

Re: Query training -- Complex queries

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.server/2004-02/1123.html>

From: Erland Sommarskog (*sommar_at_algonet.se*)

Date: 02/25/04

Date: Wed, 25 Feb 2004 23:39:33 +0000 (UTC)

DW (None) writes:

- > *Can anyone please suggest a book (or a training class) that gets into*
- > *the guts of complex queries? I need to understand, for example, the*
- > *internal differences between this update query:*
- >
- > *Update PositionsEOM*
- > *Set MktVal = PM.Price * PositionsEOM.Quantity_ASD*
- > *From PositionsEOM Inner Join [Price Master] As PM On*
- > *PositionsEOM.Fund_ID = PM.Fund_ID And*
- > *PositionsEOM.Position_Date = PM.Price_Date*
- >
- > *and this one:*
- >
- > *Update PositionsEOM*
- > *Set MktVal = Derived.MktVal From*
- > *(Select PM.Price * PositionsEOM.Quantity_ASD As MktVal*
- > *From PositionsEOM Inner Join [Price Master] As PM On*
- > *PositionsEOM.Fund_ID = PM.Fund_ID And*
- > *PositionsEOM.Position_Date = PM.Price_Date) As Derived*
- >
- > *The second example ran for 20 minutes and I cancelled it; the first one*
- > *finished in 2.5 minutes.*

As Adam pointed out, the problem with the second query is that the derived table is not correlated with the updated table, so you get a magnificent cross join, and I'm not surprise that it did not complete in 20 minutes.

Since I don't know your tables, I cannot really suggest a better way to use the derived table. What you could write, is this:

```
Update PositionsEOM
Set MktVal = PositionsEOM.Quantity_ASD *
  (Select PM.Price
   FROM [Price Master] As PM
   WHERE PositionsEOM.Fund_ID = PM.Fund_ID And
         PositionsEOM.Position_Date = PM.Price_Date)
```

Here you don't have a derived table, but a correlated subquery. This is actually the way you should do it, if you stick to ANSI. I need to confess, though, that I prefer the syntax with FROM over the using a correlated subquery.

> *You wouldn't KNOW that you could write "Set" expressions like the first example, from ANY of the online documentation.*

Nah, I would say that this is quite clear from the syntax diagram for UPDATE in Books Online.

> *Also, I decided that when you mix inner and outer joins in a select, which I do sometimes, if you change the order, the result set WILL be different (not just the time it takes). (I think this is true...)*

Yes, this is correct. As long as you are only using inner joins, you can have them in any order, but with left joins the table are turn. Consider:

```
SELECT ....  
FROM a  
LEFT JOIN b ON a.col = b.col  
JOIN c ON b.col2 = c.col2
```

What you probably want is:

```
SELECT  
FROM a  
LEFT JOIN (b JOIN c ON b.col2 = c.col2) ON a.col = b.col
```

In the first case, you take all rows in a, and if there is no matching rows in b, you set all b:s columns to NULL. But then join the table (a LEFT JOIN b) with c, and you will include rows that matches in c, so all those NULL rows in b are now dropped.

In the second case you first join b with c, and then join this table with a.

> *I found one online article that mentioned that if you mix inner and outer joins in a query, you CAN control the order of the joins -- and it shows various ways to do that, including parentheses and derived or virtual tables. But only ONE online article. It's VERY rare to see any of this documented, and SQL Books Online sure doesn't show examples.....*

It should be in the ANSI-92 standard for SQL, but it's hardly the kind of book you read first.

The best book for reading about the various join techniques, is Joe Celko's "SQL For Smarties", but of course it has the drawback that it is not written for SQL Server. Tom Moreau's and Itzik Ben-Gan's book

microsoft.public.sqlserver.server: Re: Query training -- Complex queries

is a better choice, if you want something which is more on target. And quite some of these subjects are covered in Kalen Delaney's "Inside SQL Server 2000". It was from her SQL7 book, that I learnt to use paranthesis with JOIN.

--

Erland Sommarskog, SQL Server MVP, sommar@algonet.se
Books Online for SQL Server SP3 at
<http://www.microsoft.com/sql/techinfo/productdoc/2000/books.asp>