

Firebird for the database expert: episode 2 - Page types

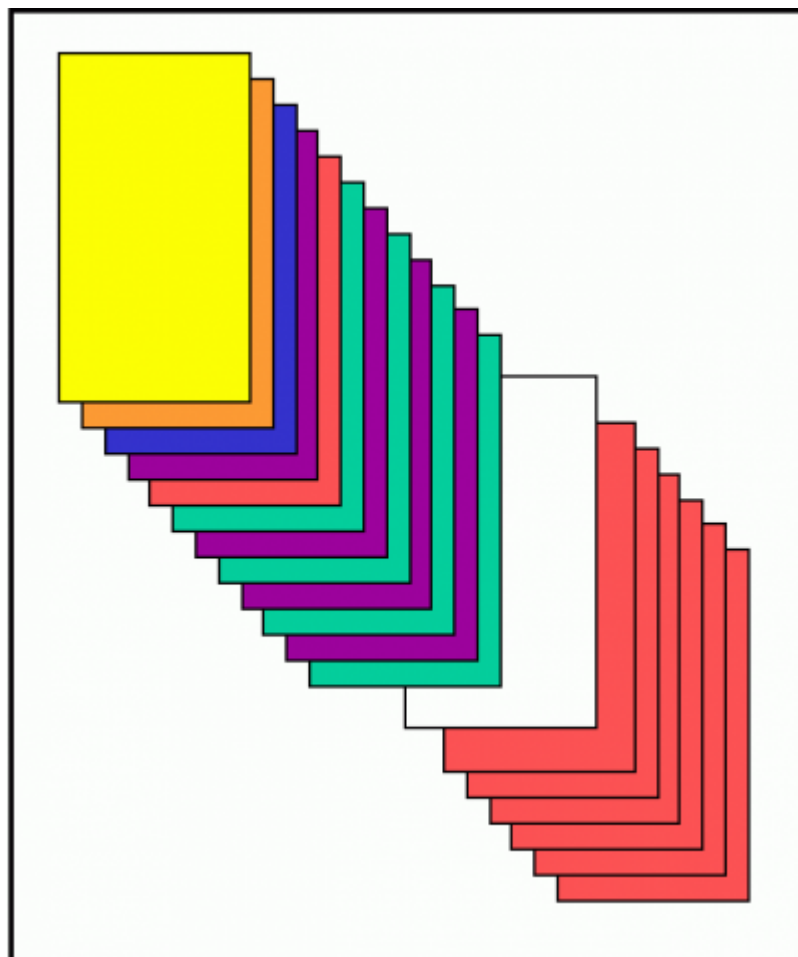
[Youtube Tutorial Available](#)

By Ann Harrison

Database file

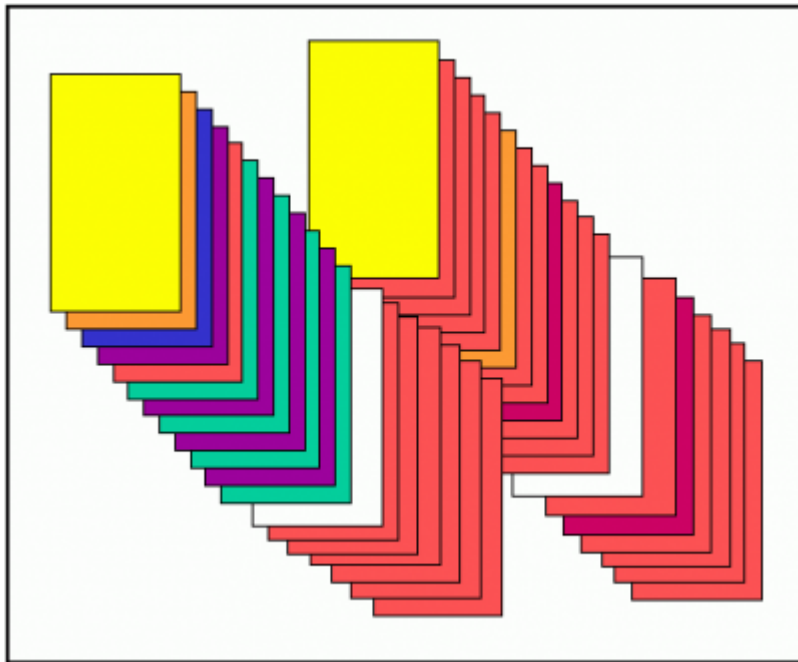
A Firebird database is a sequence of fixed length pages normally all contained in a single file.

Different pages have different functions - in this case the yellow page is the [database header](#), followed by a [PIP](#), the unused [WAL](#), a [pointer page](#), a [data page](#), then alternating [index root](#) and [pointer](#) pages. The white page indicates that the diagram skips several hundred pages then continues with data pages.



Multi-file database

A multi-file [database](#) breaks the sequence into multiple files, each with a header page. Aside from the extra [header pages](#), there is no difference between a multi-file database and a single file database.



Generic page header

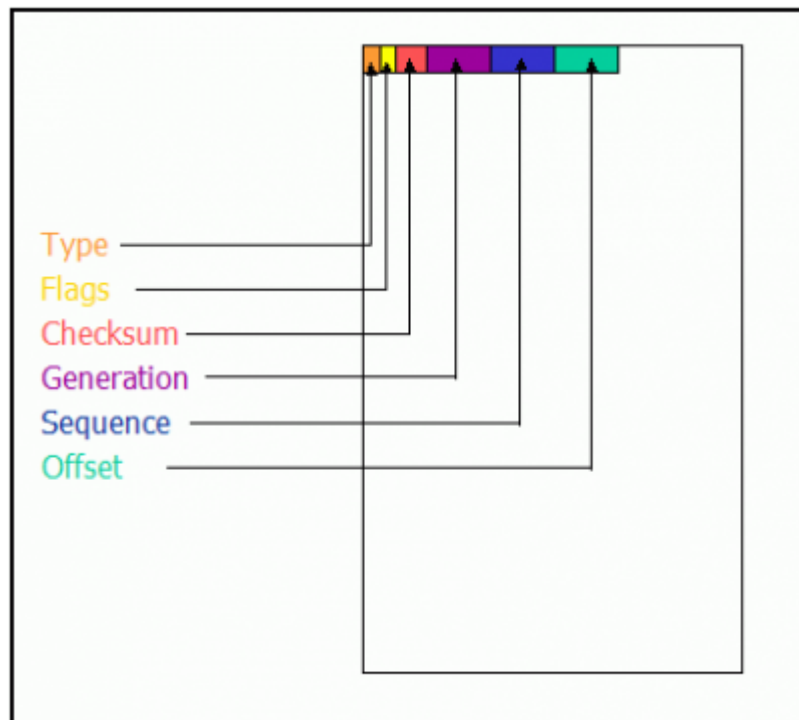
Each page has a header that indicates what type of page it is, and provides other information that applies to all pages. Most page types have additional header information that follows the standard header. In the standard header, the first byte is the page type.

The next byte contains flags that are specific to individual page types. Currently, only [blob pages](#) and [b-tree \(index\) pages](#) use the page flags. Other page types - the header for one - also have a separate area for flags.

The next two bytes were a checksum, but now always contain the value 12345.

The next four bytes are the page generation incremented each time the page is written.

The next eight bytes are reserved for the sequence and offset of the page's entry in a log. The logging project has been abandoned and those bytes are waiting for a good use.



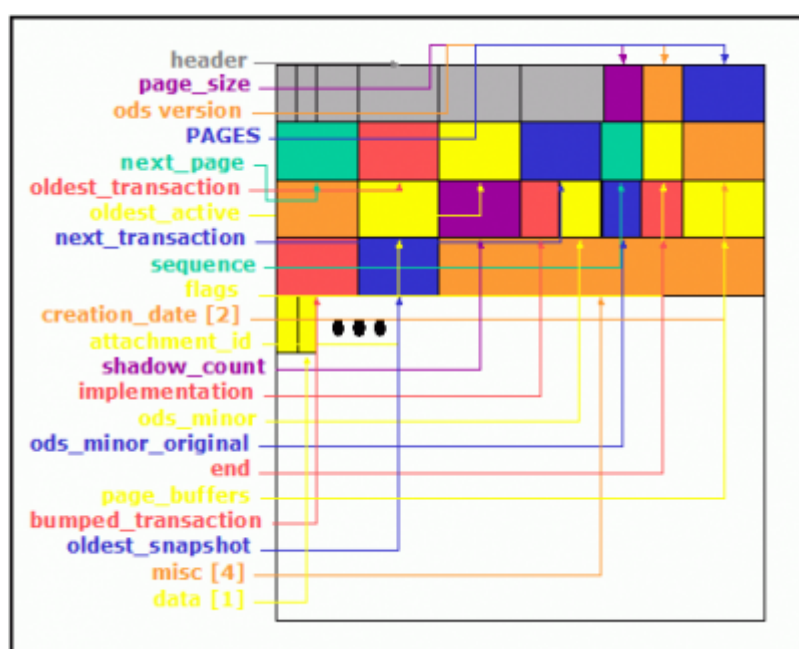
[back to top of page](#)

Header page (HDR)

Page Type 1 is a header page. Each database file has one header page, which is page 0 in the file.

The first header page in a database describes the database: the [page size](#), next [transaction id](#), various settings, etc.

The header pages of subsequent files in the database contain only the length of the current file and the name of the next file.



Please also refer to [Structure of a header page](#).

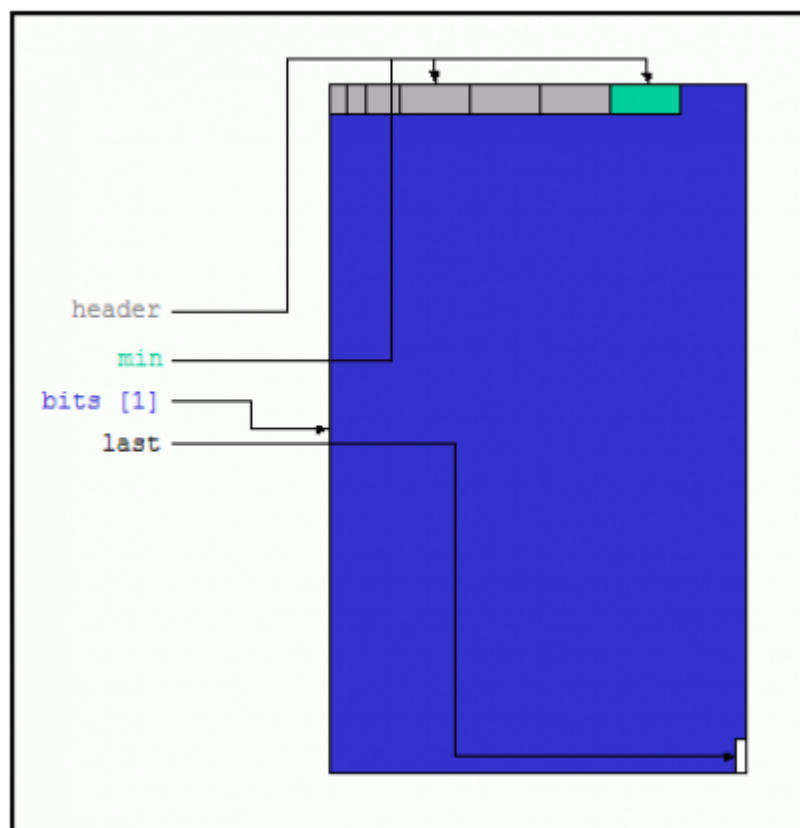
[back to top of page](#)

Page Inventory Page (PIP)

Page Type 2 is a [page inventory page \(PIP\)](#). PIPs map allocated and free pages. The header of a PIP includes the offset on this page of the bit that indicates the first available page on the PIP.

The body of a PIP contains an [array](#) of single bits that reflect the state of pages in the database. If the bit is one, then the corresponding page is not in use. If the bit is zero, then the page is in use.

PIPs occur at regular intervals through the database, starting at page 1. The last page allocated on each PIP is the next PIP.



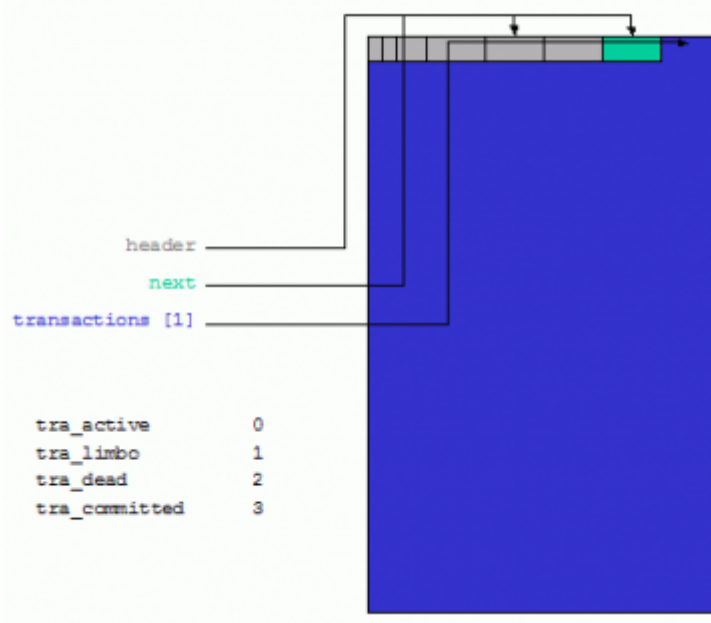
[back to top of page](#)

Transaction Inventory Page (TIP)

Page Type 3 is a [transaction information page \(TIP\)](#). The TIP header includes the address of the next TIP.

The body of a TIP is an [array](#) of pairs of bits that reflect the state of [transactions](#). If both bits are 0, the transaction is active or has not started. If both bits are 1, the transaction is [committed](#). If the first bit is 1 and the second bit is 0, the transaction is [in limbo](#). If the first bit is 0 and the second is 1, the transaction is [in limbo](#).

Limbo is the state of a two phase transaction that has completed the first phase, but not the second.



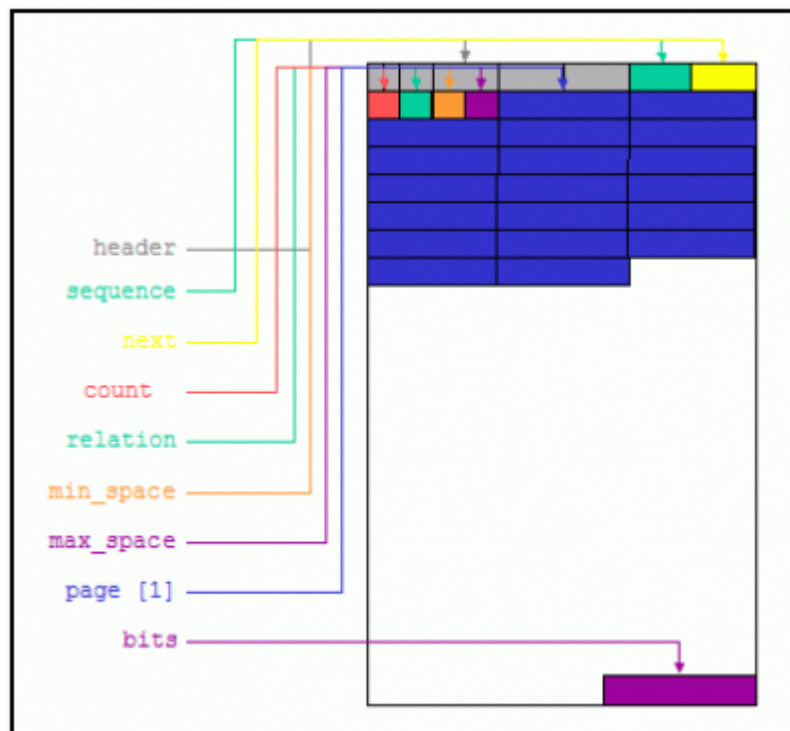
[back to top of page](#)

Pointer page (PTR)

Page Type 4 is a pointer page. Each pointer page belongs to a particular [table](#) and has a specific sequence within the table.

The additional header information on a pointer page includes its sequence in the pointer pages for this table, the page number of the next pointer page for the table, the next free slot on the page, the number of used slots on the page, the relation id of the table, the offset of the first slot on the page that indicates a page that is not full, and the offset of the last slot on the page that indicates a [data page](#) that is not full.

Pointer pages contain [arrays](#) of 32-bit [integers](#) that contain the page numbers of pages in a table. At the bottom of the pointer page, an array of bits indicates the fill level of each page.



[back to top of page](#)

Data page (DPG)

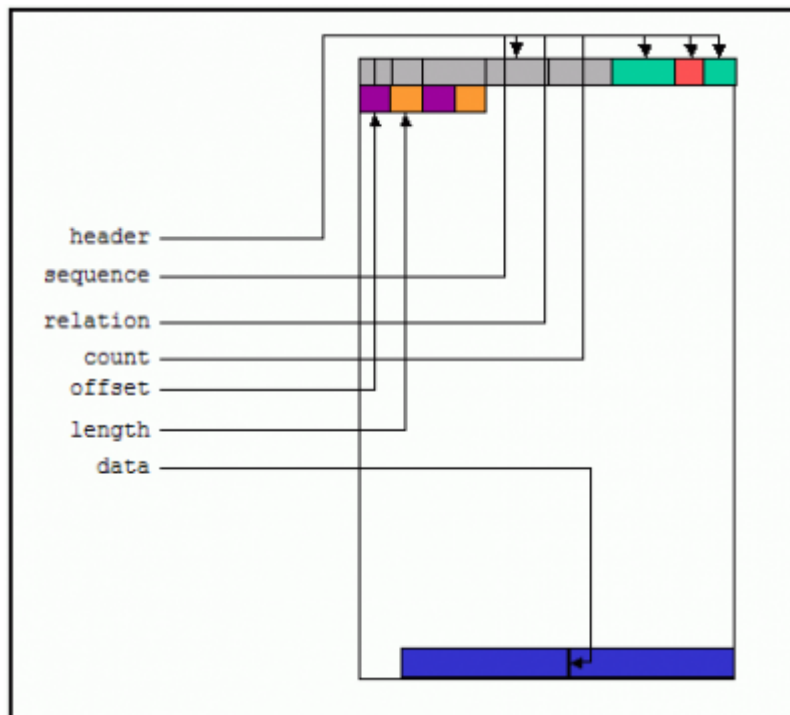
Page Type 5 is a data page. Each data page belongs to a specific [table](#).

The additional header information in a data page is the position of this page in the list of data pages for the table, the relation id of the table, and the number of entries on this page.

The body of a data pages starts with an [array](#) of pairs of 16 bit words. The first part of the pair is the offset on the page of a piece of data - a record, [blob](#), or record fragment. The second part of the pair is the length of the data. As more data is stored on the page, the [index](#) grows downward.

The data - records, blobs, and fragments - start at the end of the page and go upward.

Further information can be found in the chapters, [Structure of a data page](#) and [Where do data pages come from](#).



[back to top of page](#)

Index Root page (IRT)

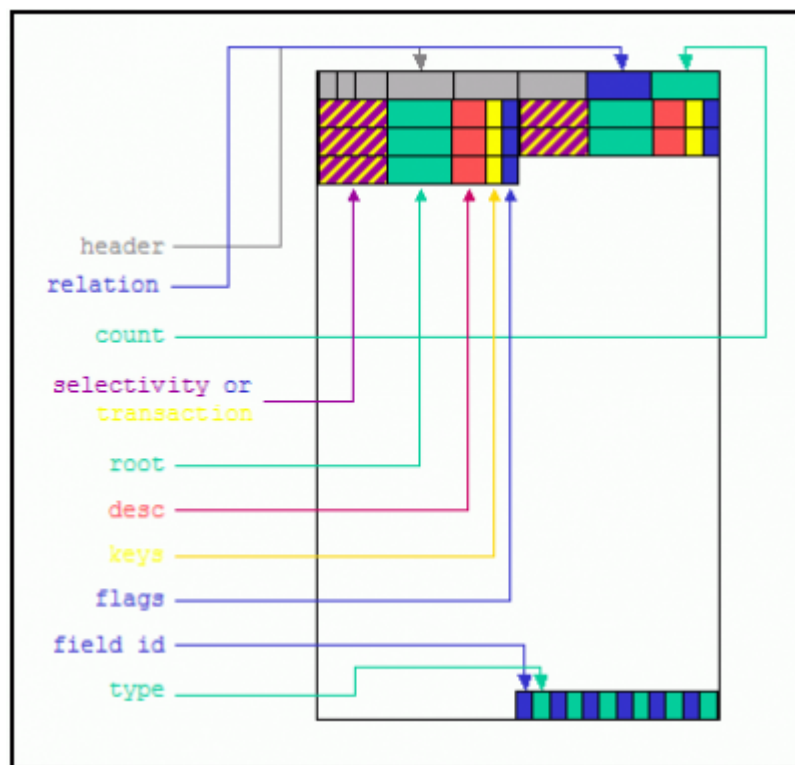
Page Type 6 is an [index](#) root page. Each [table](#) has a single index root page that describes the [indexes](#) for the table. This page describes the IRT in Firebird 1.5 and earlier.

The additional header information for an index root page is the identifier of the relation to which the page belongs, and a count of the number of indexes for that table.

The body of an index root page contains an [array](#) of index descriptors coming down from the top of the page and an array of index segment descriptors coming up from the bottom.

Each index descriptor starts with the selectivity if the index has already been created, or a [transaction id](#) if the index is being created. The next 32 bits are the page number of the top of the actual index. Next is the 32-bit offset of the field descriptors for the index at the bottom of the page. The next byte is the number of key [fields](#), then a flag byte.

The array of segment descriptors contains two bytes per segment, one for the field id and one for the field type.



[back to top of page](#)

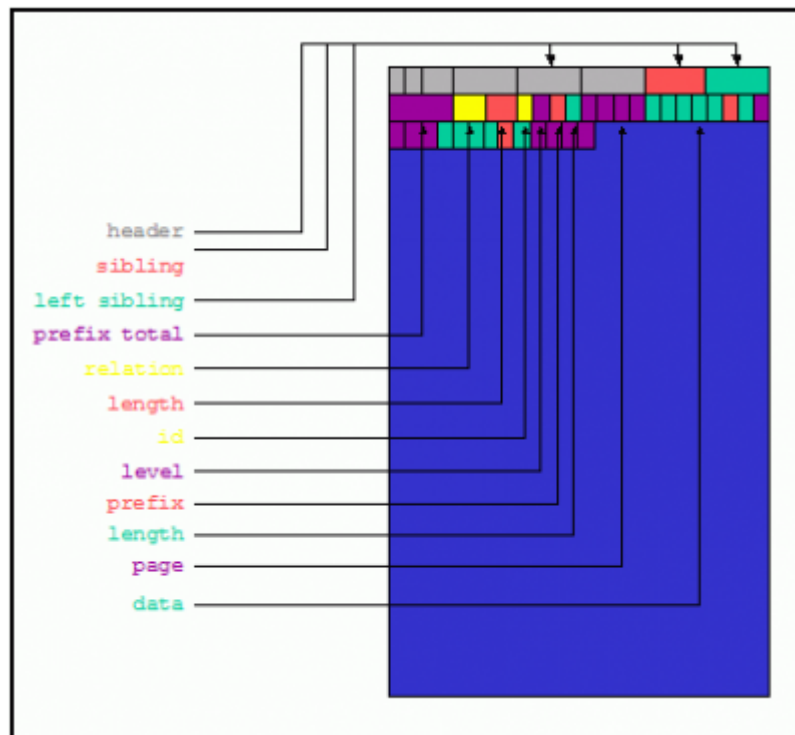
B-tree page (BTR)

Page Type 7 is an index or b-tree page.

All [indexes](#) in Firebird are a b-tree variant, starting with a single page at the top - confusingly called the root - confusing both because the root is at the top and because the root of an index is different from the [table's](#) index root page.

The additional header data in a b-tree page includes the number of the page with the next higher values for this level of the index, the address of the page with the next lower values for this level, the total amount of space which is saved on this page by the use of prefix compression, the relation id of the table this index describes, the amount of space used on this page, the identifier of the index in which this page participates, and the level of this page in the index.

The rest of the page is filled with index entries.



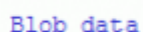
[back to top of page](#)

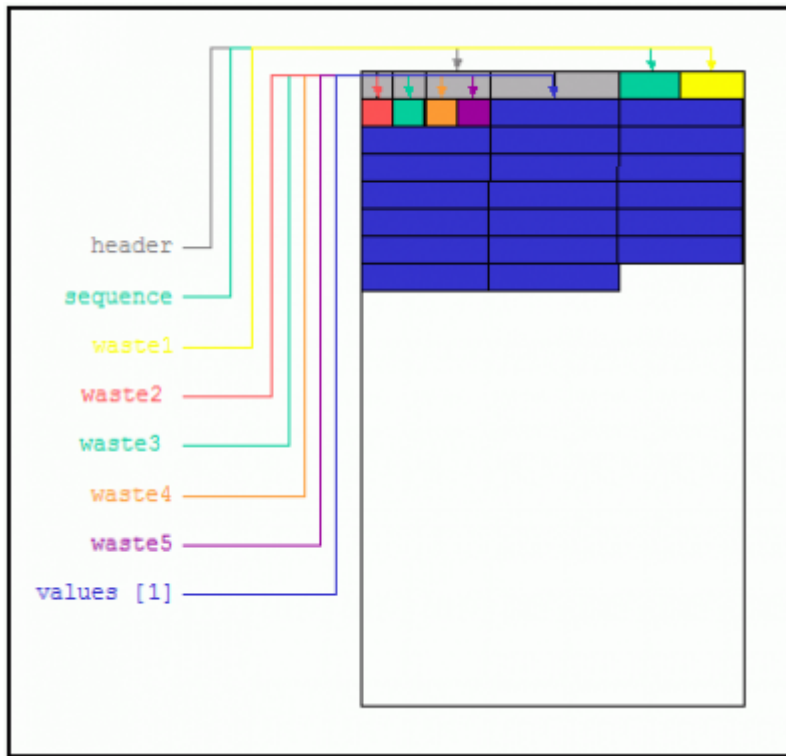
Blob page (BLP)

Page Type 8 is a [blob](#) page. Small blobs are stored on [data pages](#). Blobs larger than a page are stored on a sequence of blob pages.

The type-specific header information for a blob page includes the page number of the first page of this blob, the position (sequence) of this page in the list of pages that contain the blob, the amount of [data](#) stored on the page, and a pad word to allow the blob data to start on a long word boundary.

The remainder of the page contains blob data for a single blob.





This paper was written by Ann Harrison in June 2005, and is copyright Ms. Harrison and IBPhoenix.

From:
<http://ibexpert.com/docu/> - IBExpert

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-05-database-technology:database-technology-articles:firebird-interbase-ods:firebird-for-the-database-expert-page-types>

Last update: 2023/06/16 12:57

