Understanding Pages and Extents

SQL Server 2008 R2

The fundamental unit of data storage in SQL Server is the page. The disk space allocated to a data file (.mdf or .ndf) in a database is logically divided into pages numbered contiguously from 0 to *n*. Disk I/O operations are performed at the page level. That is, SQL Server reads or writes whole data pages.

Extents are a collection of eight physically contiguous pages and are used to efficiently manage the pages. All pages are stored in extents.

Pages

In SQL Server, the page size is 8 KB. This means SQL Server databases have 128 pages per megabyte. Each page begins with a 96-byte header that is used to store system information about the page. This information includes the page number, page type, the amount of free space on the page, and the allocation unit ID of the object that owns the page.

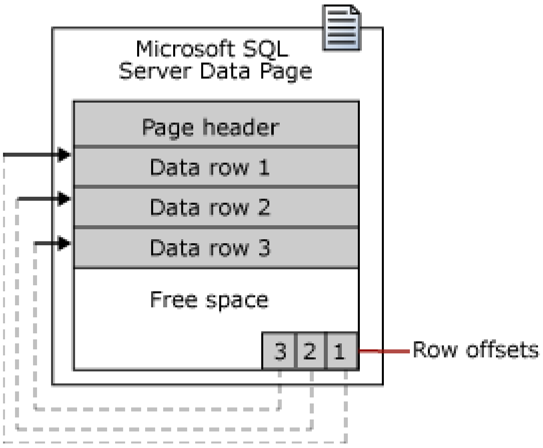
The following table shows the page types used in the data files of a SQL Server database.

Page type	Contents
Data	Data rows with all data, except **text**, **ntext**, **image**, **nvarchar(max)**, **varchar(max)**, **varbinary(max)**, and **xml** data, when **text in row** is set to ON.
Index	Index entries.
Text/Image	Large object data types: • **text** , **ntext**, **image**, **nvarchar(max)**, **varchar(max)**, **varbinary(max)**, and **xml** data Variable length columns when the data row exceeds 8 KB: • **varchar** , **nvarchar**, **varbinary**, and **sql_variant**
Global Allocation Map, Shared Global Allocation Map	Information about whether extents are allocated.
Page Free Space	Information about page allocation and free space available on pages.
Index Allocation Map	Information about extents used by a table or index per allocation unit.
Bulk Changed Map	Information about extents modified by bulk operations since the last BACKUP LOG statement per allocation unit.
Differential Changed Map	Information about extents that have changed since the last BACKUP DATABASE statement per allocation unit.

Note

Log files do not contain pages; they contain a series of log records.

Data rows are put on the page serially, starting immediately after the header. A row offset table starts at the end of the page, and each row offset table contains one entry for each row on the page. Each entry records how far the first byte of the row is from the start of the page. The entries in the row offset table are in reverse sequence from the sequence of the rows on the page.



Large Row Support

Rows cannot span pages, however portions of the row may be moved off the row's page so that the row can actually be very large. The maximum amount of data and overhead that is contained in a single row on a page is 8,060 bytes (8 KB). However, this does not include the data stored in the Text/Image page type. This restriction is relaxed for tables that contain **varchar**, **nvarchar**, **varbinary**, or **sql_variant** columns. When the total row size of all fixed and variable columns in a table exceeds the 8,060 byte limitation, SQL Server dynamically moves one or more variable length columns to pages in the ROW_OVERFLOW_DATA allocation unit, starting with the column with the largest width. This is done whenever an insert or update operation increases the total size of the row beyond the 8060 byte limit. When a column is moved to a page in the ROW_OVERFLOW_DATA allocation unit, a 24-byte pointer on the original page in the IN_ROW_DATA allocation unit is maintained. If a subsequent operation reduces the row size, SQL Server dynamically moves the columns back to the original data page. For more information, see Row-Overflow Data Exceeding 8 KB[1].

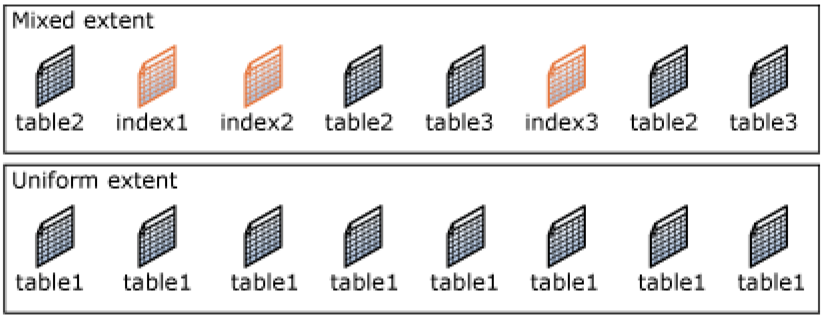
Extents

Extents are the basic unit in which space is managed. An extent is eight physically contiguous pages, or 64 KB. This means SQL Server databases have 16 extents per megabyte.

To make its space allocation efficient, SQL Server does not allocate whole extents to tables with small amounts of data. SQL Server has two types of extents:

- Uniform extents are owned by a single object; all eight pages in the extent can only be used by the owning object.
- Mixed extents are shared by up to eight objects. Each of the eight pages in the extent can be owned by a different object.

A new table or index is generally allocated pages from mixed extents. When the table or index grows to the point that it has eight pages, it then switches to use uniform extents for subsequent allocations. If you create an index on an existing table that has enough rows to generate eight pages in the index, all allocations to the index are in uniform extents.



See Also

Concepts

Files and Filegroups Architecture[2]

Other Resources

Tables and Index Data Structures Architecture[3]

Links Table

[1] *http://msdn.microsoft.com/en-us/library/ms186981.aspx*

[2] *http://msdn.microsoft.com/en-us/library/ms179316.aspx*

[3] *http://msdn.microsoft.com/en-us/library/ms180978.aspx*

Community Content

PAGE SIZE

A page is 8KB. 1024 bytes X 8KB = 8192 Bytes.
Header is 96 Bytes
Row offset is 36 Bytes
Total 132 Bytes
So, 8192 Bytes - 132 Bytes = 8060 Bytes.

11/7/2011
Eldho Scaria - Intel BITS

CLR data types

There is no mention of where CLR data types are stored..

9/23/2011
ChinnoDog

This article can be little more precise.

1. You have mentioned that *96-byte header is used to store system information about the page*, at the same time you could also mentioned the space used by the **row offset table**.

2. You mentioned that the *maximum amount of data and overhead that is contained in a single row on a page is 8,060 bytes (8 KB)*. Here I don't understand why you mentioned 8 KB in the bracket. 8,060 Bytes != 8 KB.

3. It will be more informative, If you detail the space allocation (i.e.) how the 8 KB (Single Page) is classified.
For example: Single row on a page (8,060 bytes) + Header (96 bytes) + __________= 8,192 (8 KB)

9/15/2011
rajakumarn

9/15/2011
rajakumarn