

Re: SQLXML for xml explicit

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.xml/2005-10/msg00091.html>

- *From:* "Toby" <Toby@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 24 Oct 2005 22:25:01 -0700
-

Thanks michael and eugene. Yes i would add indexes to the table.

i'd be given a course id..
from that i'd be required to generate a xml for that course
containing lesson details..

then from those lessonids i need to form another xml containing topic ids.

then from those topic ids i need to form another xml containing page details..

So as you see that would be a lot of XML say 20 or more ..

as you can see it would result in executing the complex 'FOR XML EXPLICIT'
everytime i need to generate the XML (20 times).
once this is done the process gets over.

If I am going to use the sql server 2000's 'FOR XML EXPLICIT' it'd take some
time.

So Is this the best way to go for such a requirement. OR should the
developers (Java :() develop a code to generate their own XML.

or Is there any other efficient way to do. (oh no dont say 2005 we wont be
able to upgrade at this point of time)..

waiting eagerly for ur invaluable suggestions,
toby

"Michael Rys [MSFT]" wrote:

- > In addition to Eugene's recommendation, I would also like to point out what
- > performance issues you have to compare against the alternative of doing the
- > XML-ification on the client:

Re: SQLXML for xml explicit

>
> Unless some of the XML that you need is static (such so-called wrapper
> elements such as the root element or the orders element containing all order
> elements), doing the joins and unions on the server normally is better since
> it avoids multiple client-server roundtrips, joins can be optimized by the
> server optimizer and less data may be transported between the server and the
> client.
>
> Best regards
> Michael
>
> "Eugene Kogan [MSFT]" <ekogan@xxxxxxxxxxxxxxxxxxxxxx> wrote in message
> <news:uNnOPYQ2FHA.3596@xxxxxxxxxxxxxxxxxxxxxxxxxxxx
> > > To make a FOR XML EXPLICIT query to perform the best you could create
> > enough indexes on the tables so that ORDER BY does not result in a SORT in
> > the query plan and the indexes are used by Query Processor to implement
> > UNION ALLs as MERGE UNIONS.
> > As you can see this is not a FOR XML specific advice and the general
> > indexing trade-off between query performance, DML performance and data
> > size should be applied.
> >
> > Best regards,
> > Eugene
> > ---
> >> SELECT
> >> 1 AS tag,
> >> 0 AS parent,
> >> y.coursetype AS [X!1!type],
> >> x.coursetitle AS [X!1!coursetitle],
> >> x.courseid AS [X!1!courseid],
> >> y.curriculumid AS [X!1!curriculumid],
> >> y.coursehelptext AS [X!1!helptext],
> >> y.mastery AS [X!1!mastery],
> >> null AS [Y!2!type],
> >> null AS [Y!2!name],
> >> null AS [Y!2!helptext],
> >> null AS [Z!3!type],
> >> null AS [Z!3!name],
> >> null AS [Z!3!location],
> >> null AS [Z!3!active],
> >> null AS [Z!3!duration],
> >> null AS [Z!3!objective],
> >> null AS [Y!2!lessonid!hide],
> >> null AS [Z!3!topicid!hide]
> >> FROM
> >> X x,
> >> Y y
> >>
> >> WHERE
> >> x.courseid=y.courseid
> >>

```
>>> UNION ALL
>>>
>>> SELECT
>>> 2,
>>> 1,
>>> y.coursetype,
>>> x.coursetitle ,
>>> x.courseid,
>>> y.curriculumid,
>>> y.coursehelptext,
>>> y.mastery,
>>> z.lesontype,
>>> z.lessonname ,
>>> z.lessonhelptext,
>>> null,
>>> null,
>>> null,
>>> null,
>>> null,
>>> null,
>>> z.lessonid,
>>> null
>>>
>>> FROM
>>> X x,
>>> Y y,
>>> Z z
>>>
>>> WHERE
>>> x.courseid = y.courseid AND
>>> z.courseid = x.courseid
>>>
>>>
>>> UNION ALL
>>>
>>> SELECT
>>> 3,
>>> 2,
>>> y.coursetype,
>>> x.coursetitle ,
>>> x.courseid,
>>> y.curriculumid,
>>> y.coursehelptext,
>>> y.mastery,
>>> z.lesontype ,
>>> z.lessonname ,
>>> z.lessonhelptext ,
>>> a.topicquiztype,
>>> a.topicquizname,
>>> a.quizlocation,
>>> a.active,
```

```
>>> a.topicquizobjective,  
>>> a.topicquizduration,  
>>> z.lessonid,  
>>> a.topicquizid  
>>>  
>>>  
>>> FROM  
>>> X x,  
>>> Y y,  
>>> Z z,  
>>> A a  
>>>  
>>> WHERE  
>>> x.courseid = y.courseid AND  
>>> z.courseid = x.courseid AND  
>>> a.courseid = x.courseid AND  
>>> a.lessonid = z.lessonid  
>>>  
>>>  
>>> ORDER BY [X!1!courseid],[Y!node!2!lessonid!hide],[Z!3!topicid!hide]  
>>>  
>>> FOR XML EXPLICIT  
>>>  
>>> waiting eagerly ( scratching my head),  
>>> toby
```

• **Follow-Ups:**

- ◆ **[Re: SQLXML for xml explicit](#)**
◇ From: Michael Rys [MSFT]

• **References:**

- ◆ **[SQLXML for xml explicit](#)**
◇ From: Toby
- ◆ **[Re: SQLXML for xml explicit](#)**
◇ From: Eugene Kogan [MSFT]
- ◆ **[Re: SQLXML for xml explicit](#)**
◇ From: Michael Rys [MSFT]

- Prev by Date: **[Re: Import XML in SQL Server 2000](#)**
- Next by Date: **[Re: Schema for Bulk Load of sub-elements in XML File](#)**
- Previous by thread: **[Re: SQLXML for xml explicit](#)**
- Next by thread: **[Re: SQLXML for xml explicit](#)**
- Index(es):
 - ◆ **[Date](#)**
 - ◆ **[Thread](#)**