**Clustered Index Structures**

**SQL Server 2008 R2**

[Other Versions](javascript:;)

http://i.technet.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [SQL Server 2008](http://technet.microsoft.com/en-us/library/ms177443(d=printer,v=sql.100).aspx)
* [SQL Server 2005](http://technet.microsoft.com/en-us/library/ms177443(d=printer,v=sql.90).aspx)

In SQL Server, indexes are organized as B-trees. Each page in an index B-tree is called an index node. The top node of the B-tree is called the root node. The bottom level of nodes in the index is called the leaf nodes. Any index levels between the root and the leaf nodes are collectively known as intermediate levels. In a clustered index, the leaf nodes contain the data pages of the underlying table. The root and intermediate level nodes contain index pages holding index rows. Each index row contains a key value and a pointer to either an intermediate level page in the B-tree, or a data row in the leaf level of the index. The pages in each level of the index are linked in a doubly-linked list.

Clustered indexes have one row in [sys.partitions](http://technet.microsoft.com/en-us/library/ms175012(v=sql.105).aspx), with **index\_id** = 1 for each partition used by the index. By default, a clustered index has a single partition. When a clustered index has multiple partitions, each partition has a B-tree structure that contains the data for that specific partition. For example, if a clustered index has four partitions, there are four B-tree structures; one in each partition.

Depending on the data types in the clustered index, each clustered index structure will have one or more allocation units in which to store and manage the data for a specific partition. At a minimum, each clustered index will have one IN\_ROW\_DATA allocation unit per partition. The clustered index will also have one LOB\_DATA allocation unit per partition if it contains large object (LOB) columns. It will also have one ROW\_OVERFLOW\_DATA allocation unit per partition if it contains variable length columns that exceed the 8,060 byte row size limit. For more information about allocation units, see [Table and Index Organization](http://technet.microsoft.com/en-us/library/ms189051(v=sql.105).aspx).

The pages in the data chain and the rows in them are ordered on the value of the clustered index key. All inserts are made at the point where the key value in the inserted row fits in the ordering sequence among existing rows. The page collections for the B-tree are anchored by page pointers in the **sys.system\_internals\_allocation\_units** system view.

|  |
| --- |
| **Important noteImportant** |
| The **sys.system\_internals\_allocation\_units** system view is reserved for Microsoft SQL Server internal use only. Future compatibility is not guaranteed. |

For a clustered index, the **root\_page** column in **sys.system\_internals\_allocation\_units** points to the top of the clustered index for a specific partition. SQL Server moves down the index to find the row corresponding to a clustered index key. To find a range of keys, SQL Server moves through the index to find the starting key value in the range and then scans through the data pages using the previous or next pointers. To find the first page in the chain of data pages, SQL Server follows the leftmost pointers from the root node of the index.

This illustration shows the structure of a clustered index in a single partition.

