

Re: dataadapter.Update() multiple table update strategies

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.framework.adonet/2006-11/msg00073.html>

- *From:* "Earl" <brikshoe@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 2 Nov 2006 00:50:50 -0500
-

The problem is not with the dataset, it's with how you are submitting your updates. I am battling a very similar issue, as it appears that Microsoft generally does not support a database that has structures more complex than parent-child, so many-to-many or other complex db scenarios have to be hand-crafted. Most books I've read just blow right past the subject, although I've heard that Sahil Malik's "Pro ADO" book discusses it. Sceppa's book discusses ADO.Net transactions but other than a brief mention there's not much on HOW to submit a massive update to a transaction sproc.

One thing, you'll never accomplish what you are trying to do with a join. Each table has to be handled separately, even if you are going to submit the updates as a transaction. Something I'm playing around with is to create all the individual table parameters with the DACW and then stitch them together for the transaction (you can dump the visual adapters after you copy out the params).

"jarb" <no@xxxxxxxx> wrote in message [news:unb2h.209\\$3K7.53@xxxxxxxxxxxxxxxxxxxx](mailto:news:unb2h.209$3K7.53@xxxxxxxxxxxxxxxxxxxx)

Im not sure how to proceed on this one...

I have a dataset with one table being populated by a complex join from several tables. Basically a fancy SELECT statement. Designer doesn't support generating a dataset from a query like this (right?) so I hand created the dataset and the SELECT statement. Now dataadapter.Fill() works fine. Its time to get Update() working. Since I need to put values back into multiple tables I need to do several SQL UPDATES. I tried crafting a BEGIN END; block with all the needed SQL (im using Oracle). But this doesn't return an affected row count so a concurrency exception gets thrown. Seems like I have the wrong approach.

Am I supposed to not do SQL joins, but instead pull my whole database schema, or at least all the necessary tables, locally into the dataset and then work with it using the dataset paradigm, relations, and constraints? I really don't like this. I don't want an in-memory wanna-be database. I just want a lightweight cache to save network bandwidth and allow for form control binding.

Re: dataadapter.Update() multiple table update strategies

Or should I create multiple data adapters for each SQL table UPDATE I need to do? Or just create multiple update command objects and then assign them to the dataadapter and Update() the dataset one command at a time? But im going to need them all to be part of a transaction.

Maybe the simplest thing would just be to create yet another view in Oracle that makes everything look like a single table. Too bad the db developer is swamped with work.

TIA