

## Clustered indexes on a Non Primary-Key (MS SQL 2000)

Posted At : July 21, 2005 11:11 AM | Posted By : Mark Kruger

Related Categories: SQL tips, MS SQL Server

Ever wonder why a table is only allowed one "clustered" index? It's because the clustered index is the actual sort order that of the table. When you implement a clustered index you are "rearranging" the rows so that they are actually ordered by that index. That makes this index faster. Choosing the right clustered index can have a major impact on the speed of searches in your table. Wait a minute - isn't the Primary Key always the clustered index? Well... yes, when you use enterprise manager to set a primary key you will find that it also designates that column or columns as the default clustered index. It doesn't have to be that way however.

There are times when you may want some other field or field to be clustered. If you look closely at your queries you will see where it might make sense. For example, if you have a table where a date field is often queried as a range as in this query:

```
<cfquery name="getLeads" datasource="#dsn#">
SELECT    D.lead_id, D.User_id, D.co_id,
D.Firstname, D.Lastname, D.Email,
    E.DateSent AS DateAdded, E.subject, E.emailType,
    W.dls, IsNull(A.dls,0) AS Attempts

FROM
myLeads D LEFT JOIN myEmailLog DE
ON D.lead_id = DE.lead_id
RIGHT JOIN myEmailMsg E
ON E.email_id = DE.email_id
    LEFT JOIN logSummary W
ON D.Lead_id = w.lead_id
LEFT JOIN vwDlAttempts A
ON D.Lead_id = A.Lead_id
WHERE    D.User_id = ?
AND      e.emailtype = 'download'
AND      D.co_id = ?
AND      (E.DateSent BETWEEN ? AND ?)
</cfquery>
```

If *DateSent* is the most often used field in a select a clustered index may perform better than a regular index. The problem is that the primary key index is already designated as clustered. If you try and make *DateSent* the clustered index the server will complain that only 1 clustered index is allowed on the table. Fortunately you can work around that.

You see Enterprise Manager only sets the Primary Key as a clustered index if no other clustered indexes are found. Otherwise it's just a constraint. So, using enterprise manager, first remove the primary key - or you can drop it in query analyzer if you like. Then go to the index wizard and create a clustered index on *DateSent*. Finally (and don't forget this). Go back and reinstate your Primary Key. There you have it - a clustered index on a non primary key field.

Remember, knowledge is power - use it wisely. You have to choose a clustered index carefully and it should be based on your knowledge of how the tables and table relationships are used in your application. Sometimes you can add a clustered index and solve 1 problem, but create several others. For example, if you clustered index

isn't "incremental" (like a "DateSent" field) and you do a lot of inserts updates or deletes on that particular field your database will have to work a lot harder to keep the table sorted. For example, if you cluster "lastName" and add my name, Mark Kruger, the database will need to create a row in the middle of the table and all the other records will have to "move down" or "move up".