# Pages and Extents Architecture Guide

## Pages and Extents

The fundamental unit of data storage in SQL Server is the page. 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

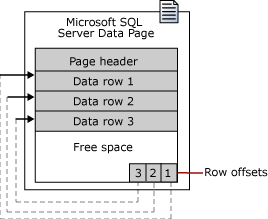
In SQL Server, the page size is 8 KB. This means SQL Server databases have 128 pages per megabyte. Each page begins with a 96-byte header that is used to store system information about the page. This information includes the page number, page type, the amount of free space on the page, and the allocation 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 Page** | 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 Page** | Index entries. |
| **LOB Page** | 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 (PFS)** | 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

SQL Server uses 13 types of disk pages.



**Row Offset Table:**

Row offset table contains one entry for each row on 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:

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.

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.

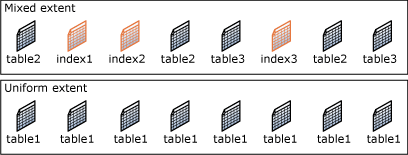
### Extents

Extents are the basic unit in which space is managed. An extent is eight physically contiguous pages, or 64 KB (8 Data pages)

**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.

* 

### Managing Extent Allocations

SQL Server uses two types of allocation maps to record the allocation of extents:

* **Global Allocation Map (GAM)**  
  GAM pages record what extents have been allocated. Each GAM covers 64,000 extents, or almost 4 GB of data. The GAM has one bit for each extent in the interval it covers. If the bit is 1, the extent is free; if the bit is 0, the extent is allocated.
* **Shared Global Allocation Map (SGAM)**   
  SGAM pages record which extents are currently being used as mixed extents and have at least one unused page. Each SGAM covers 64,000 extents, or almost 4 GB of data. The SGAM has one bit for each extent in the interval it covers. If the bit is 1, the extent is being used as a mixed extent and has a free page. If the bit is 0, the extent is not used as a mixed extent, or it is a mixed extent and all its pages are being used.

This causes simple extent management algorithms.

* To allocate a uniform extent, the SQL Server Database Engine searches the GAM for a 1 bit and sets it to 0.
* To find a mixed extent with free pages, the SQL Server Database Engine searches the SGAM for a 1 bit.
* To allocate a mixed extent, the SQL Server Database Engine searches the GAM for a 1 bit, sets it to 0, and then also sets the corresponding bit in the SGAM to 1.
* To deallocate an extent, the SQL Server Database Engine makes sure that the GAM bit is set to 1 and the SGAM bit is set to 0.

### Tracking free space

**Page Free Space (PFS)** pages record the allocation status of each page, whether an individual page has been allocated, and the amount of free space on each page. The PFS has one byte for each page, recording whether the page is allocated, and if so, whether it is empty, 1 to 50 percent full, 51 to 80 percent full, 81 to 95 percent full, or 96 to 100 percent full. The second page (page 1) of a file is a PFS page, as is every 8,088th page thereafter.

## Managing space used by objects

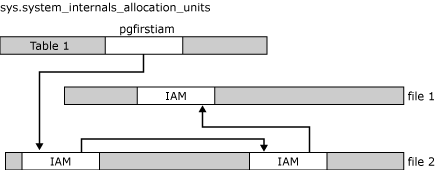
An **Index Allocation Map (IAM)** page maps the extents in a 4-gigabyte (GB) part of a database file used by an allocation unit. An allocation unit is one of three types:

* **IN\_ROW\_DATA**  
  Holds a partition of a heap or index.
* **LOB\_DATA**  
  Holds large object (LOB) data types, such as xml, varbinary(max), and varchar(max).
* **ROW\_OVERFLOW\_DATA**  
  Holds variable length data stored in varchar, nvarchar, varbinary, or sql\_variant columns that exceed the 8,060 byte row size limit.

Each partition of a heap or index contains at least an IN\_ROW\_DATA allocation unit. It may also contain a LOB\_DATA or ROW\_OVERFLOW\_DATA allocation unit.

An IAM page covers a 4-GB range in a file and is the same coverage as a GAM or SGAM page. If the allocation unit contains extents from more than one file, or more than one 4-GB range of a file, there will be multiple IAM pages linked in an IAM chain.

Select \* from sys.system\_internals\_allocation\_units



IAM pages linked in a chain per allocation unit An IAM page has a header that indicates the starting extent of the range of extents mapped by the IAM page. The IAM page also has a large bitmap in which each bit represents one extent. The first bit in the map represents the first extent in the range, the second bit represents the second extent, and so on. If a bit is 0, the extent it represents is not allocated to the allocation unit owning the IAM. If the bit is 1, the extent it represents is allocated to the allocation unit owning the IAM page.

When the SQL Server Database Engine has to insert a new row and no space is available in the current page, it uses the IAM and PFS pages to find a page to allocate, or, for a heap or a Text/Image page, a page with sufficient space to hold the row. The SQL Server Database Engine uses the IAM pages to find the extents allocated to the allocation unit. For each extent, the SQL Server Database Engine searches the PFS pages to see if there is a page that can be used. Each IAM and PFS page covers many data pages, so there are few IAM and PFS pages in a database. This means that the IAM and PFS pages are generally in memory in the SQL Server buffer pool, so they can be searched quickly.

## Tracking Modified Extents

SQL Server uses two internal data structures to track extents modified by bulk copy operations and extents modified since the last full backup. These data structures greatly speed up differential backups. They also speed up the logging of bulk copy operations when a database is using the bulk-logged recovery model. Like the Global Allocation Map (GAM) and Shared Global Allocation Map (SGAM) pages, these structures are bitmaps in which each bit represents a single extent.

**Differential Changed Map (DCM)**   
This tracks the extents that have changed since the last BACKUP DATABASE statement. If the bit for an extent is 1, the extent has been modified since the last BACKUP DATABASE statement. If the bit is 0, the extent has not been modified.

**Bulk Changed Map (BCM)**

This tracks the extents that have been modified by bulk-logged operations since the last BACKUP LOG statement. If the bit for an extent is 1, the extent has been modified by a bulk logged operation after the last BACKUP LOG statement. If the bit is 0, the extent has not been modified by bulk logged operations.

**Minimal Logging Changed Map (ML)**on the eighth page (page 7) is used when an extent in the file is used in a minimally or bulk-logged operation.

Data pages two types of Large Object (LOB) pages, row-overflow pages, index pages, Page Free Space (PFS) pages, Global Al­location Map and Shared Global Allocation Map (GAM and SGAM) pages, Index Allocation Map (IAM) pages, Minimally Logged (ML) pages, and Differential Changed Map (DIFF) pages.

Like GAM and SGAM pages, DIFF and ML map pages have 1 bit for each extent in the section of the file they represent. They occur at regular intervals—every 511,230 pages.