

## Re: Getting info from SQL DB

---

*Source:*

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.vb/2008-04/msg01592.html>

---

- *From:* "Steve Gerrard" <[mynamehere@xxxxxxxxxxx](mailto:mynamehere@xxxxxxxxxxx)>
  - *Date:* Fri, 25 Apr 2008 21:41:19 -0700
- 

Bill Schanks wrote:

The application pulls users that are in an Active Directory Group. And allows users to pull add'l data from the HR Records that is not avail from Active Directory.

And it doesn't allow 30,000 records in the in clause it errors out. I don't know what the limit is. Now 30,000 is an extreme example. Users most likely won't want to pull info on 30,000 People. But I'd like the app to work regardless.

Got it. So they aren't in the DB. I don't know what the limit in on an IN clause might be, but I suspect it is variable, depending on configuration and possibly load on the server as well.

I would pick a number for the limit of one query, somewhere in the 20 to 50 range maybe. You can then run the query repeatedly, getting 20 to 50 results each time. If you made the query with parameters, you can get pretty good results this way; after the first one, the server caches the statement, and can crank them out. It is a little bit like using a keyset cursor, something some of the database drivers do – or used to do. It will help if PreferredID is indexed.

The SQL with just 3 parameters might look like  
... Where PreferredID IN (@P1, @P2, @P3) ...

you would add the parameters to your command:  
command.Parameters.Add("@P1", SqlDbType.VarChar, 80)  
command.Parameters.Add("@P2", SqlDbType.VarChar, 80)  
command.Parameters.Add("@P3", SqlDbType.VarChar, 80)

for each batch, you set the parameter values instead of building a string:  
command.Parameters("@P1").Value = item.Text

and clear any unused parameters if the last batch has less than a full set.

It would take some fiddling to get it to work right, but once you have it,

## Re: Getting info from SQL DB

something like that could handle any volume, and give a reasonable response time as well.

The alternative, uploading the list of items to the server first, may or may not be useful. Inserting several 1000 records is not particularly fast, but it might work well, depending on – well, lots of things, I guess.