

Re: Filtering DB Call based on local Data

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2007-12/msg02434.html>

- *From:* "LVP" <lvp_agentman@xxxxxxxxxxx>
 - *Date:* Wed, 19 Dec 2007 09:38:44 -0800
-

depending on the speed you are looking for,

Send the Ids as XML, parse it in a SP into a temp table then Join and return your result set.

or depending on the number of Ids you can choose one method or another method

How long does it take to put 10K into a temp table via many means? then Join

LVP

"Nicholas Paldino [.NET/C# MVP]" <mvp@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:e5bosC1QIHA.4684@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Bill,

Not really. You should create some sort of dynamic query, using an IN statement, appending a new parameter for each of the items you want to filter on. SQL Server will take care of the rest.

I think that SQL Server has a limitation on the number of parameters that you can use on dynamic queries (around 2048 I think) so you might exceed that limitation.

If you are using SQL Server 2008, you can use table-valued parameters, but I doubt you are doing that.

If you are using SQL Server 2005, you can create a table-valued function in SQL Server which will take a comma-delimited string and return a table of ids, on which you can do a join (if the list of ids is unique) or perform an IN operation on. The one I wrote is a CLR function and is quite fast, even parsing 10,000 items (it beats out any other method that you can code with T-SQL).

Re: Filtering DB Call based on local Data

Then, all you have to do is take your delimited string, pass it as a single parameter to your query (it can be a stored procedure or dynamic query at this point) and you should be done.

—
– Nicholas Paldino [.NET/C# MVP]
– mvp@xx

"Bill" <wbpelen@xxxxxxxx> wrote in message
news:03dfb4ec-cbbe-49b0-8a6b-49b45eched0f@xx

This may be a simple problem (let's hope). I am trying to extract a filtered set of records from a Sql Server Database. The set of rows, however, is to be filtered based on data external to the database.

Specifically, this is an Excel Add-in.

The user selects a range which contains order numbers (single column). Loading that data into a DataTable or Array is easy enough.

Using those (say 30) order_id values, I then want to get data specific to the order from the order table on a Sql Server Database (assume 1:1 relationship).

I am new to ADO, I do not know how best to represent this data (which objects) within the code.

If I use a DataSet where the short list of order ids is in one DataTable with another table connected to the Sql Server order table, will I end up loading all 300K of the orders before I get a chance to filter the set.

Is there an easier way to select the data from the database using different design options?

Ultimately it breaks down to a query like this:
Can I somehow relate data located only in memory/.Net Objects to a table within the database?

The query could be written as below but the possibility is that the

Re: Filtering DB Call based on local Data

list of orders could be as much as 10K.

```
SELECT a, b, c FROM orders WHERE order_id IN ( 100, 200, 300, ..., 999 )
```

Is there a way to do this without building that query?