

Re: SQL TOP 50,000 Help

Source:

<http://www.tech-archive.net/Archive/Access/microsoft.public.access.formscoding/2007-02/msg00673.html>

- *From:* Bob Hairgrove <invalid@xxxxxxxxxxx>
 - *Date:* Sat, 10 Feb 2007 14:26:45 +0100
-

On Fri, 9 Feb 2007 12:51:01 -0800, Klatuu
<Klatuu@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote:

I like your hashing algorithm, it is going in my "bag of tricks", but unless I am too thick to catch your point, I don't see how it applies to this particular question.

A lot of the time, people use AutoNumber as their primary key, and the Mod function works well enough with that. However, there are times when it isn't easy to use it, e.g. when multiple columns are involved in the primary key or unique index, when the PK is textual, or when the column used to generate the hash number contains many equal values. This is similar to the problems which arise with TOP (as you pointed out).

To work around that problem, it sometimes helps to store a random number in an extra column and run the hash (or Mod) function over that to ensure equal distribution of values. The random number should be an integer if you want use Mod because that function is only meaningful for integers. The integer range should then be 0 to Count(<table rows>)-1. Of course, there might be a few duplicate numbers generated, but for this purpose only the statistical distribution is meaningful.

Also, if the table doesn't already contain a column of type AutoNumber, one can be added. Access will automatically populate it for existing records, and it is very quick. There is also a way to generate a sequential number using Recordset.AbsolutePosition, but that is usually way too slow.

In general, it is often better (i.e. more precise, but not necessarily faster) to open a recordset in code and iterate over N rows than to use TOP, perhaps creating a temporary table from the result set and running additional queries on that. That way, the duplicate values don't really matter.

Re: SQL TOP 50,000 Help

Bob Hairgrove
NoSpamPlease@xxxxxxxxx