
up as a "religious" debate.
And your comments are
based on details
that you have imagined, as I
have not provided them.

Also, I forgot to mention
stateless clients, such as web
browsers,
but
Joe has done that in his
reply.

--

Kevin Powick– Hide quoted
text –

– Show quoted text –

Peter,

Re: MATREADU/MATWRITE statements timing out

Re: MATREADU/MATWRITE statements timing out

General rules are :

Pessimistic locks for transactions (required for transaction boundaries anyway)

Optimistic locks for "masterfile updates" (which 'deteriorate' to field by field comparisons during actual update, when a Pessimistic locks is exerted whilst the actual update/comparison is done)

Whilst on comparisons, the 3rd alternative is that whilst your OLD record # CURRENT record, your NEW field may = CURRENT field, in which case all is AOK for 'update' (which doesn't need to take place for that field)

In Visage we have an option for a manual resolution process that displays all 3 values, but to date I'm not aware of anyone in the REAL WORLD ever having seen it (how likely is it that multiple people would change the same master file information to DIFFERENT values ... and if they do, then we HAVE to defer to a person to resolve, 'cause there are just toooooo many variables and not a fact to be found to guide us)

Multi-valued field (for notes) will typically either need some serious work to identify changes OR may be reworked so that notes are treated as unique "transactions"

HTH

(Did have a nice BIG response, but accidentally hit "discard" --> that will teach me to formulate in word first)