**Heaps (Tables without Clustered Indexes)**

**SQL Server 2014**

[Other Versions](javascript:;)

http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png?v=635533092500576233

* [SQL Server 2012](http://msdn.microsoft.com/en-us/library/hh213609(d=printer,v=sql.110).aspx)

A heap is a table without a clustered index. One or more nonclustered indexes can be created on tables stored as a heap. Data is stored in the heap without specifying an order. Usually data is initially stored in the order in which is the rows are inserted into the table, but the Database Engine can move data around in the heap to store the rows efficiently; so the data order cannot be predicted. To guarantee the order of rows returned from a heap, you must use the ORDER BY clause. To specify the order for storage of the rows, create a clustered index on the table, so that the table is not a heap.

|  |
| --- |
| **NoteNote** |
| There are sometimes good reasons to leave a table as a heap instead of creating a clustered index, but using heaps effectively is an advanced skill. Most tables should have a carefully chosen clustered index unless a good reason exists for leaving the table as a heap. |

[**When to Use a Heap**](javascript:void(0))

If a table is a heap and does not have any nonclustered indexes, then the entire table must be examined (a table scan) to find any row. This can be acceptable when the table is tiny, such as a list of the 12 regional offices of a company.

When a table is stored as a heap, individual rows are identified by reference to a row identifier (RID) consisting of the file number, data page number, and slot on the page. The row id is a small and efficient structure. Sometimes data architects use heaps when data is always accessed through nonclustered indexes and the RID is smaller than a clustered index key.

[**When Not to Use a Heap**](javascript:void(0))

Do not use a heap when the data is frequently returned in a sorted order. A clustered index on the sorting column could avoid the sorting operation.

Do not use a heap when the data is frequently grouped together. Data must be sorted before it is grouped, and a clustered index on the sorting column could avoid the sorting operation.

Do not use a heap when ranges of data are frequently queried from the table. A clustered index on the range column will avoid sorting the entire heap.

Do not use a heap when there are no nonclustered indexes and the table is large. In a heap, all rows of the heap must be read to find any row.

[**Managing Heaps**](javascript:void(0))

To create a heap, create a table without a clustered index. If a table already has a clustered index, drop the clustered index to return the table to a heap.

To remove a heap, create a clustered index on the heap.

To rebuild a heap to reclaim wasted space, create a clustered index on the heap, and then drop that clustered index.

|  |
| --- |
| **Caution noteCaution** |
| Creating or dropping clustered indexes requires rewriting the entire table. If the table has nonclustered indexes, all the nonclustered indexes must all be recreated whenever the clustered index is changed. Therefore, changing from a heap to a clustered index structure or back can take a lot of time and require disk space for reordering data in tempdb. |