Clustered Indexes Revisited

Posted At : June 3, 2008 11:15 AM | Posted By : Mark Kruger
Related Categories: MS SQL Server

In my last post I talked about using a clustered index on some column other than the primary key. There are cases where this makes sense and it can have positive impact on performance. Recently however, CF Webtools own **Jason Troy** pointed out a consequence of altering your clustered index. You may recall that a "clustered" index really means that the actual data in the table will stored in the sort order specified by the index. Consider this example of a table called "emailer":

ID (PK)	Name	Email
1	tom	bob@oldgraymare.org
2	richard	richard@johnjacob.net
3	harry	harry@jinkleheimerscmidt.biz
4	Sally	sally@monalisagrins.net

If the primary key is "ID" and no changes are made then the sort order follows the PK (1,2,3,4). But we want to use this table to select users by their email address. It might make sense to make the "email" column the sort order of the table instead of the "ID" column. If we do this, any selects where the "WHERE" clause specifies the email address are going to be faster. That's because, being a clustered index, the database doesn't have to "look up" where the data is located in the table. Instead of a card catalog where it first has to find the author, and then finds the section and row, a clustered index works more like a dictionary where the DB can predict exactly where to find the data it's looking for. After the new clustered index is in place, my table is stored like this:

ID (PK)	Name	Email
1	tom	bob@oldgraymare.org
3	harry	harry@jinkleheimerscmidt.biz
2	richard	richard@johnjacob.net
4	Sally	sally@monalisagrins.net

Now that is exactly what we want - right? Well it certainly *could* be, based on performance. But there is a "gotcha". Sometimes we lazy programmers write queries where we *assume* the sort order of the table. Suppose we wanted to find all the emails from the table that match a certain domain.

```
<cfquery name="getEmails" datasource="#myDsn#">
        SELECT     Email
        FROM    Emailer
        WHERE    domain = 'gunsAndNoses.com'
    </cfquery>
```

Before we changed the clustered indexing this query would return the emails in the order that they were entered sequentially (as in 6,8,20,3003,5384) just like our first example above. Now, however, the values will be returned in alphabetical order according to the "email" column. This can have unexpected consequences. For example, if you have a search tool displaying emails and the users are used to seeing

them in sequential order, they may assume that some values are "younger" than other values (our emailer table is probably not a good example). In other words, because you have not specified a sort order in the query, merely changing indexes will alter the view to the end user. Obviously you want indexing changes to the DB to affect performance *only* and to have no other impact that you cannot manage or predict. For this reason it is a good idea to always specify a sort order of some kind in your queries.