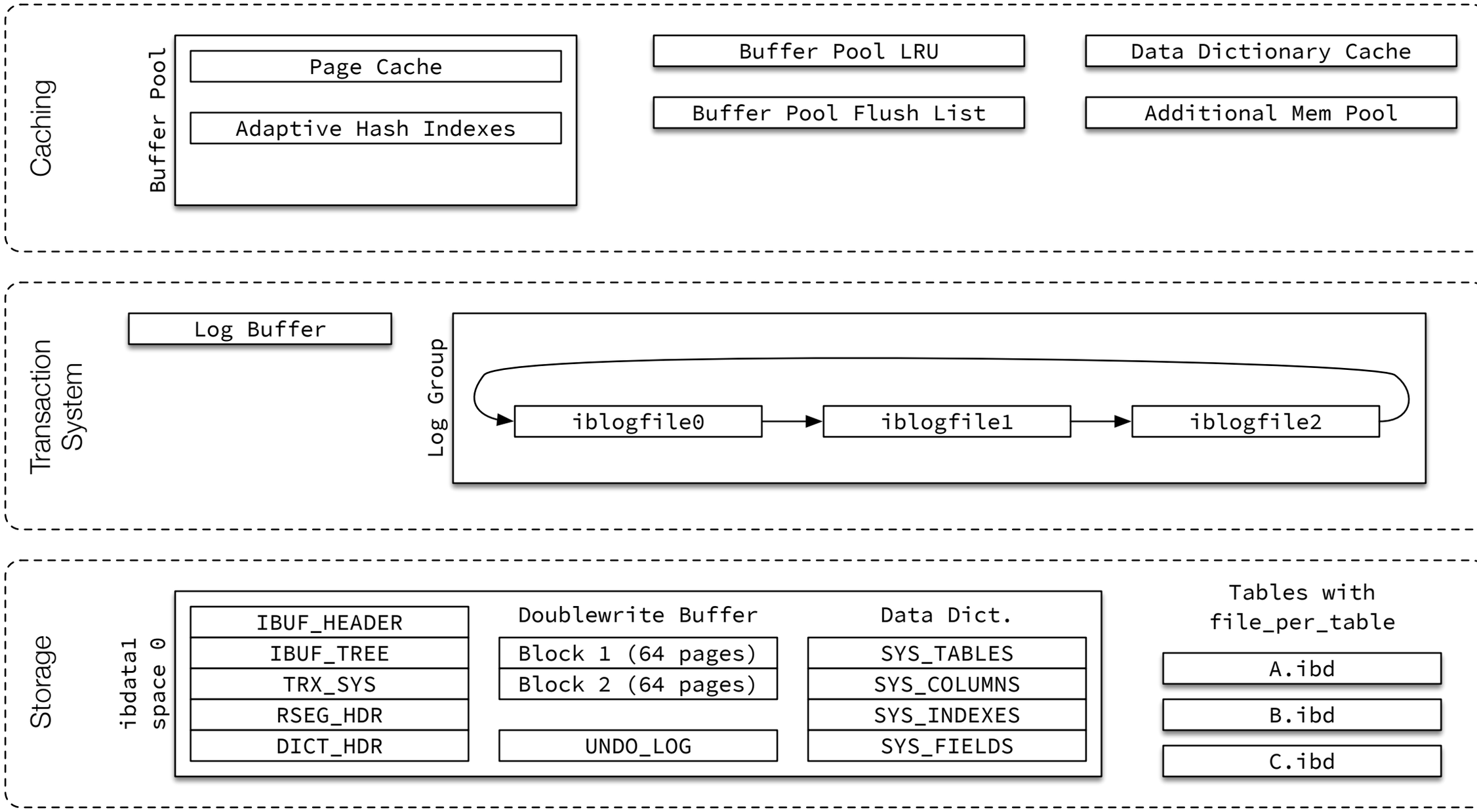
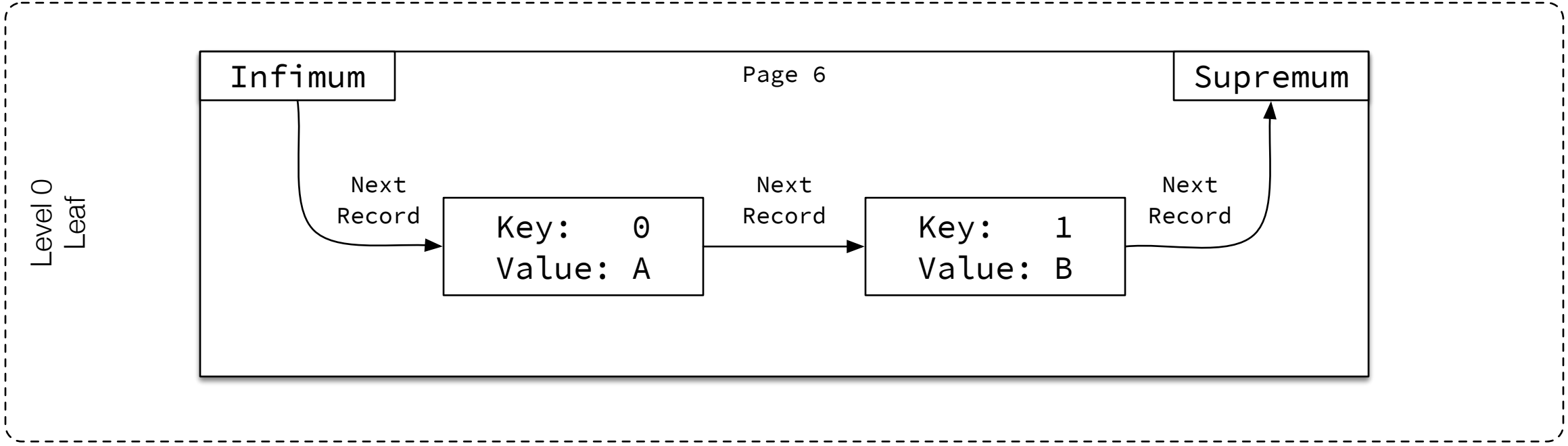
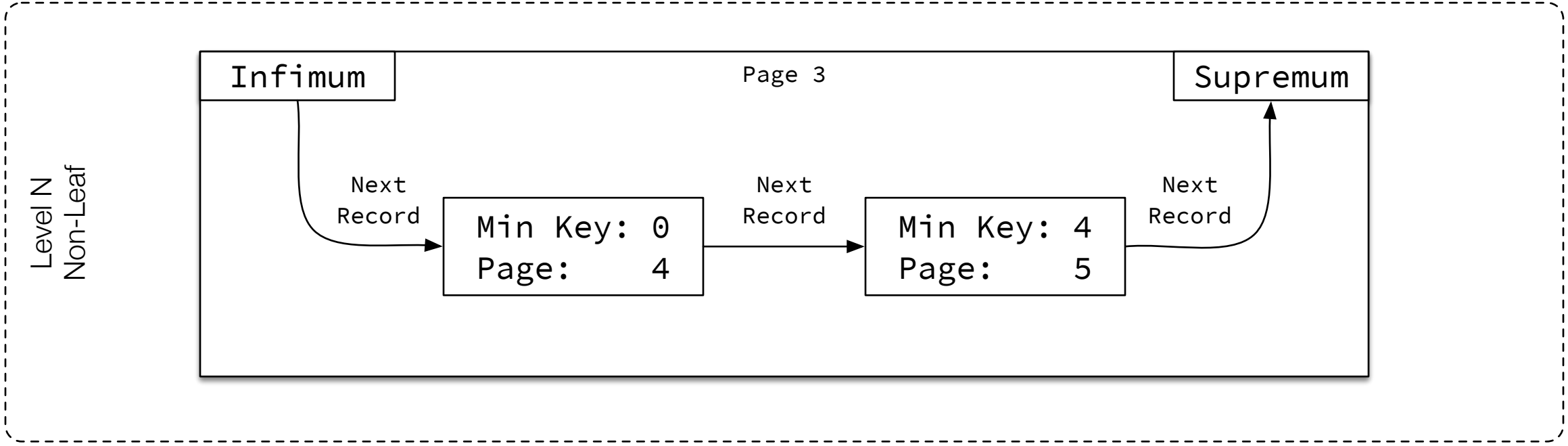
High-level Overview



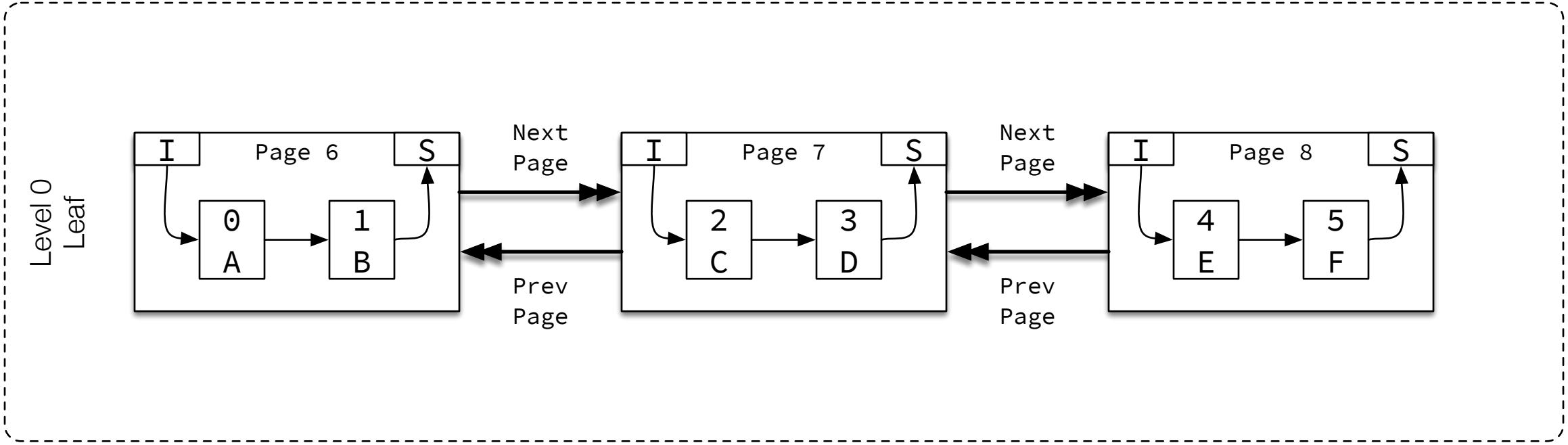
B+Tree Simplified Leaf Page



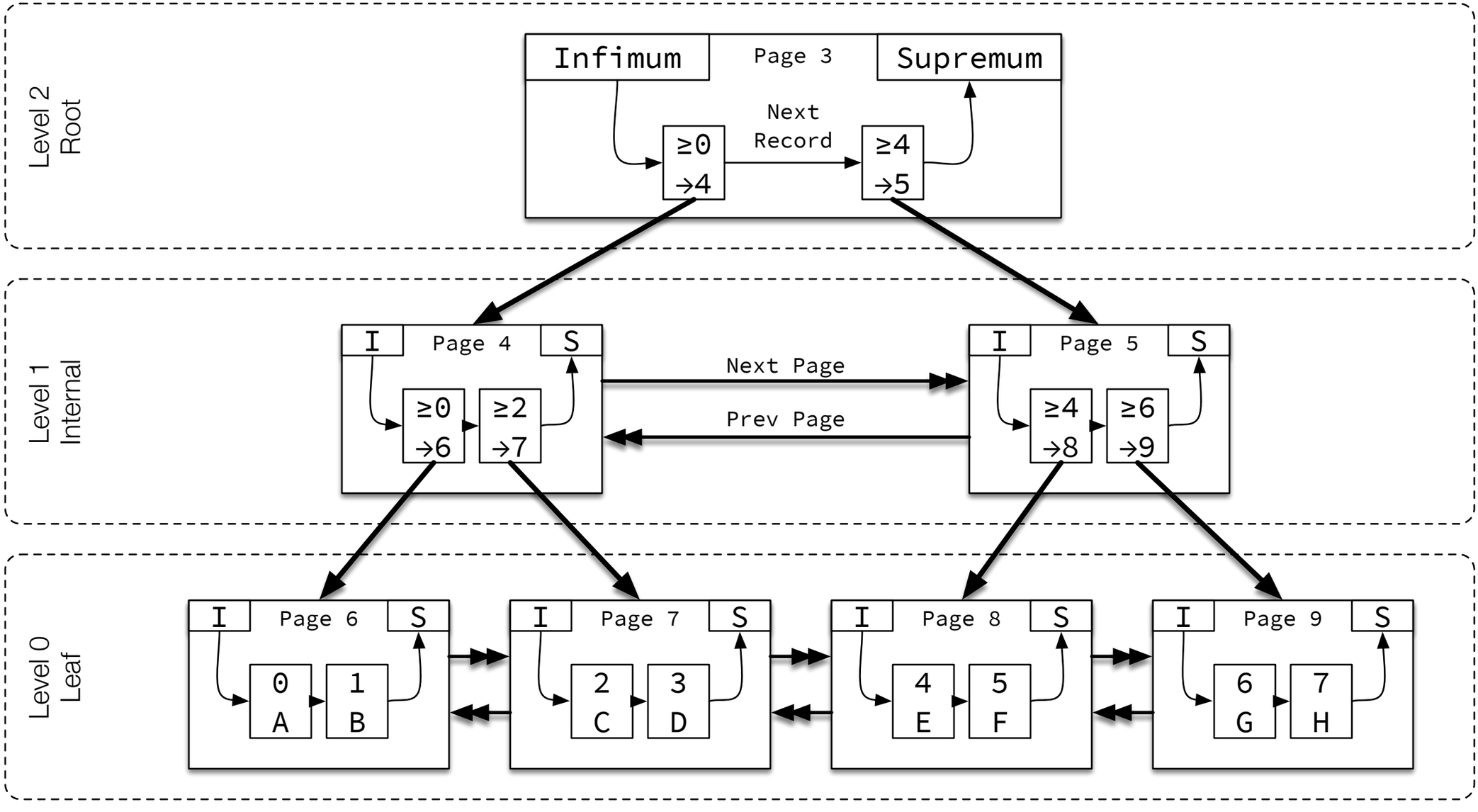
B+Tree Simplified Non-Leaf Page



B+Tree Simplified Level

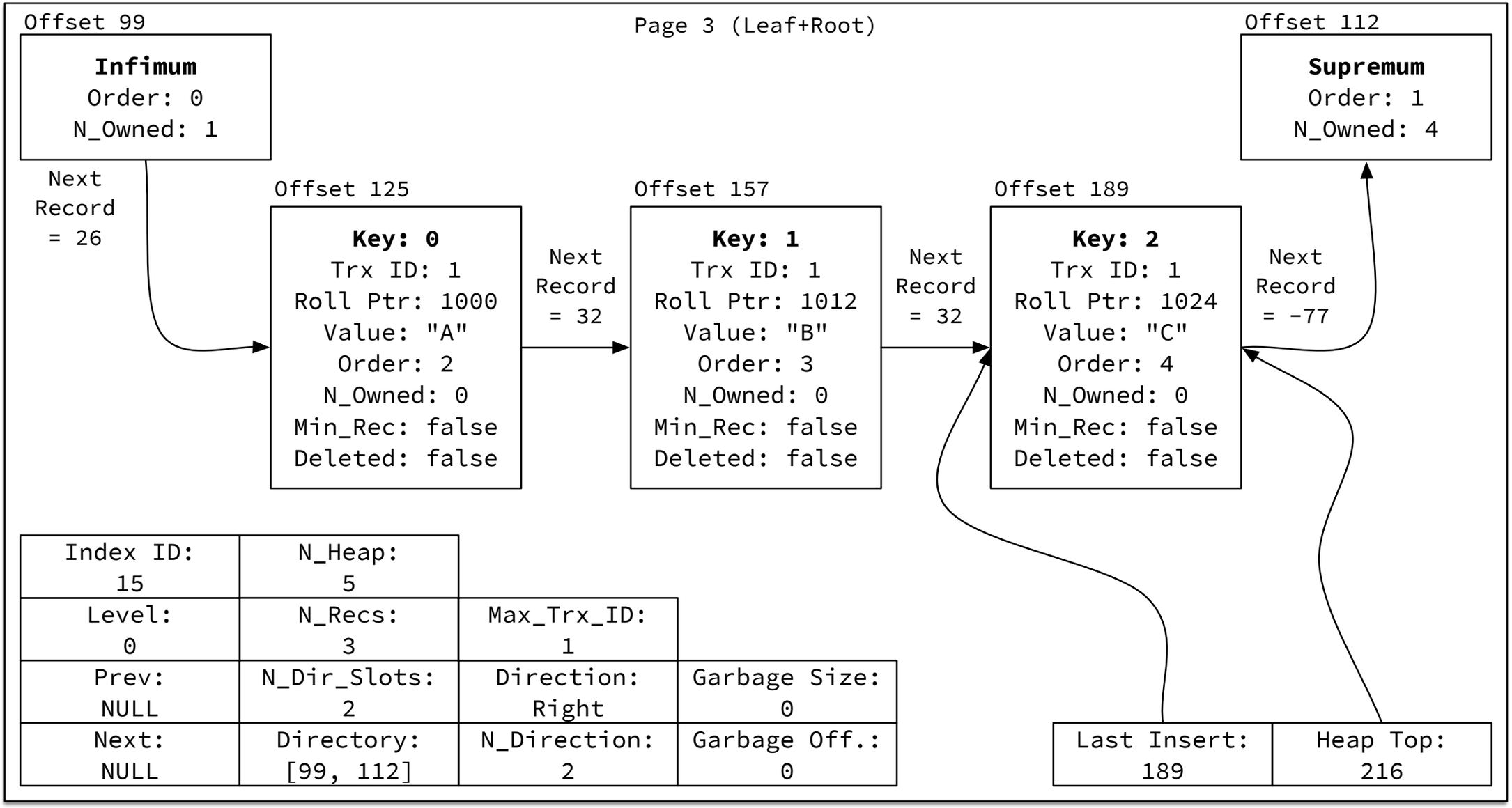


B+Tree Structure



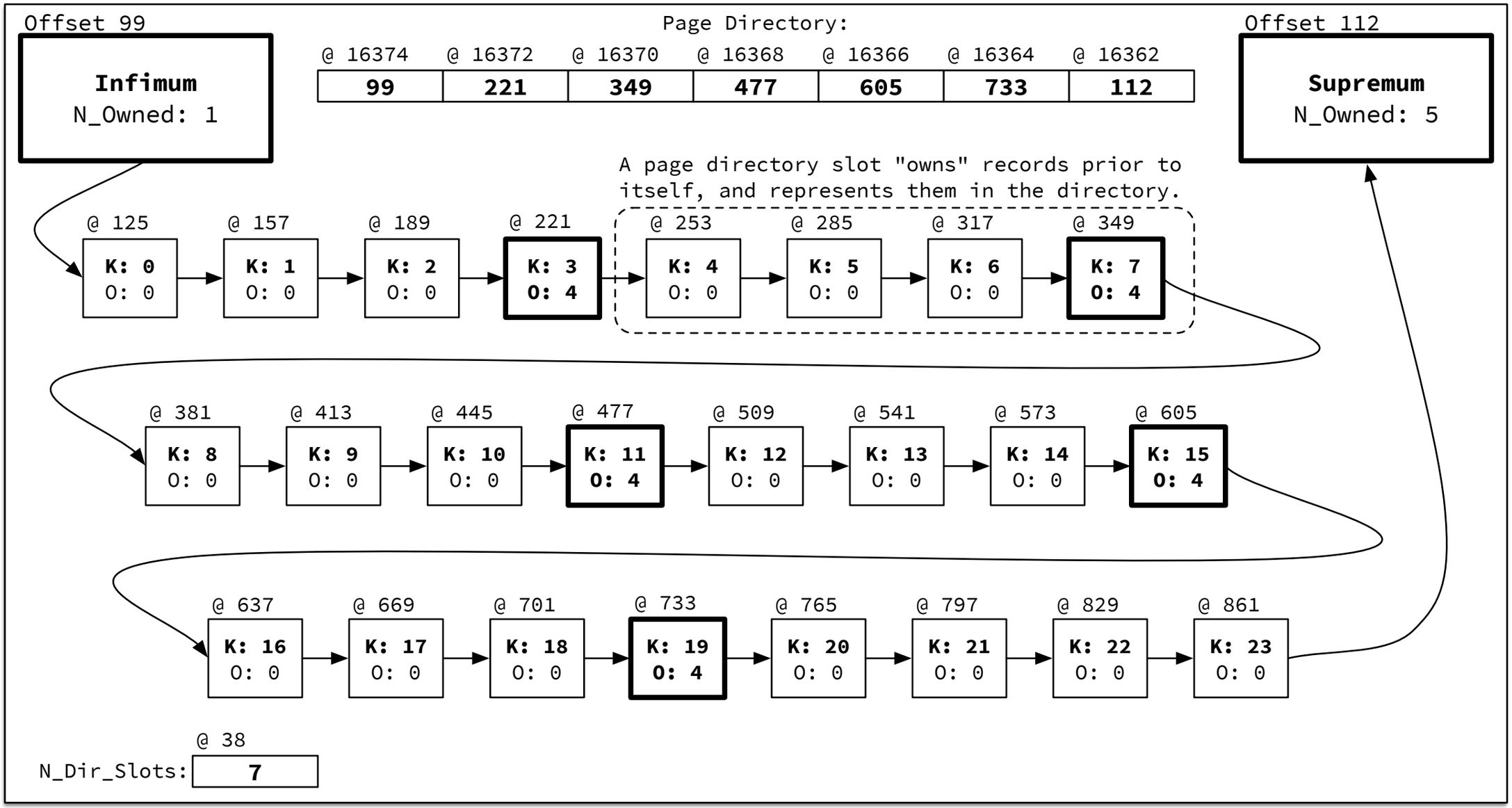
Levels are numbered starting from 0 at the leaf pages, incrementing up the tree.
Pages on each level are doubly-linked with previous and next pointers in ascending order by key.
Records within a page are singly-linked with a next pointer in ascending order by key.
Infimum represents a value lower than any key on the page, and is always the first record in the singly-linked list of records.
Supremum represents a value higher than any key on the page, and is always the last record in the singly-linked list of records.
Non-leaf pages contain the minimum key of the child page and the child page number, called a "node pointer".

B+Tree Detailed Page Structure



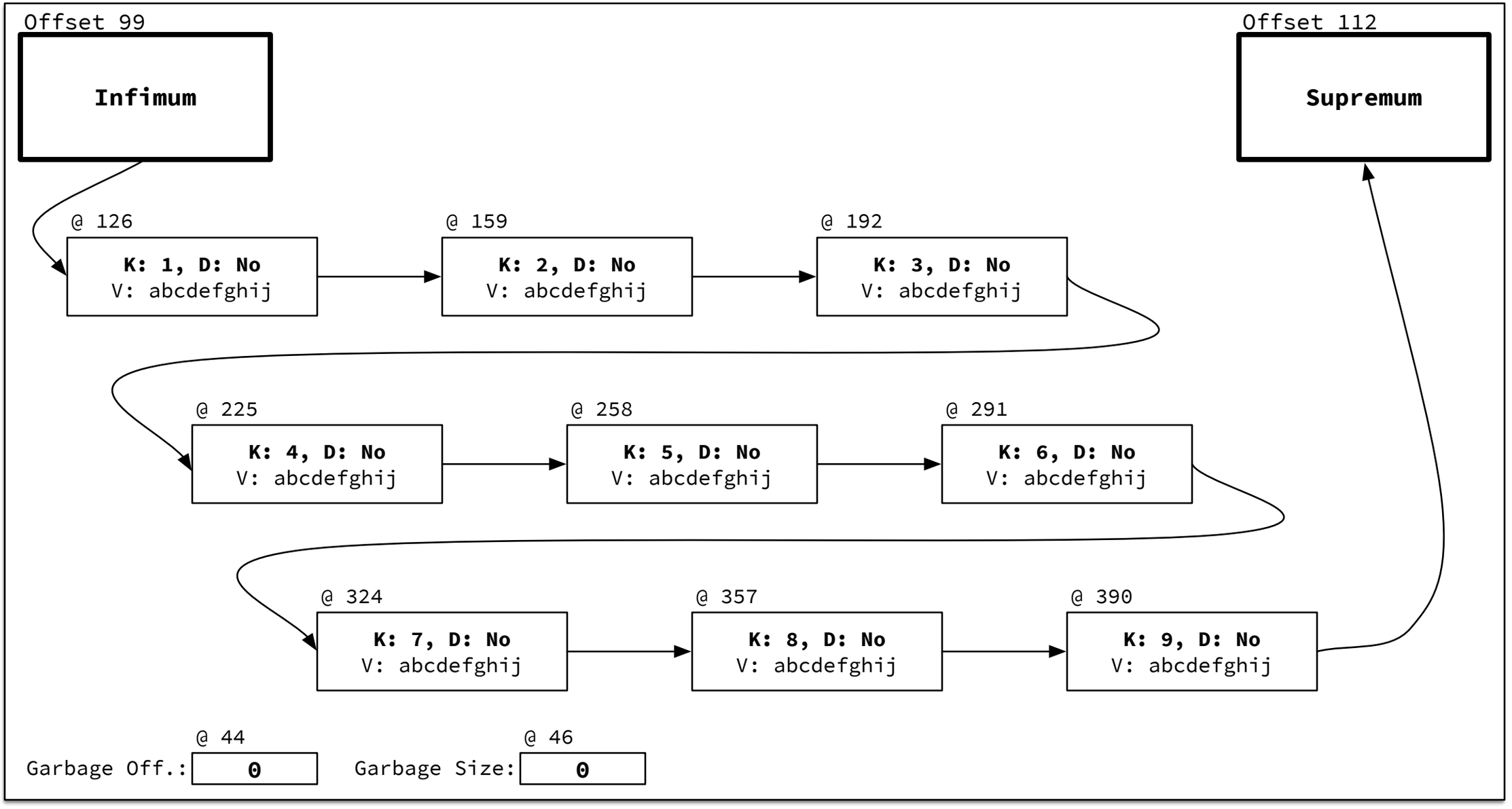
InnoDB table format is Barracuda with "compact" record structure, non-compressed.
Table created with: `CREATE TABLE t (i INT NOT NULL, s CHAR(10) NOT NULL, PRIMARY KEY(i)) ENGINE=InnoDB;`
Table populated with: `INSERT INTO t (i, s) VALUES (0, "A"), (1, "B"), (2, "C");`
Record size: 5 (header) + 4 (PK) + 6 (TRX_ID) + 7 (ROLL_PTR) + 10 (non-key fields) = 32 bytes

B+Tree Page Directory Structure



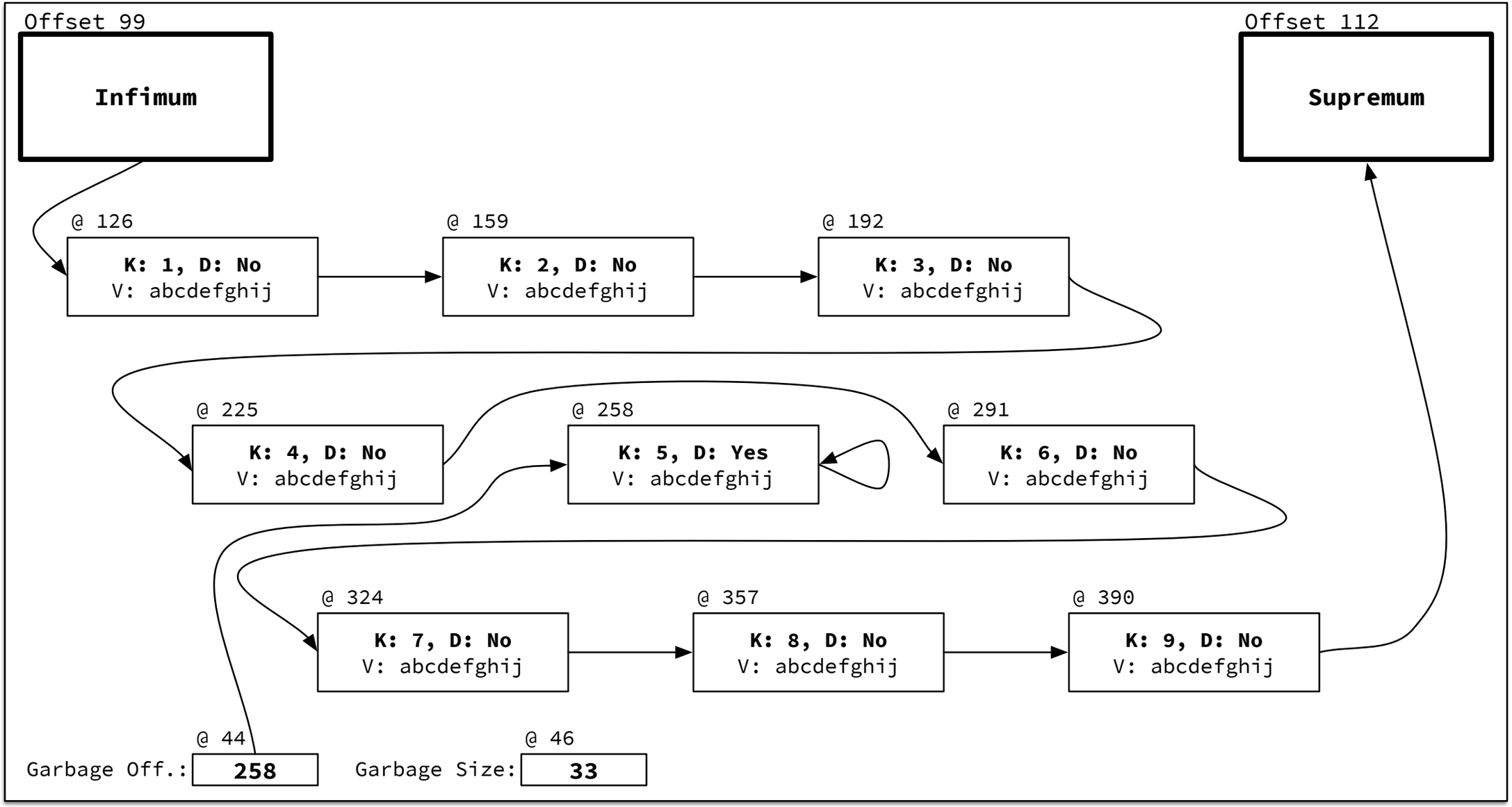
Infimum always owns only itself, so will always have a slot in the page directory with N_Owned = 1.
Supremum always owns the last few records in the page, and is allowed to own less than 4 records (if the page has fewer).
All directory slots will own a minimum of 4 and maximum of 8 records, except supremum, which may own fewer.
The page directory grows "downwards" from offset 16376, the beginning of the FIL trailer; the first directory entry starts at 16374.

B+Tree Record Initial State



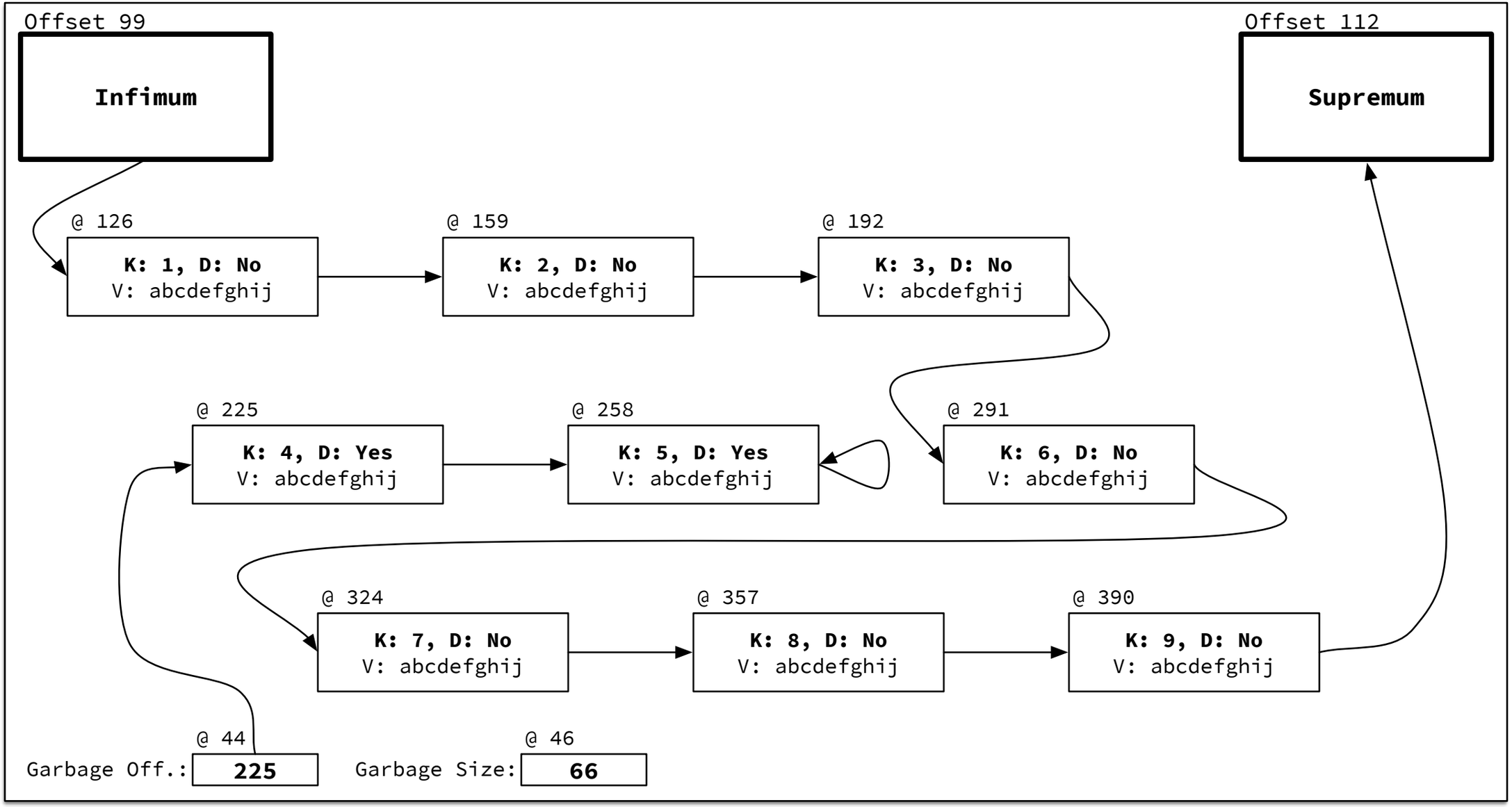
SQL: create table t (i int not null, s varchar(100) not null, primary key(i)) engine=innodb;
SQL: insert into t (i, s) values (1, "abcdefghij"); for i in 1..9

B+Tree Record Delete 1



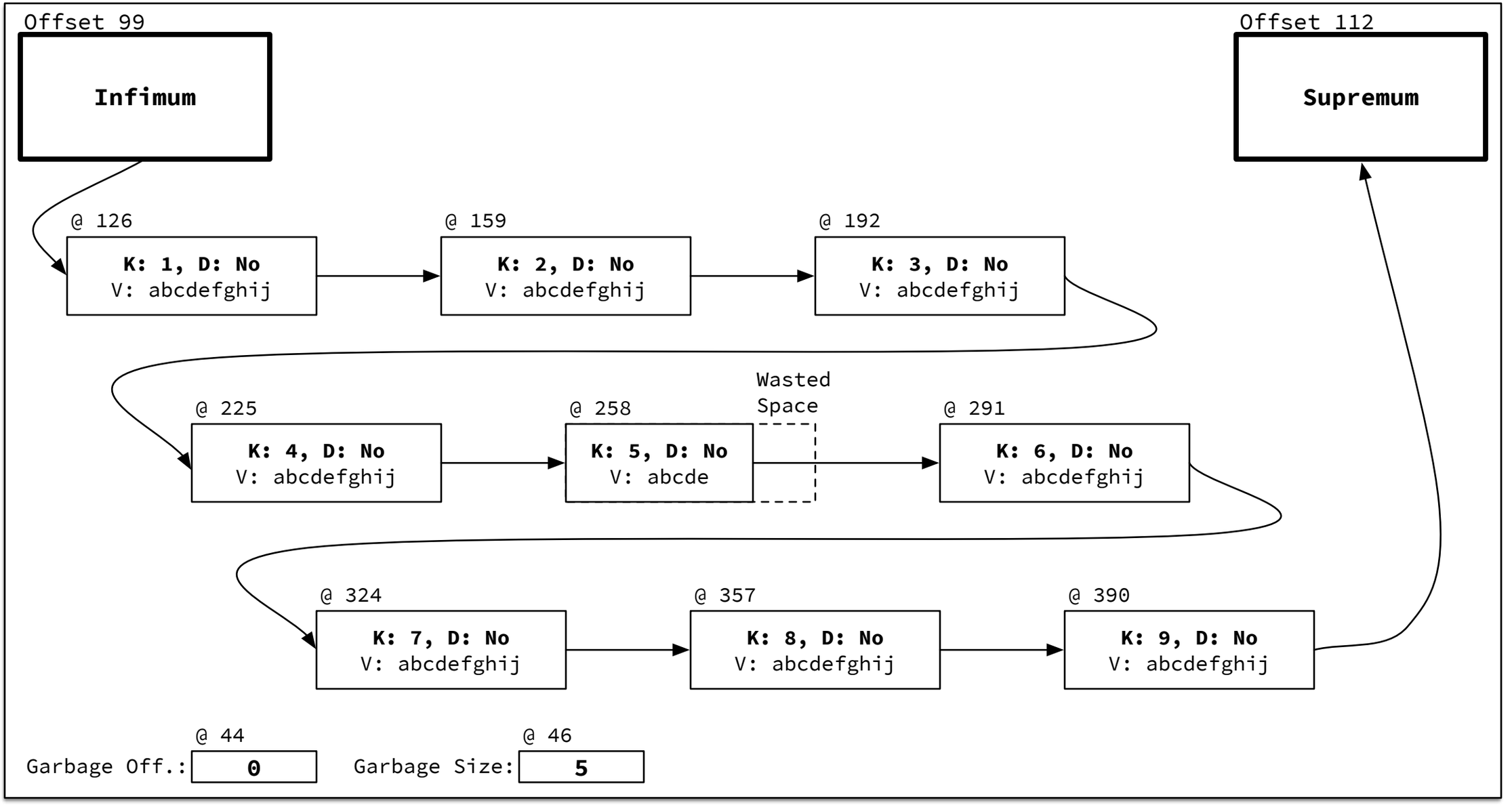
SQL: delete from t where i = 5;
Row is marked as deleted.
Garbage size is incremented by total row size.
Garbage offset is pointed to row, and row next pointer is pointed back to self.

B+Tree Record Delete 2



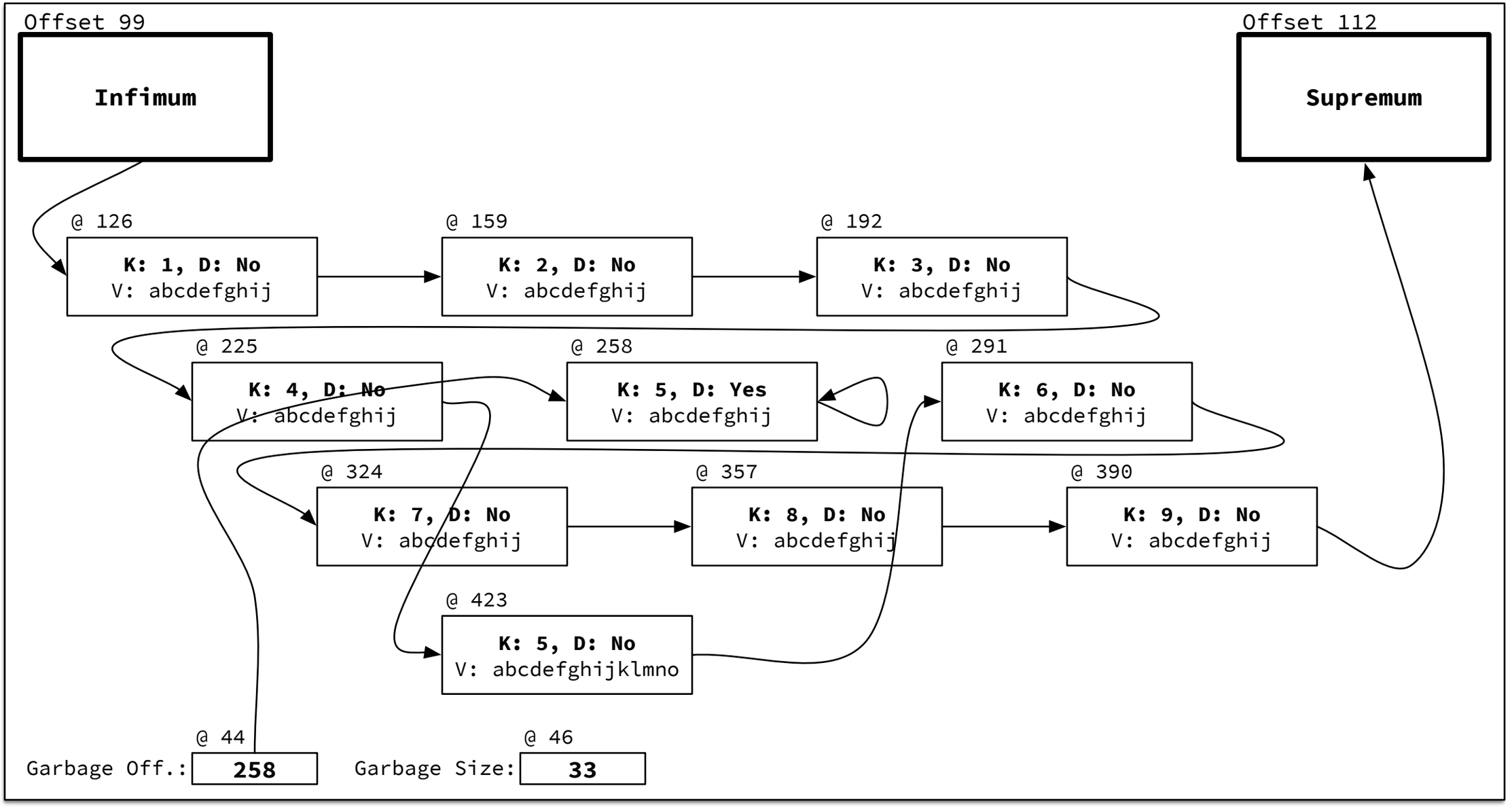
SQL: delete from t where i = 5; delete from t where i = 4;
Garbage size is incremented by total row size for each delete.
Garbage offset is pointed to row @ 258 initially, and row next pointer is pointed back to self.
Garbage offset is updated to row @ 225, and row next pointer is pointed to previous garbage offset (garbage is added to head of list).

B+Tree Record Update - Smaller



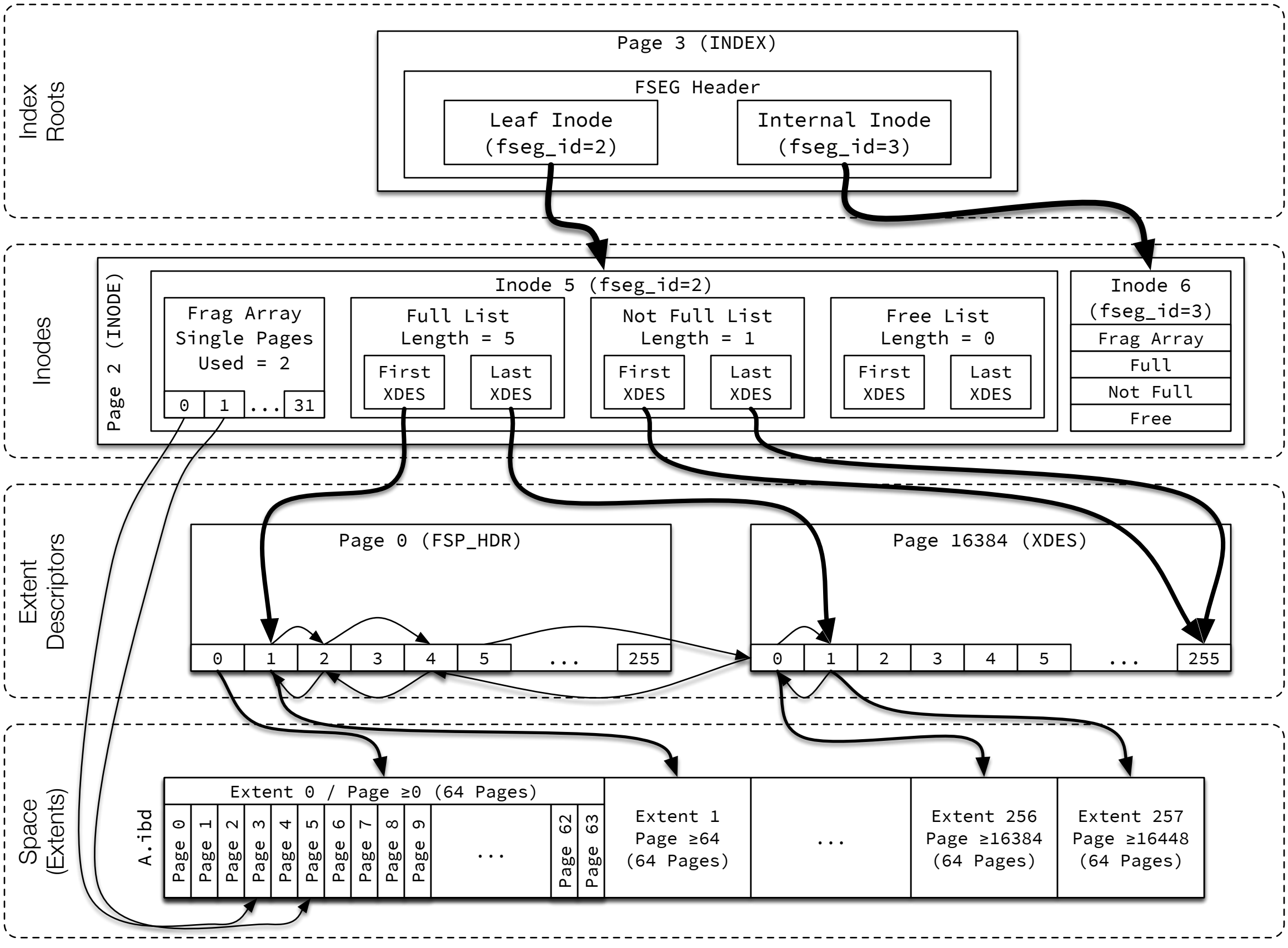
SQL: update t set s="abcde" where i = 5;
Garbage size is incremented by size of row shrinkage, but wasted space is not tracked in garbage list.

B+Tree Record Update - Larger

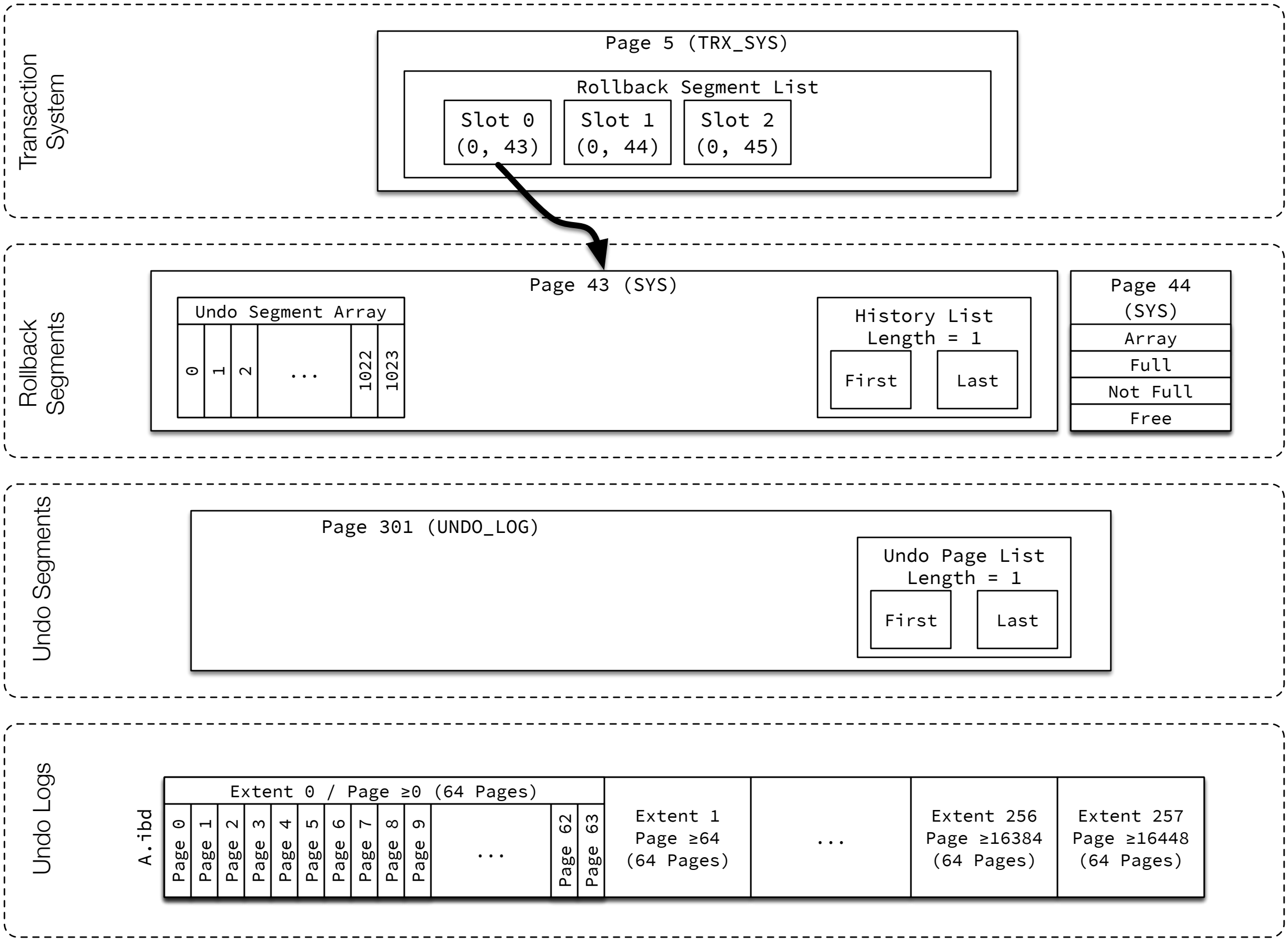


SQL: update t set s="abcdefghijklmno" where i = 5;
Row is deleted, and a new row is inserted into the heap.

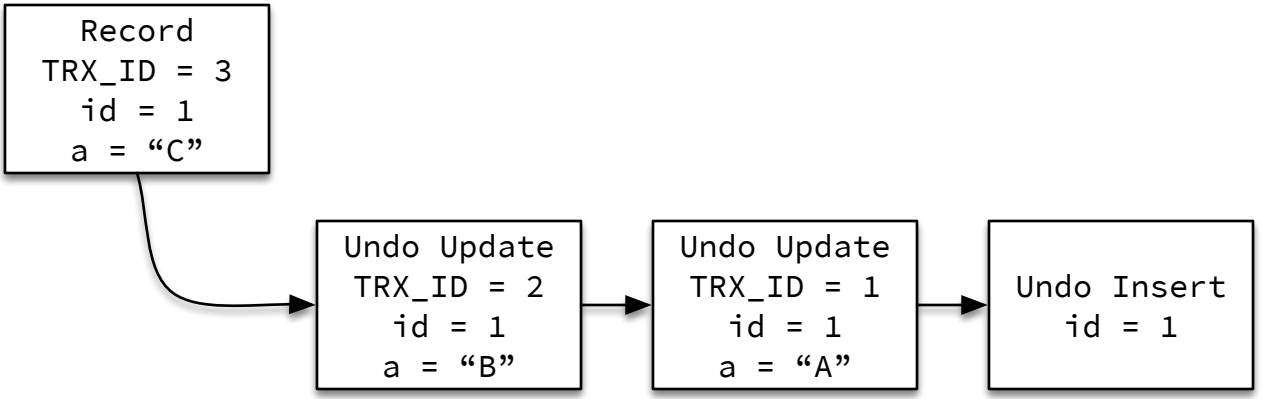
Index File Segment Structure



History Structure



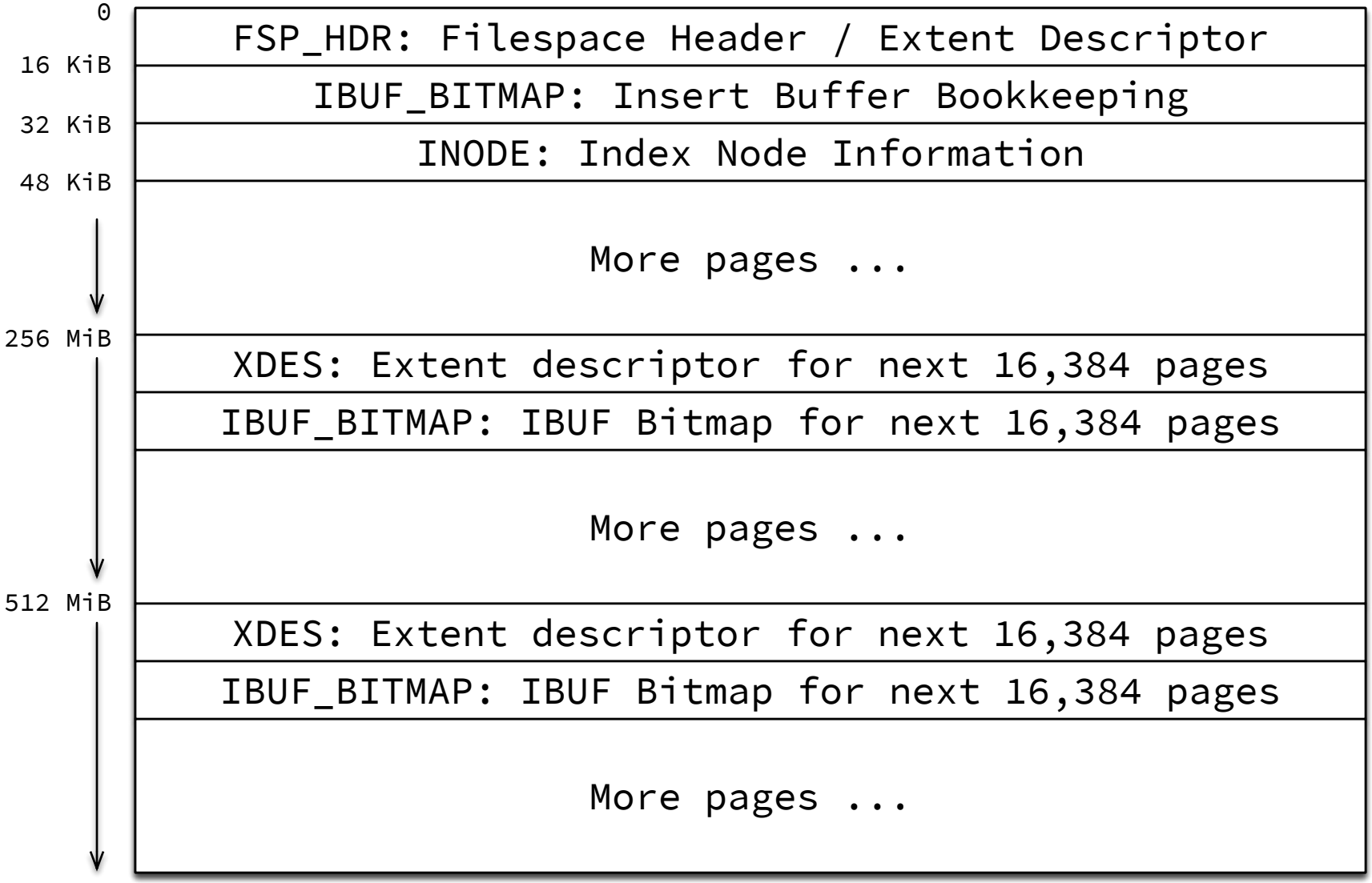
Record History



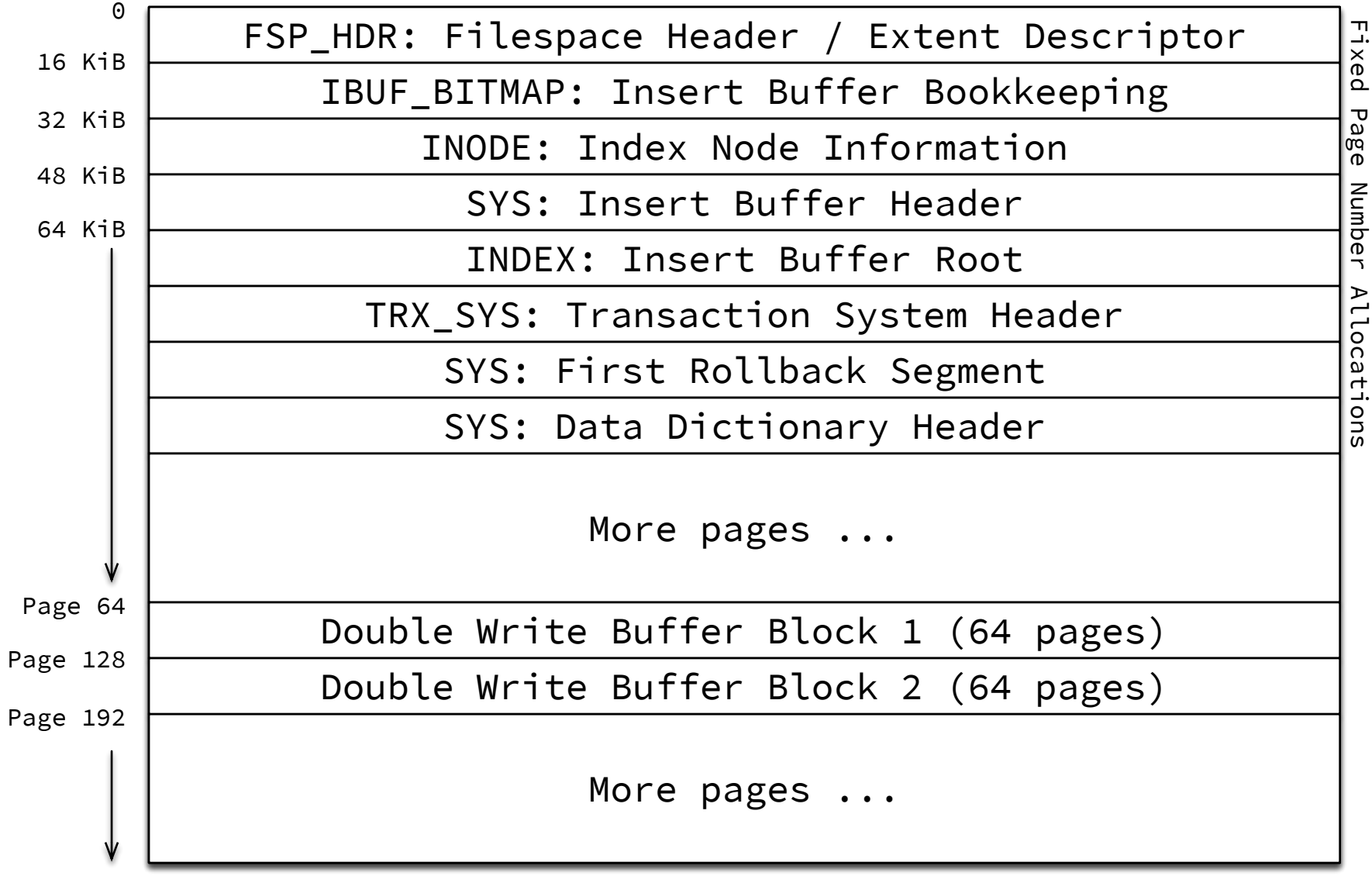
History above represents the following SQL statements:

```
INSERT INTO t (id, a) VALUES (1, “A”);
UPDATE t SET a=“B” WHERE id = 1;
UPDATE t SET a=“C” WHERE id = 1;
```

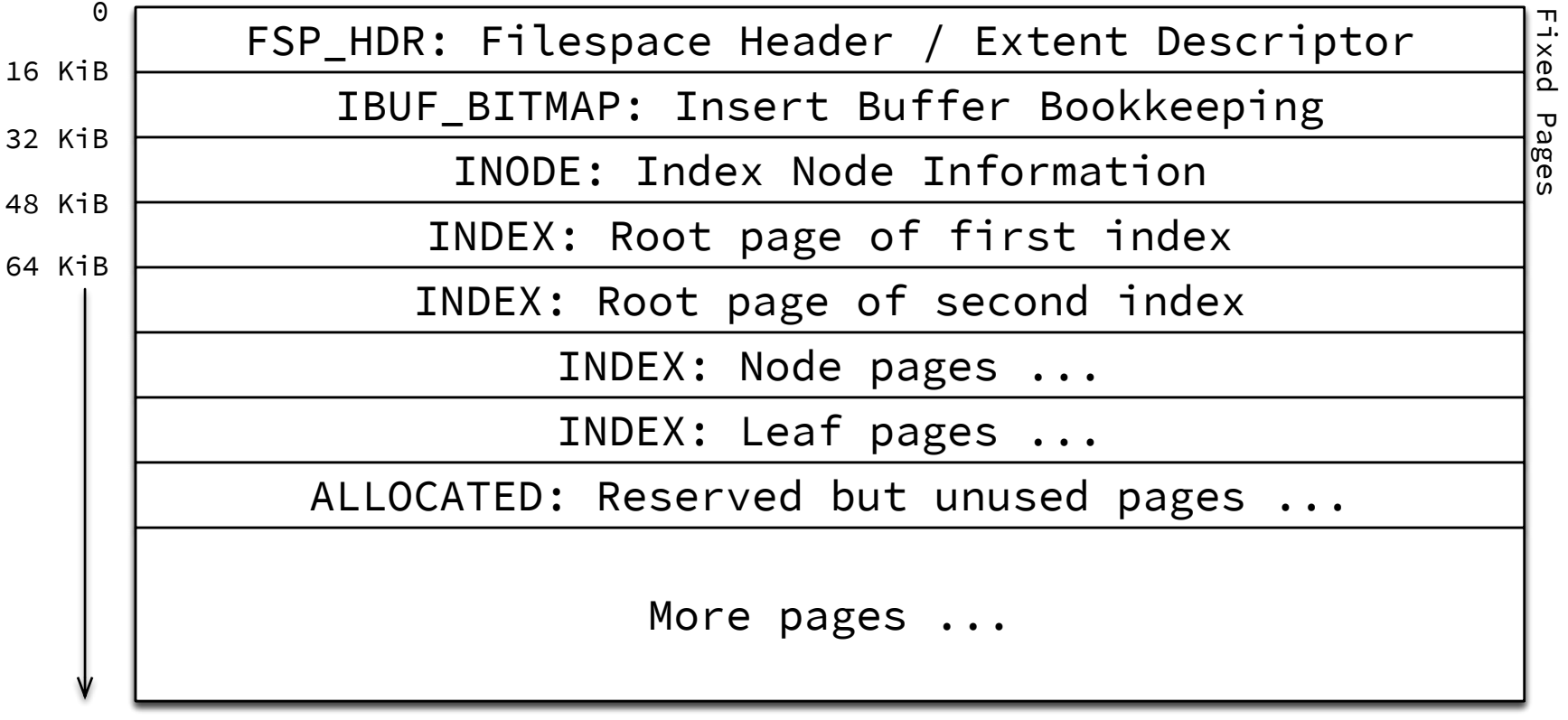
Space File Overview



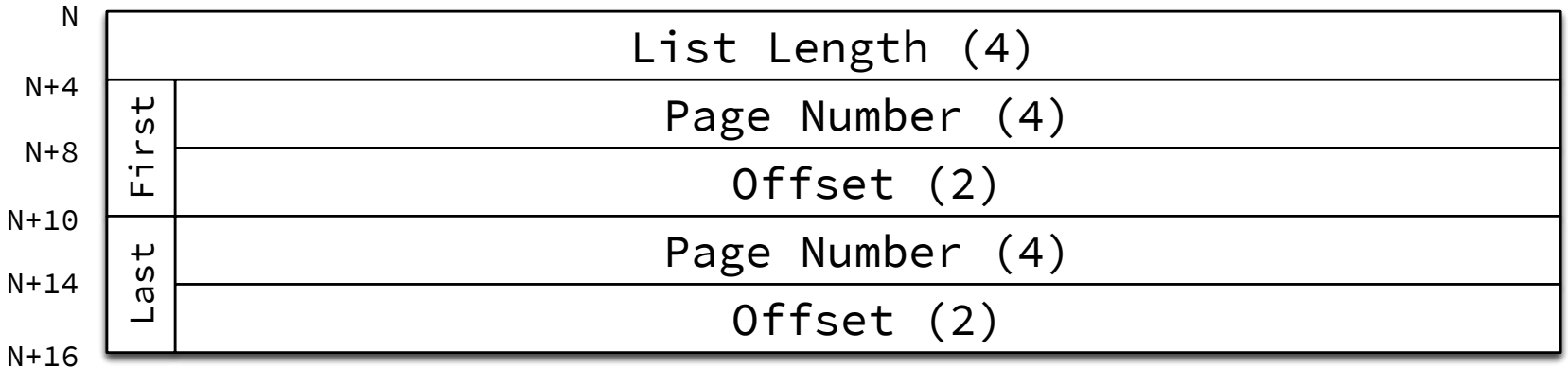
ibdata1 File Overview



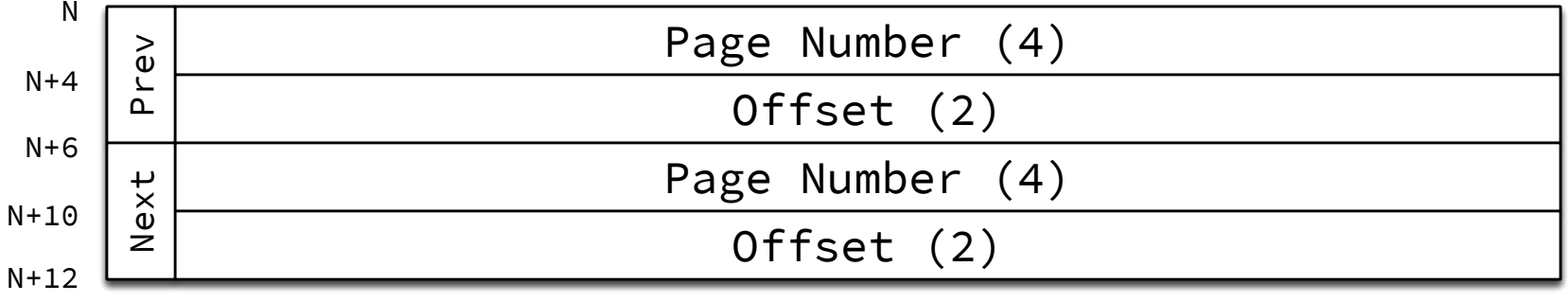
IBD File Overview



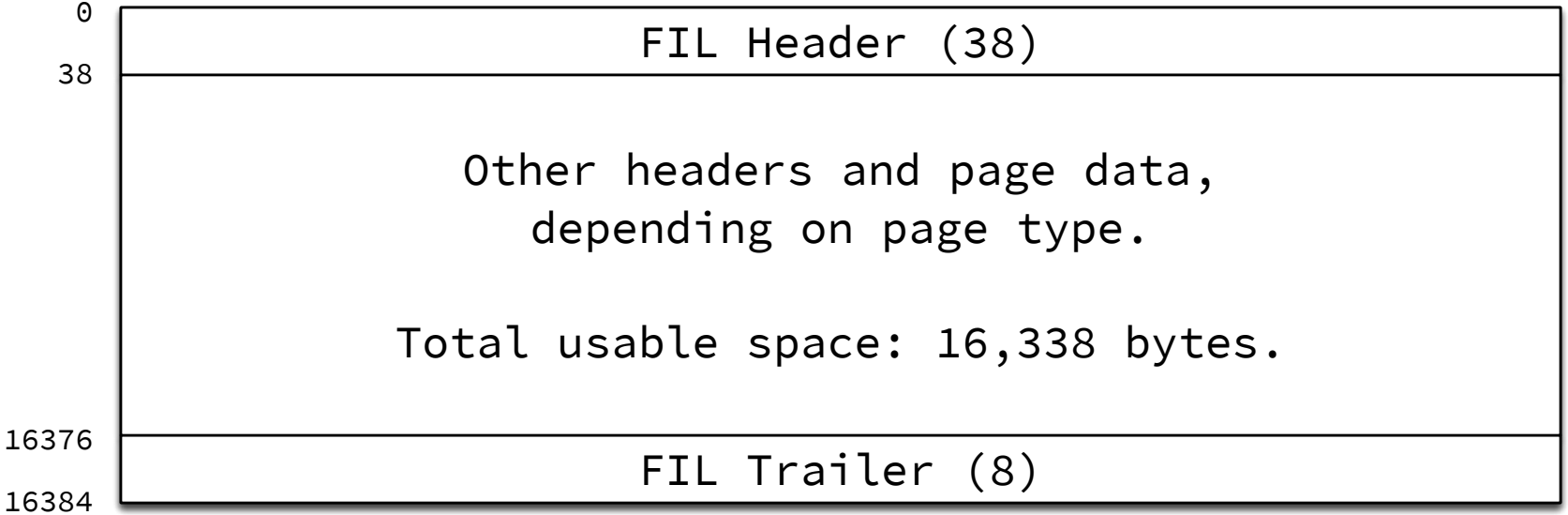
List Base Node



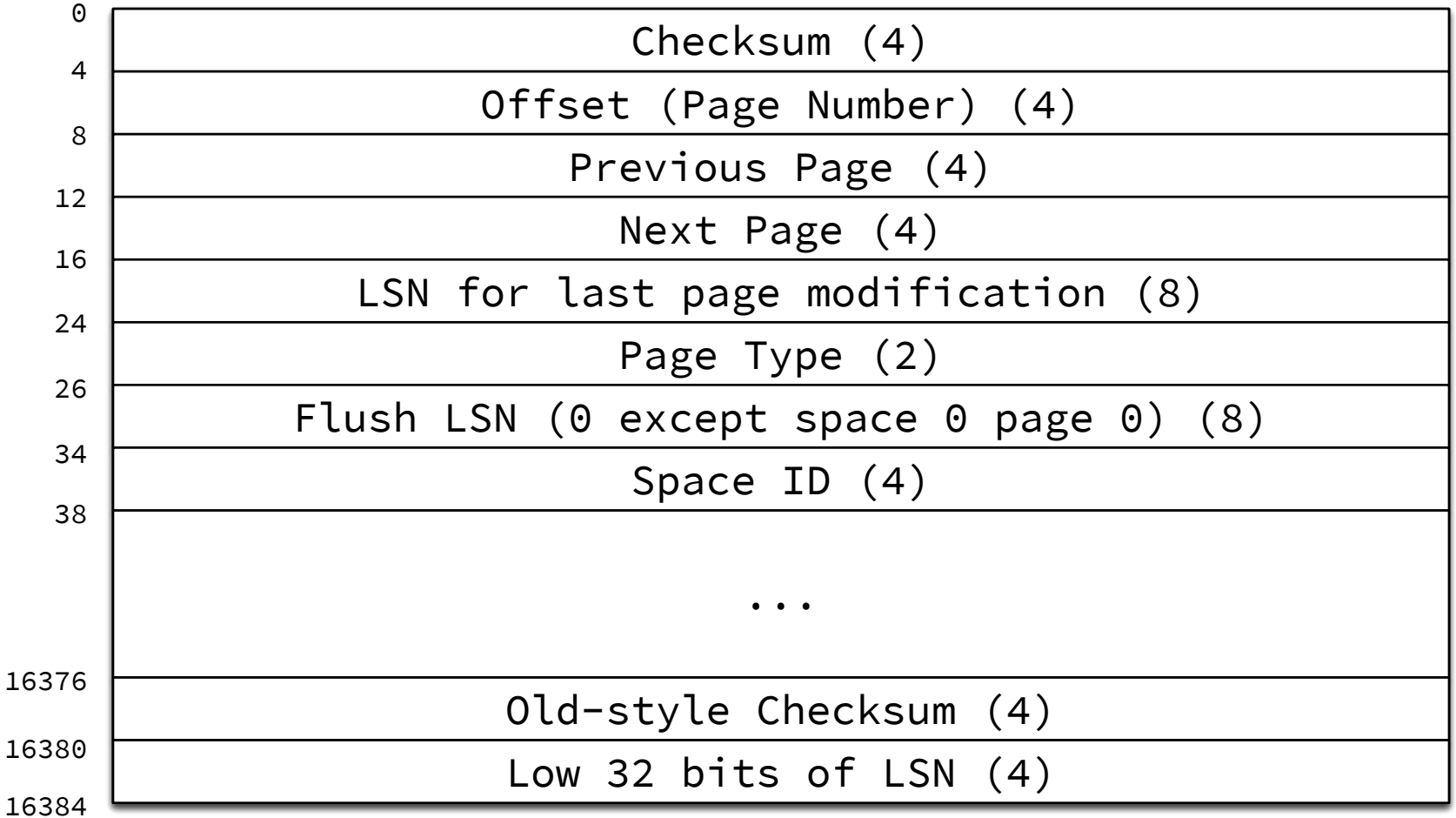
List Node


N
N+4
N+6
N+10
N+12
Prev
Next
Page Number (4)
Offset (2)
Page Number (4)
Offset (2)

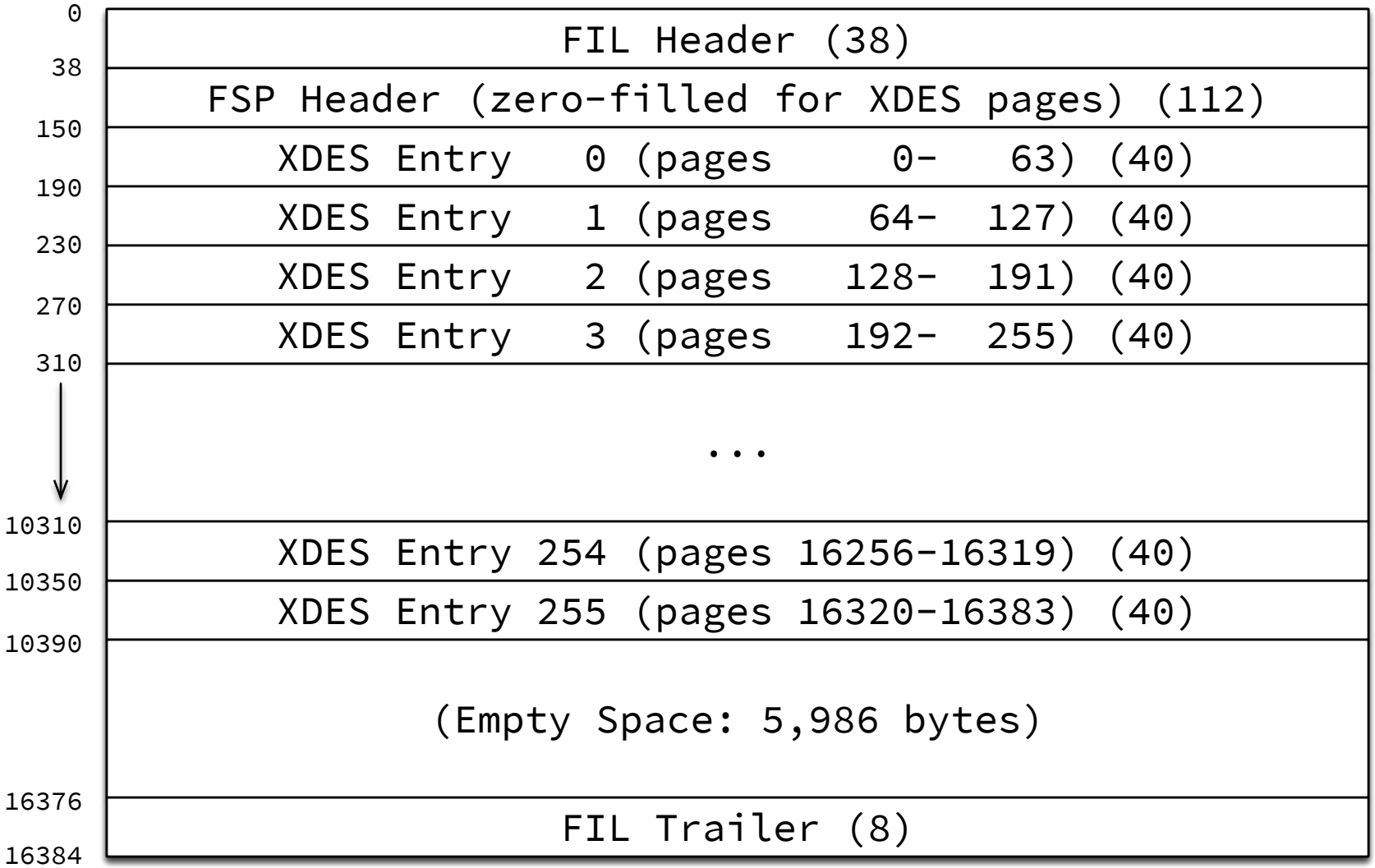
Basic Page Overview



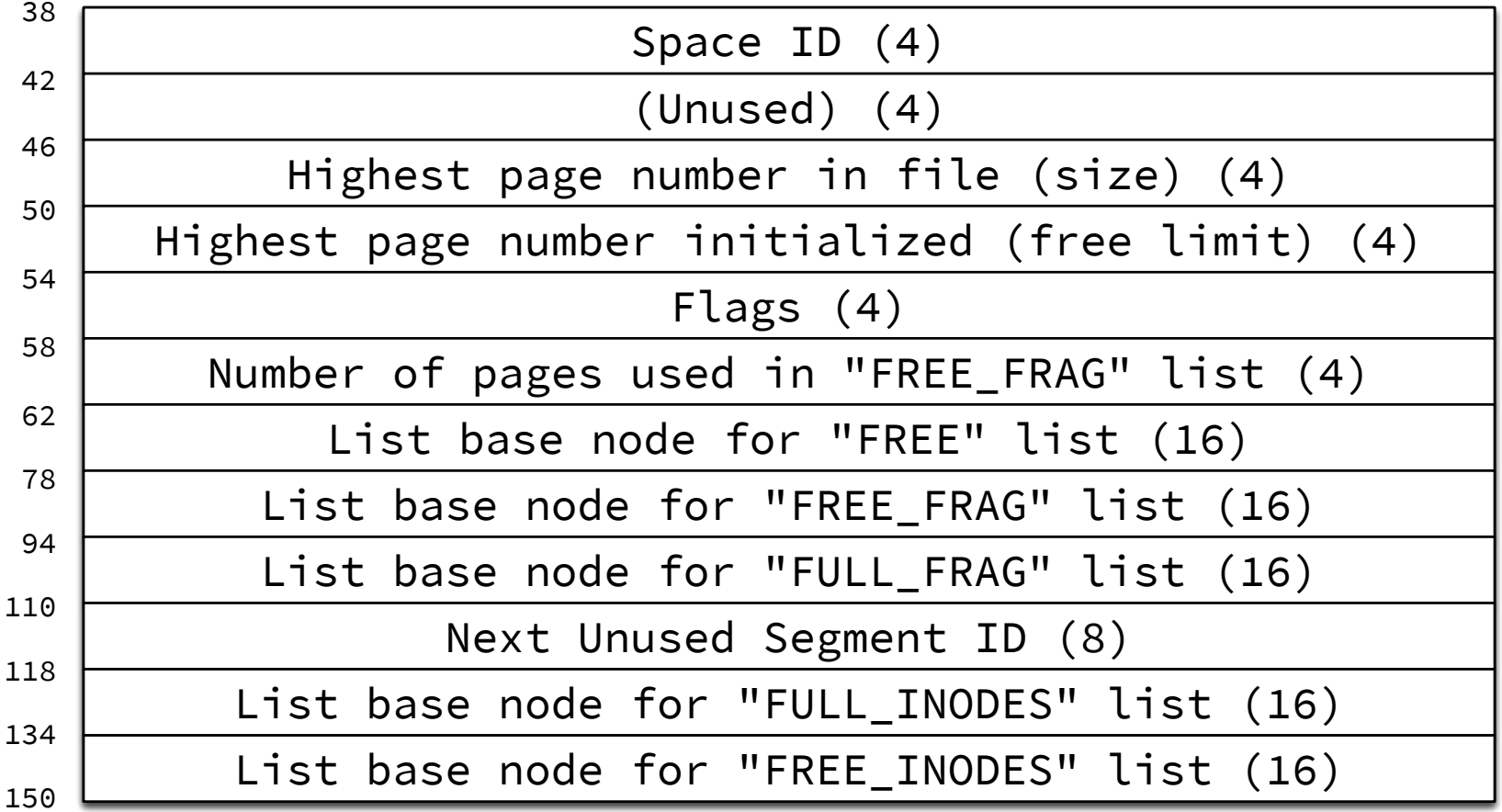
FIL Header/Trailer



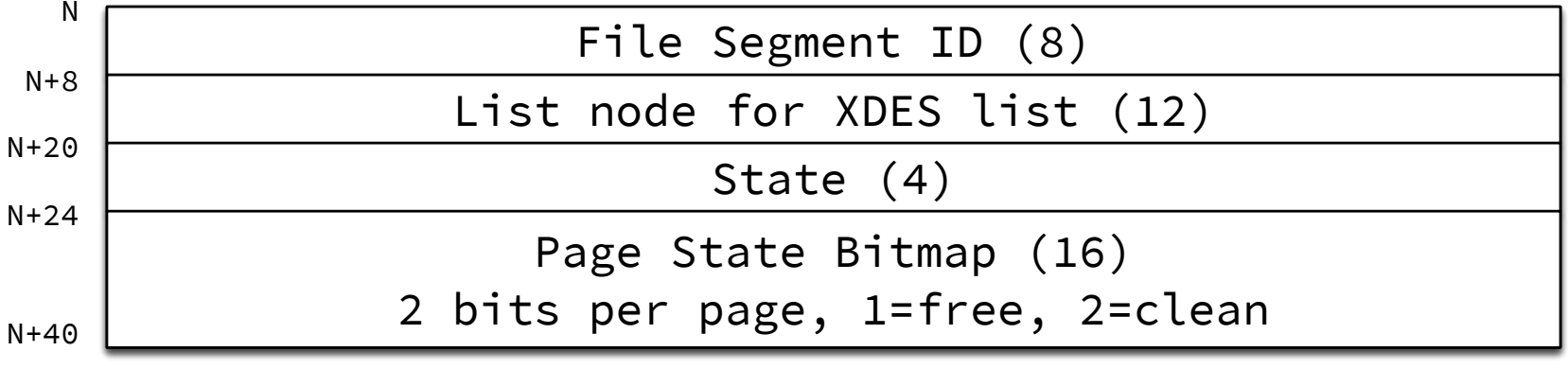
FSP_HDR/XDES Overview



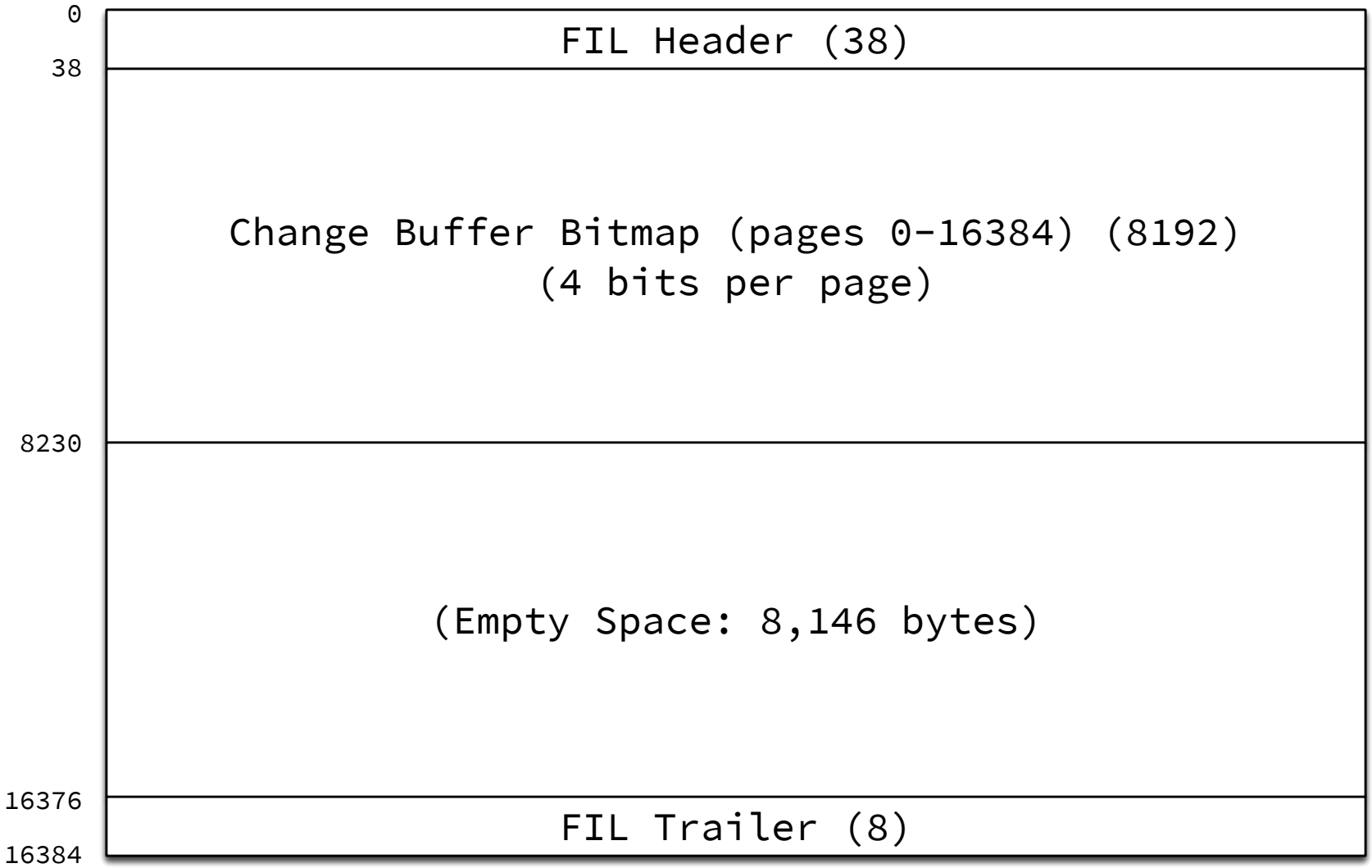
FSP Header


38
Space ID (4)
42
(Unused) (4)
46
Highest page number in file (size) (4)
50
Highest page number initialized (free limit) (4)
54
Flags (4)
58
Number of pages used in "FREE_FRAG" list (4)
62
List base node for "FREE" list (16)
78
List base node for "FREE_FRAG" list (16)
94
List base node for "FULL_FRAG" list (16)
110
Next Unused Segment ID (8)
118
List base node for "FULL_INODES" list (16)
134
List base node for "FREE_INODES" list (16)
150

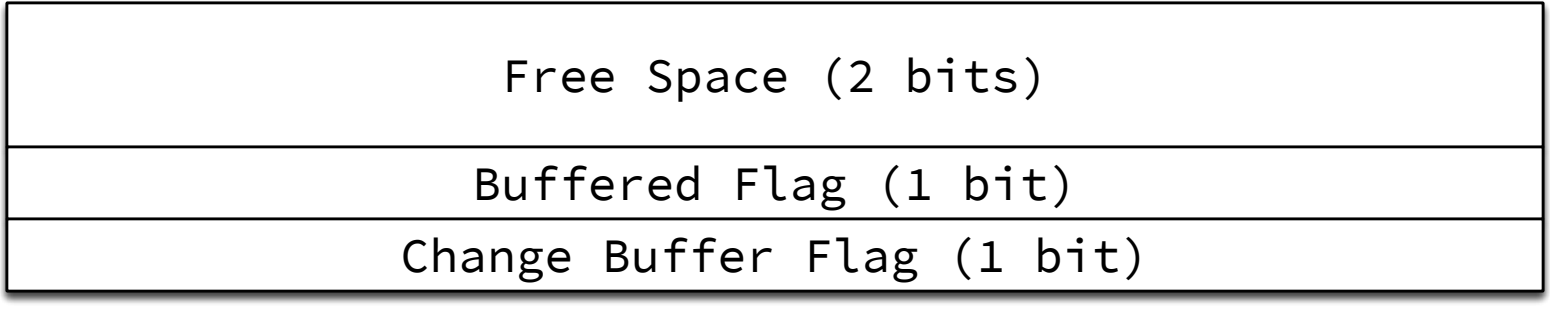
XDES Entry


N
File Segment ID (8)
N+8
List node for XDES list (12)
N+20
State (4)
N+24
Page State Bitmap (16)
2 bits per page, 1=free, 2=clean
N+40

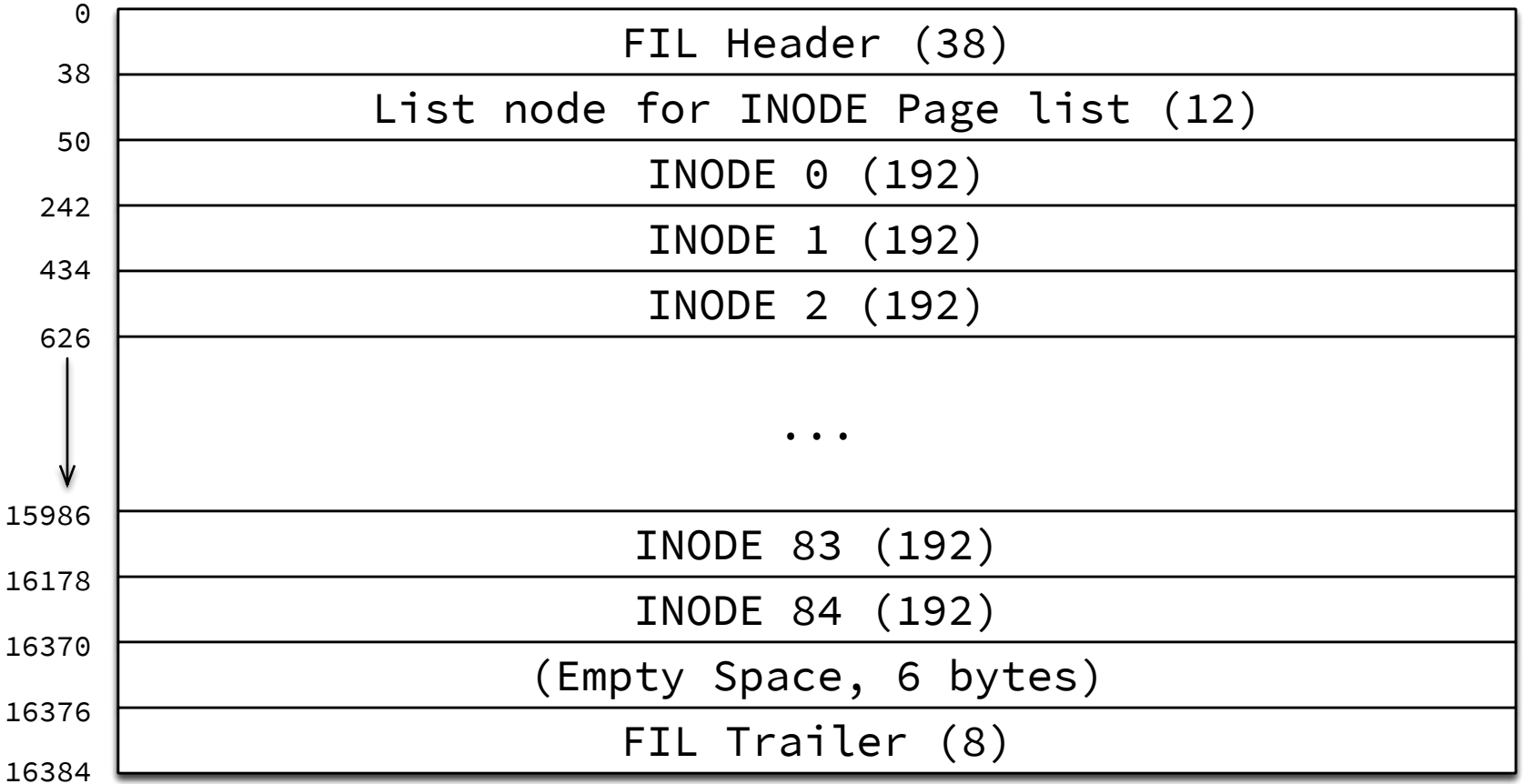
IBUF_BITMAP Overview



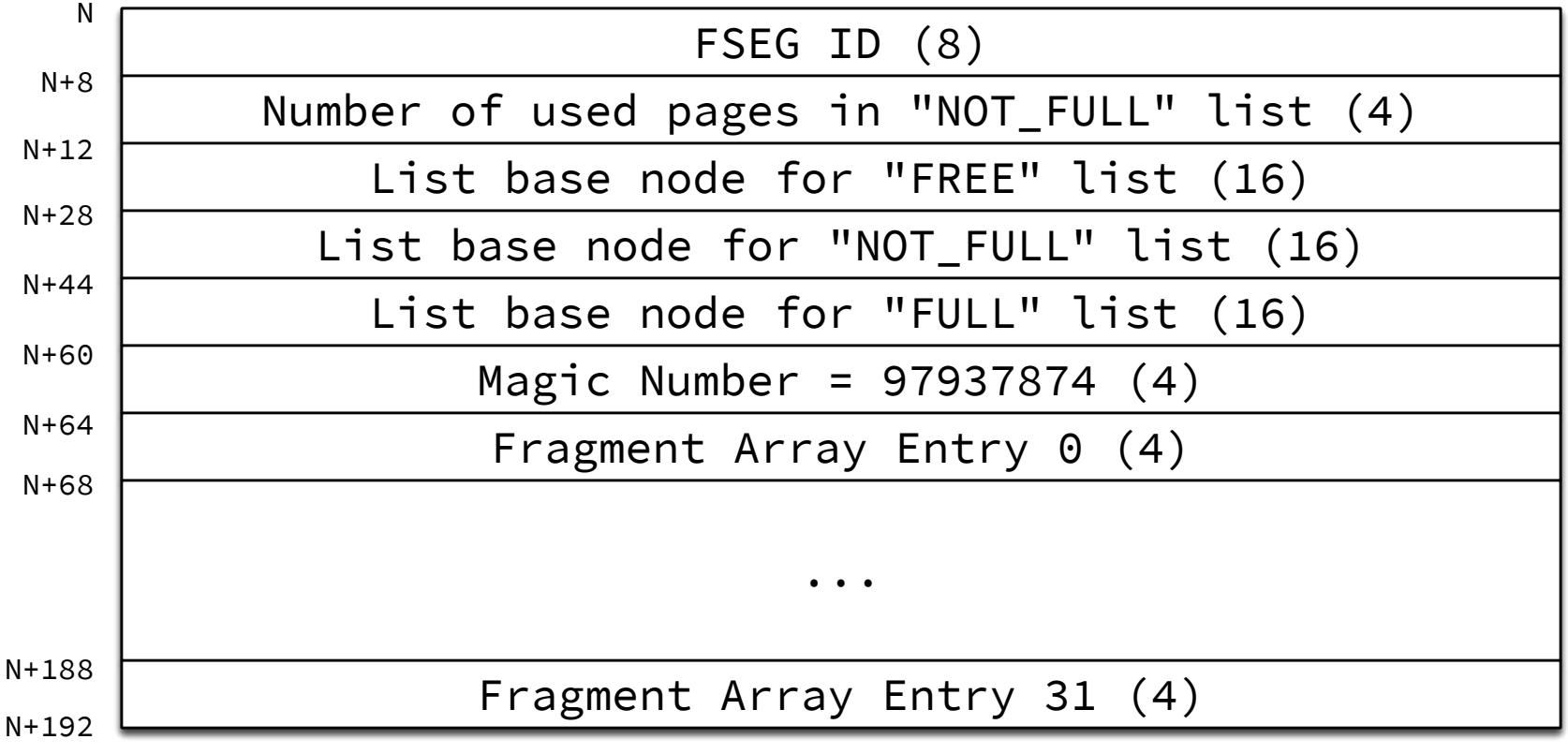
IBUF_BITMAP Page Entry



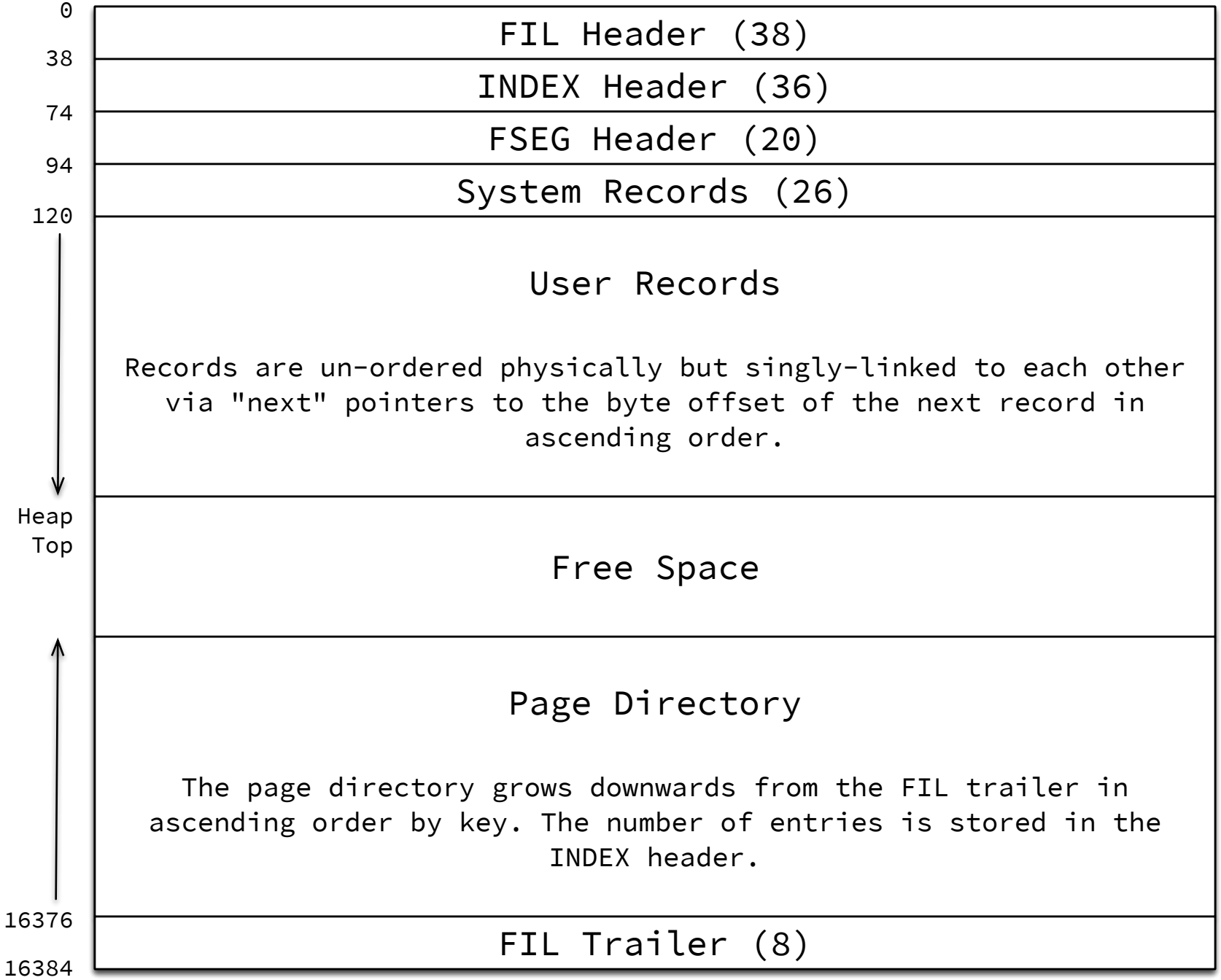
INODE Overview



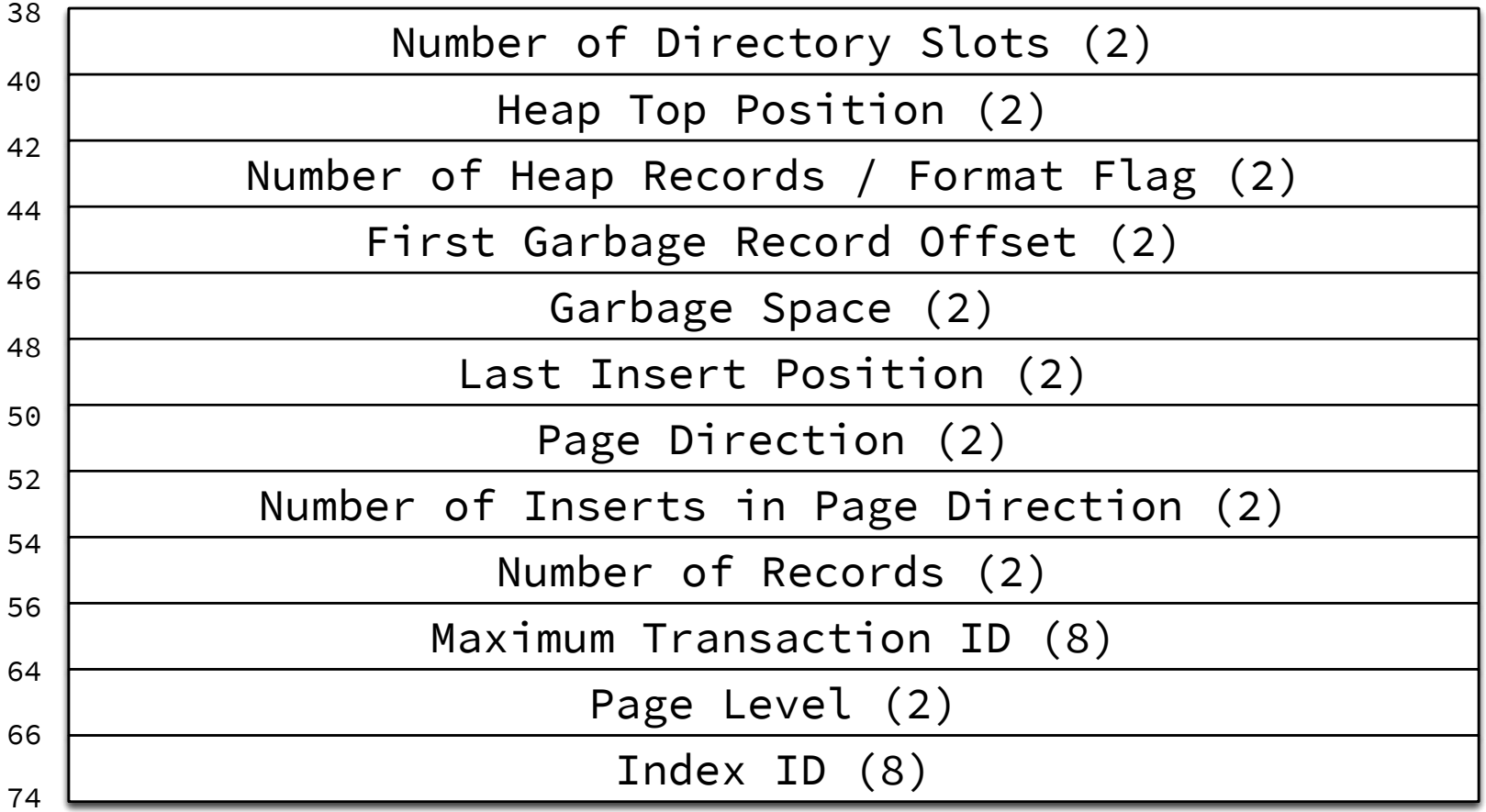
INODE Entry


N
FSEG ID (8)
N+8
Number of used pages in "NOT_FULL" list (4)
N+12
List base node for "FREE" list (16)
N+28
List base node for "NOT_FULL" list (16)
N+44
List base node for "FULL" list (16)
N+60
Magic Number = 97937874 (4)
N+64
Fragment Array Entry 0 (4)
N+68
...
N+188
Fragment Array Entry 31 (4)
N+192

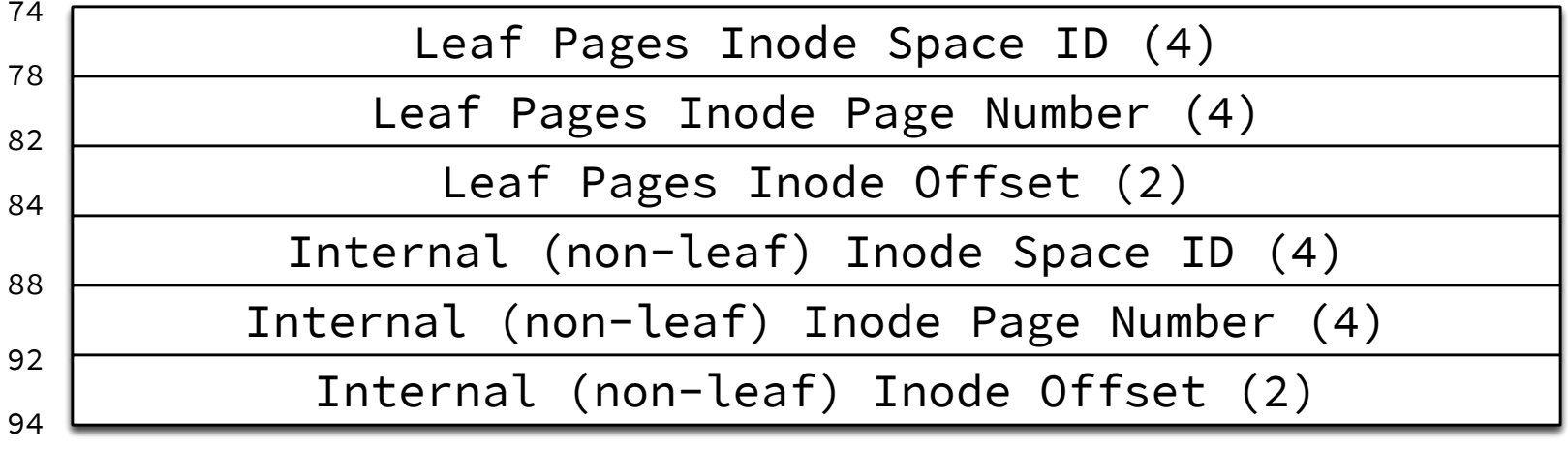
INDEX Overview



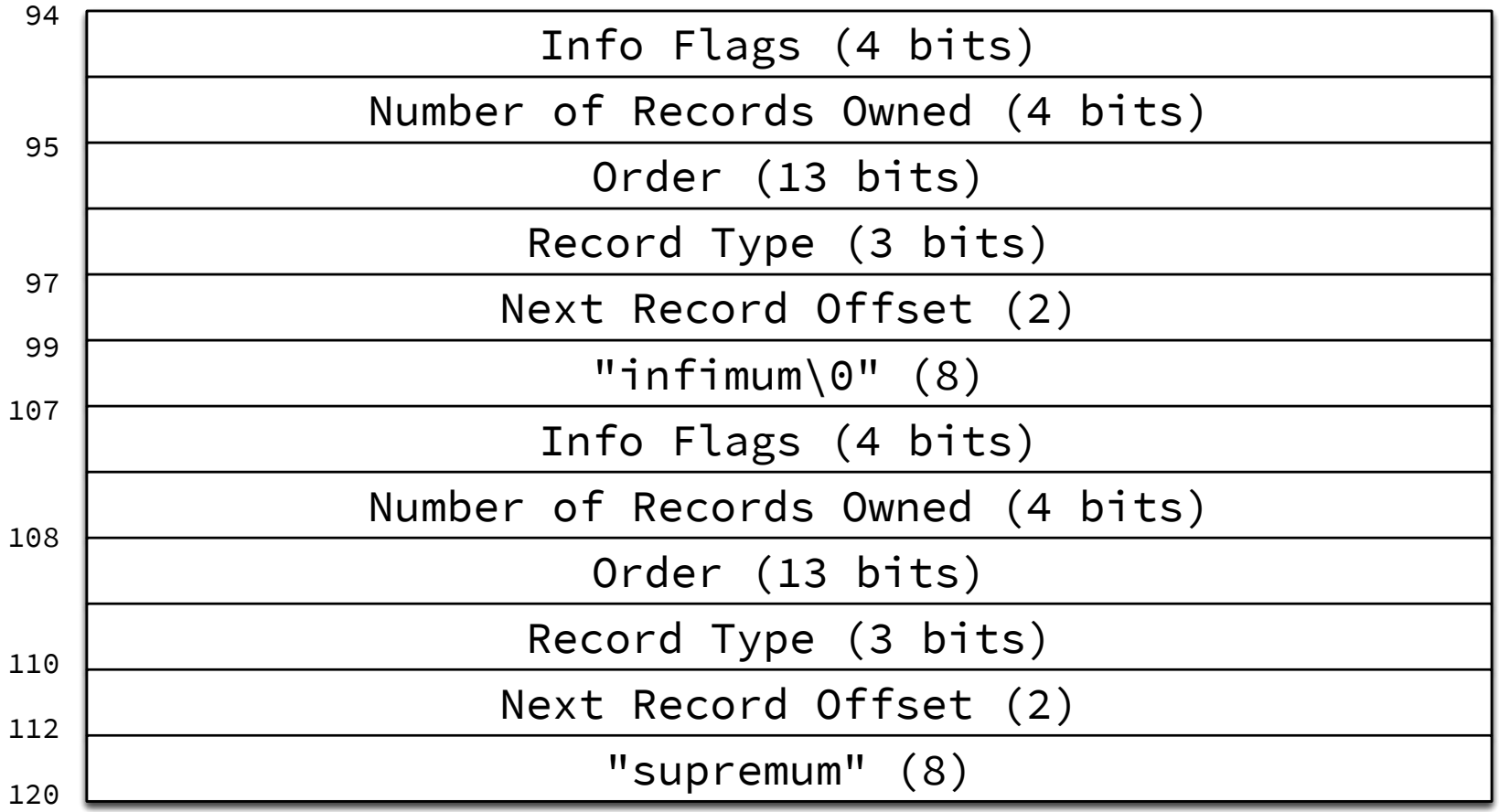
INDEX Header



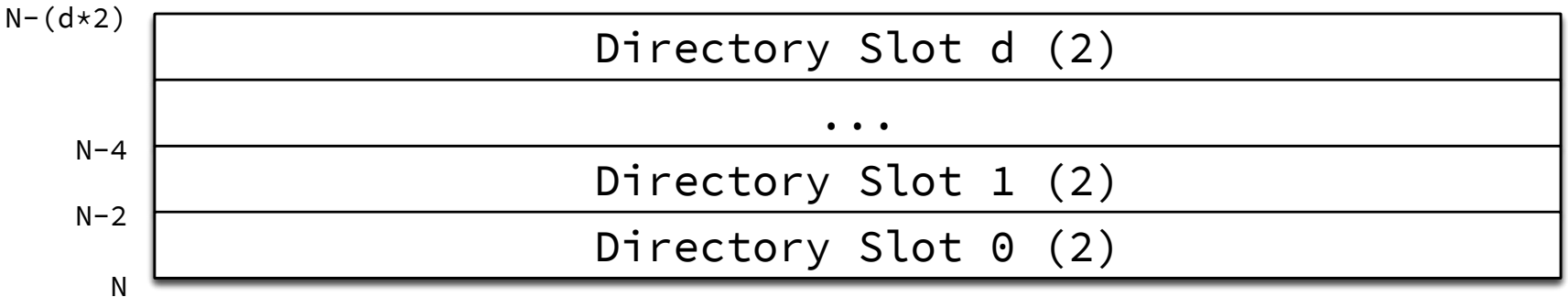
FSEG Header


74
78
82
84
88
92
94
Leaf Pages Inode Space ID (4)
Leaf Pages Inode Page Number (4)
Leaf Pages Inode Offset (2)
Internal (non-leaf) Inode Space ID (4)
Internal (non-leaf) Inode Page Number (4)
Internal (non-leaf) Inode Offset (2)

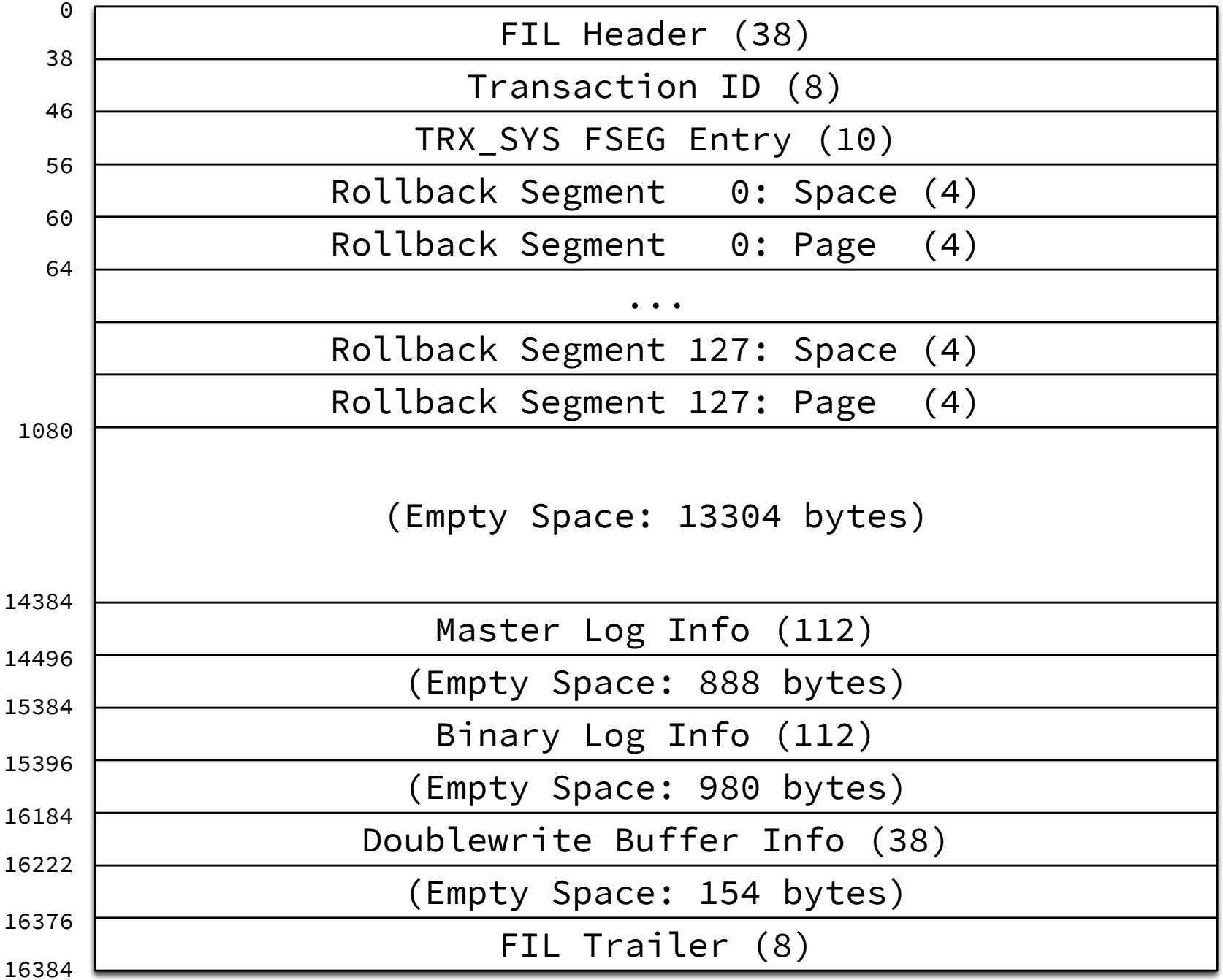
INDEX System Records



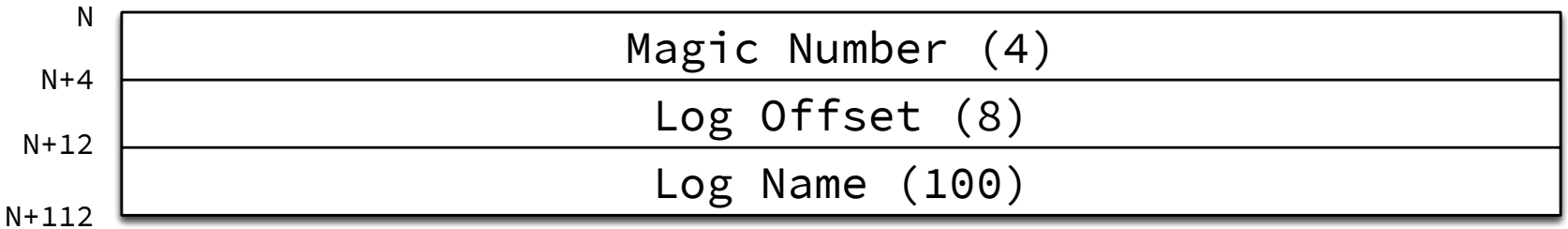
INDEX Page Directory



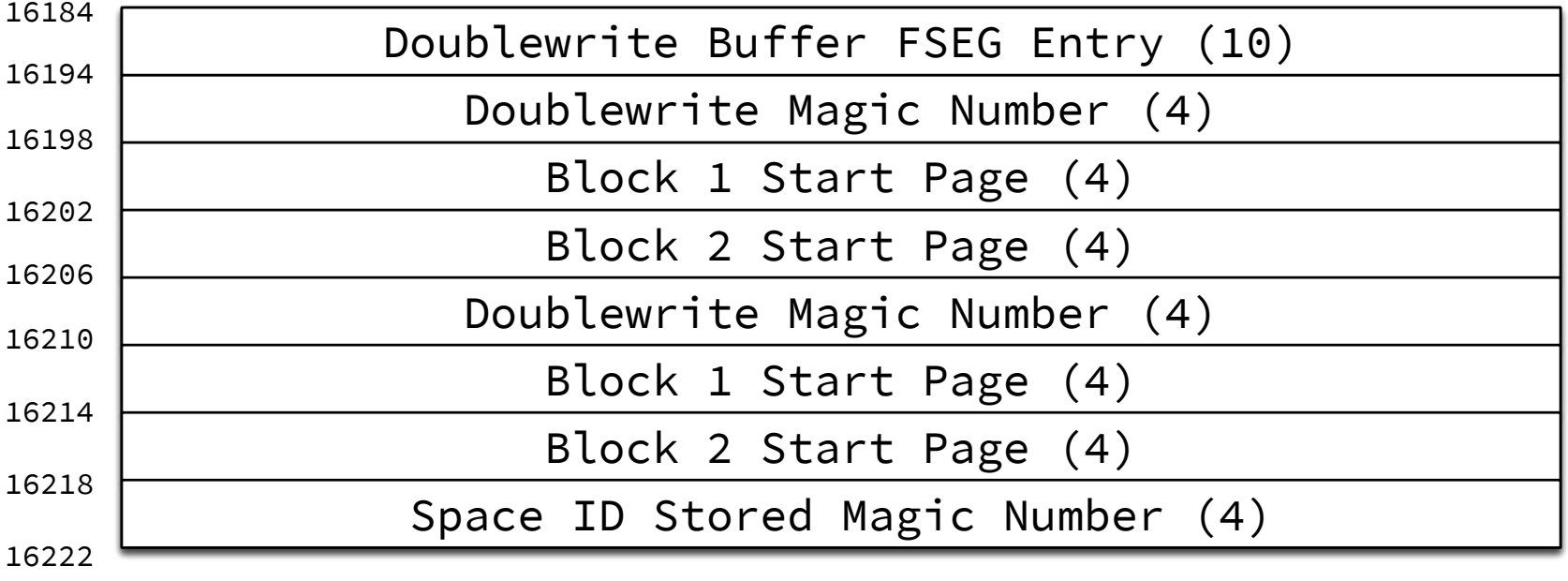
TRX_SYS Overview



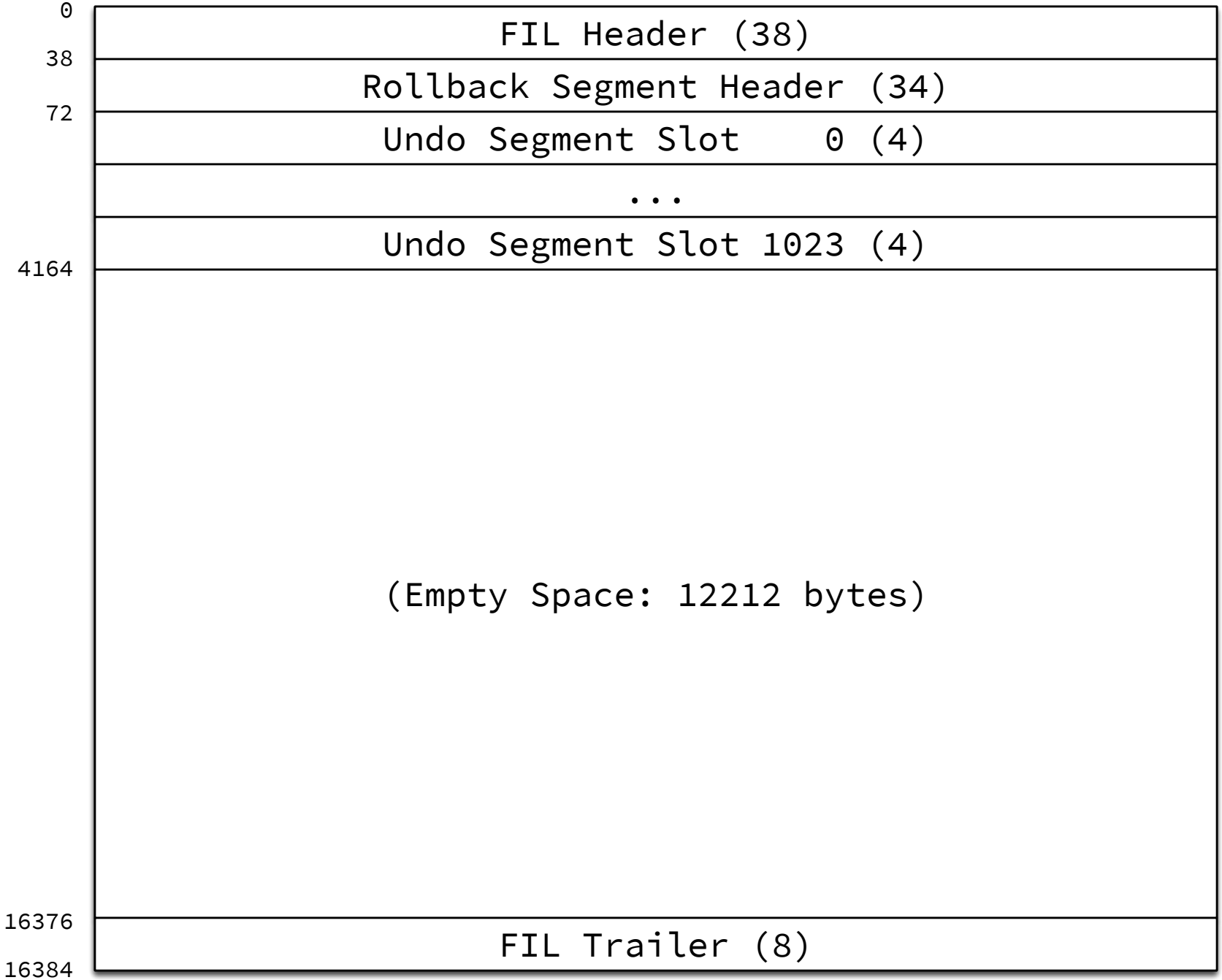
TRX_SYS MySQL Log Info



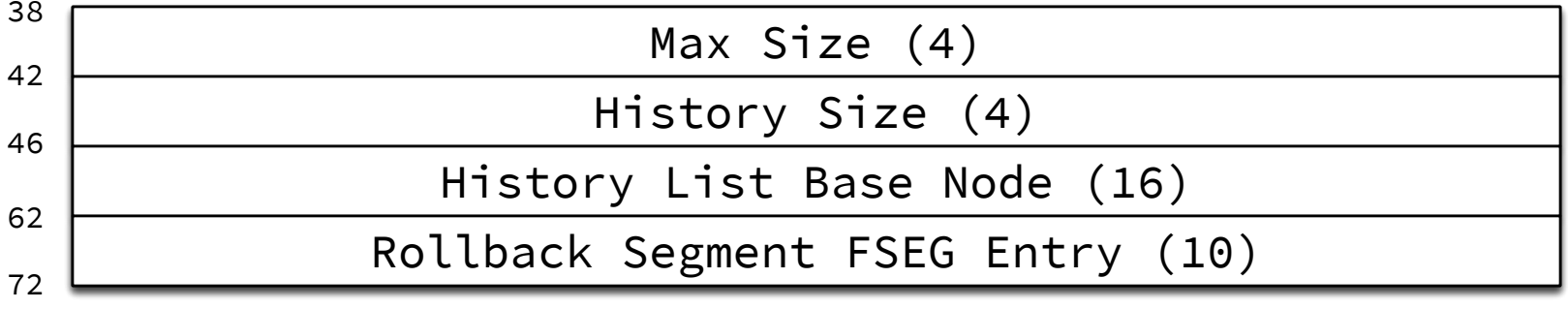
TRX_SYS Doublewrite Buffer Info



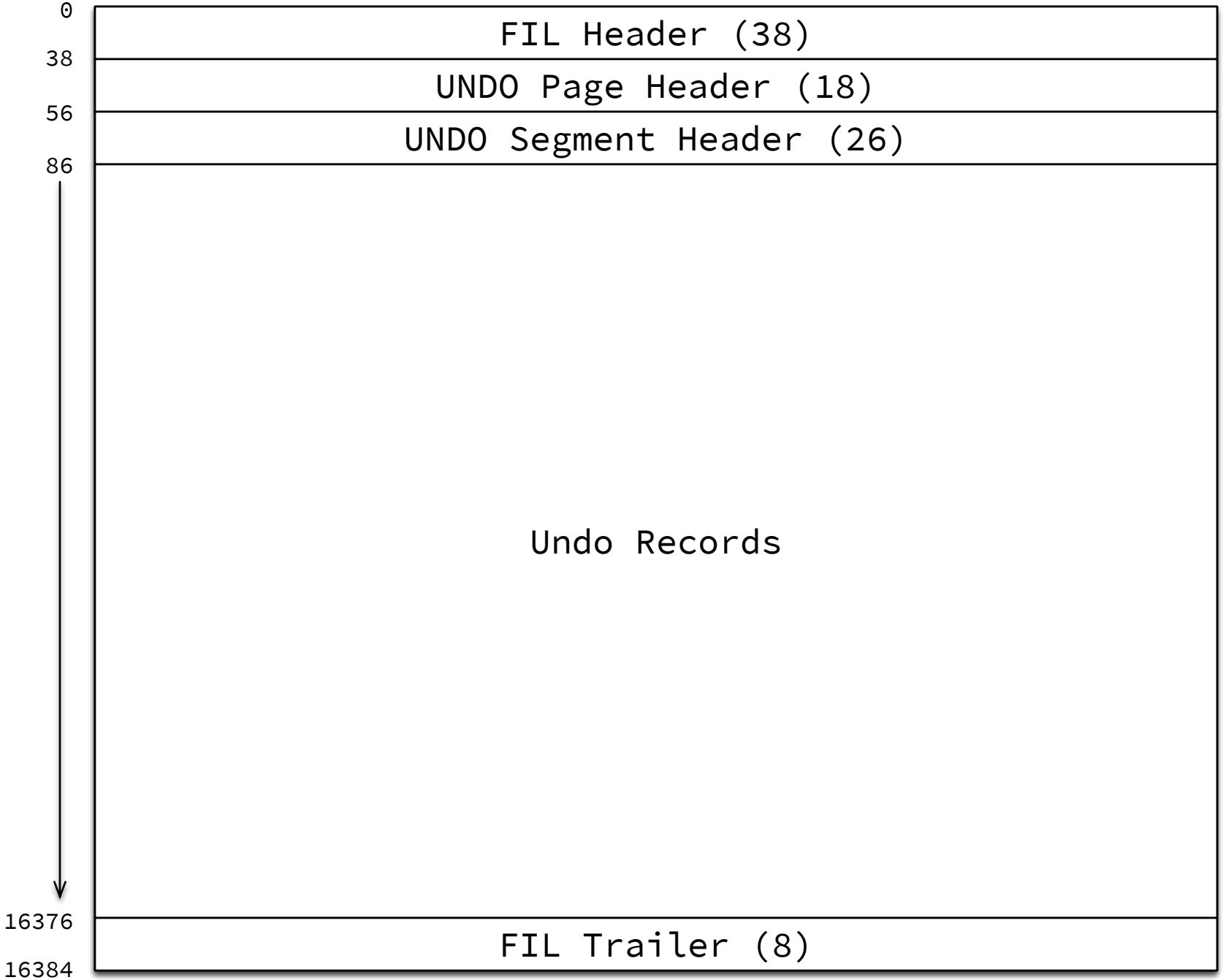
SYS_RSEG_HEADER Overview



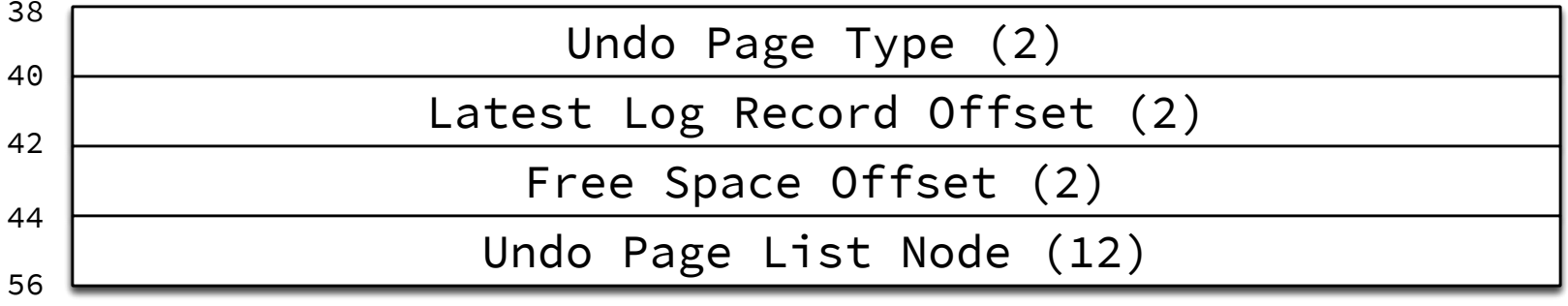
Rollback Segment Header



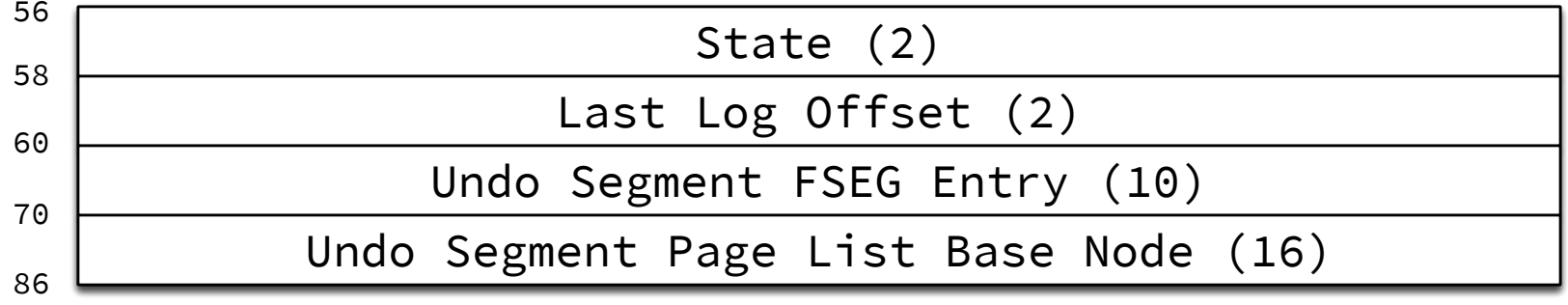
UNDO_LOG Overview



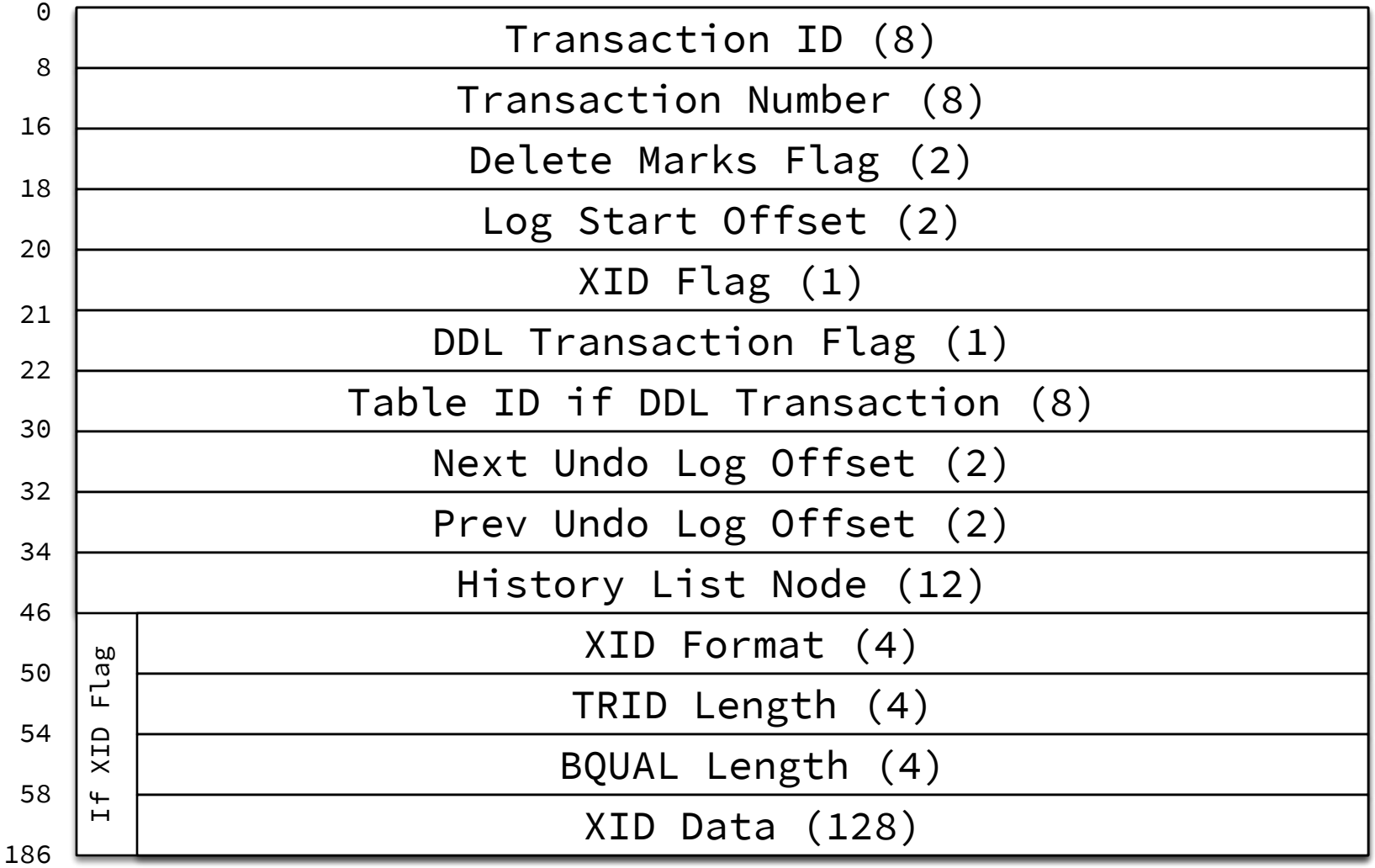
Undo Page Header



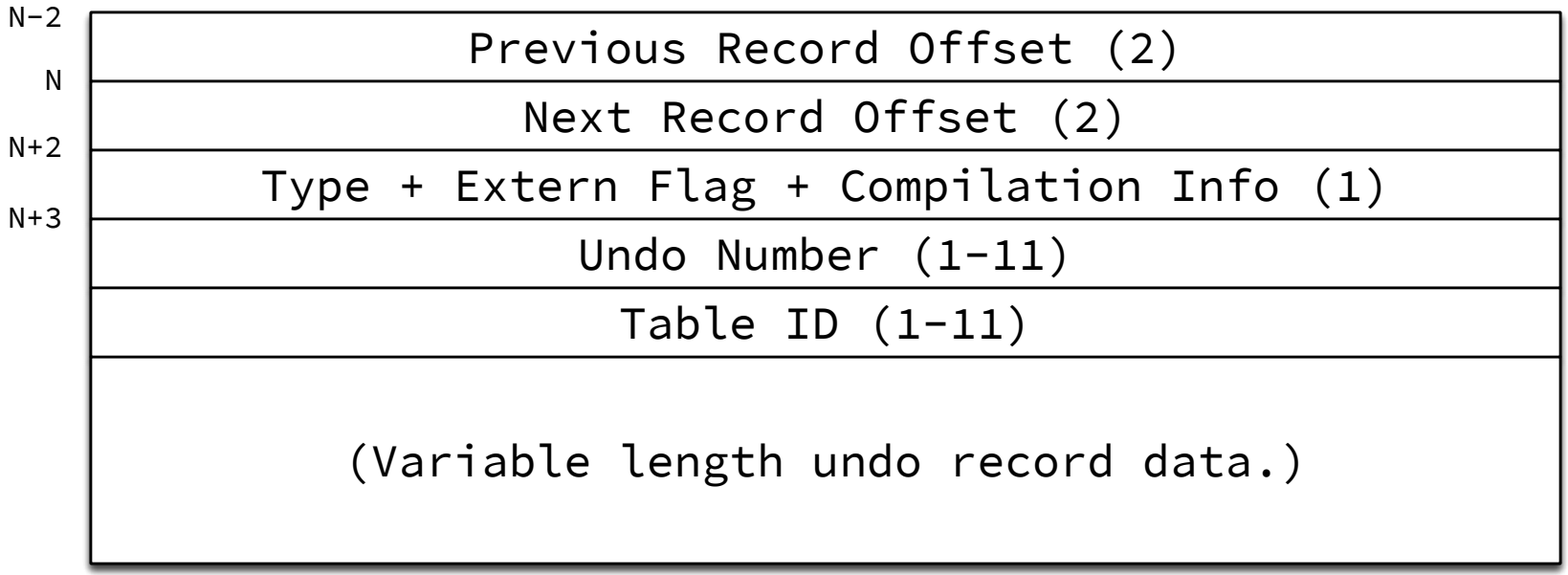
Undo Segment Header



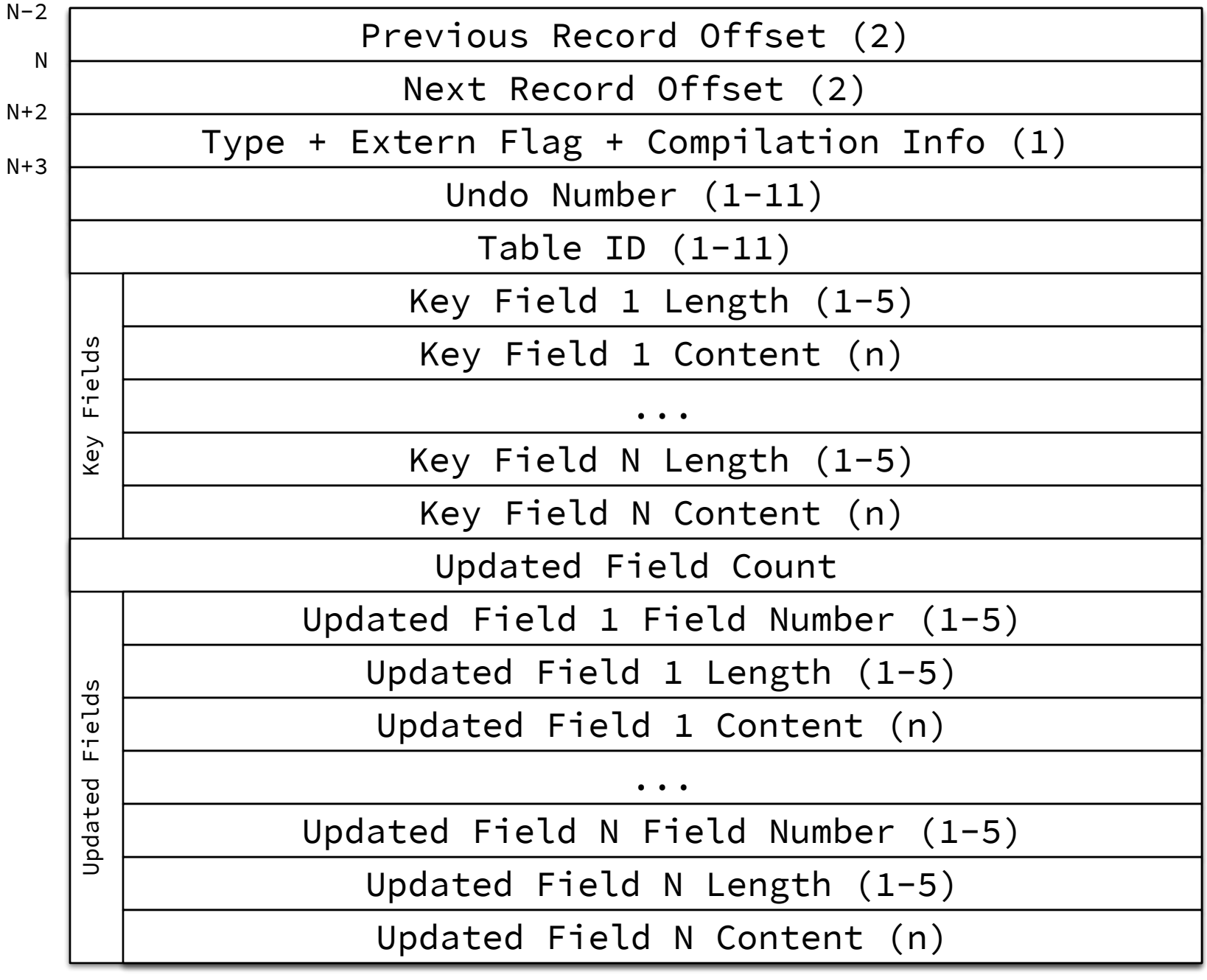
Undo Log Header

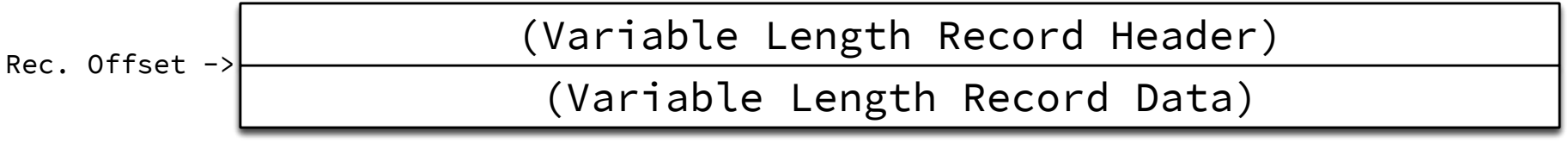


Undo Record

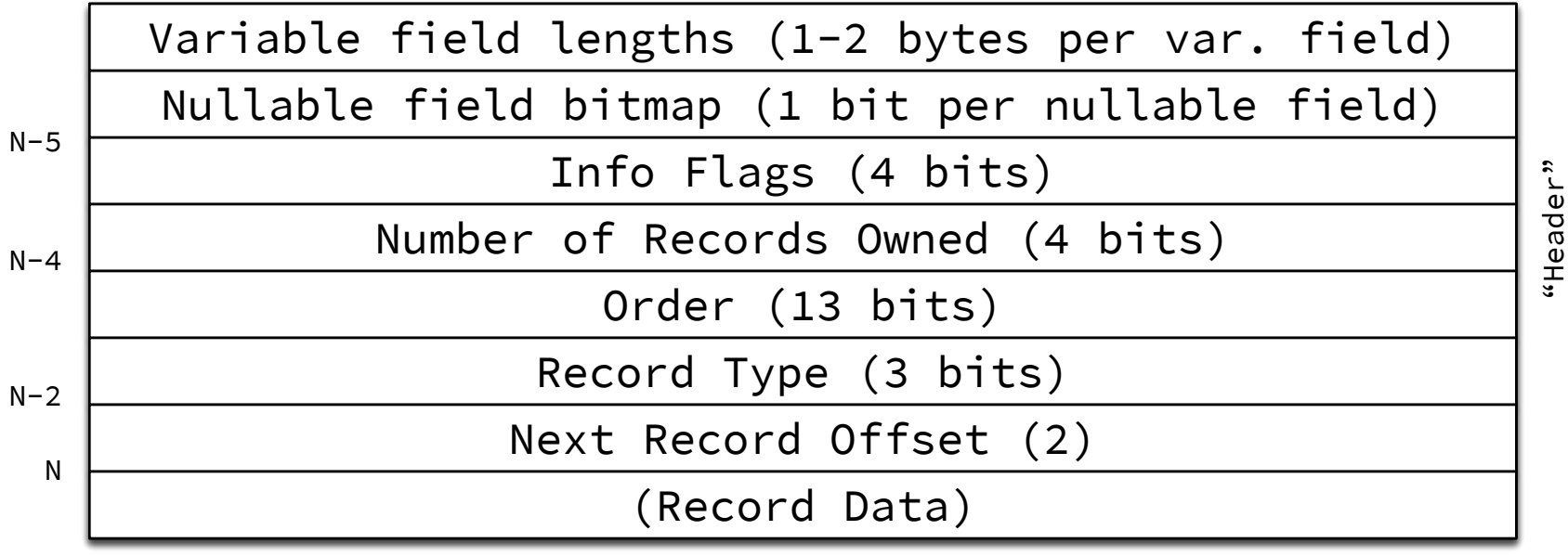

N-2
N
N+2
N+3
Previous Record Offset (2)
Next Record Offset (2)
Type + Extern Flag + Compilation Info (1)
Undo Number (1-11)
Table ID (1-11)
(Variable length undo record data.)

Undo Record for Update

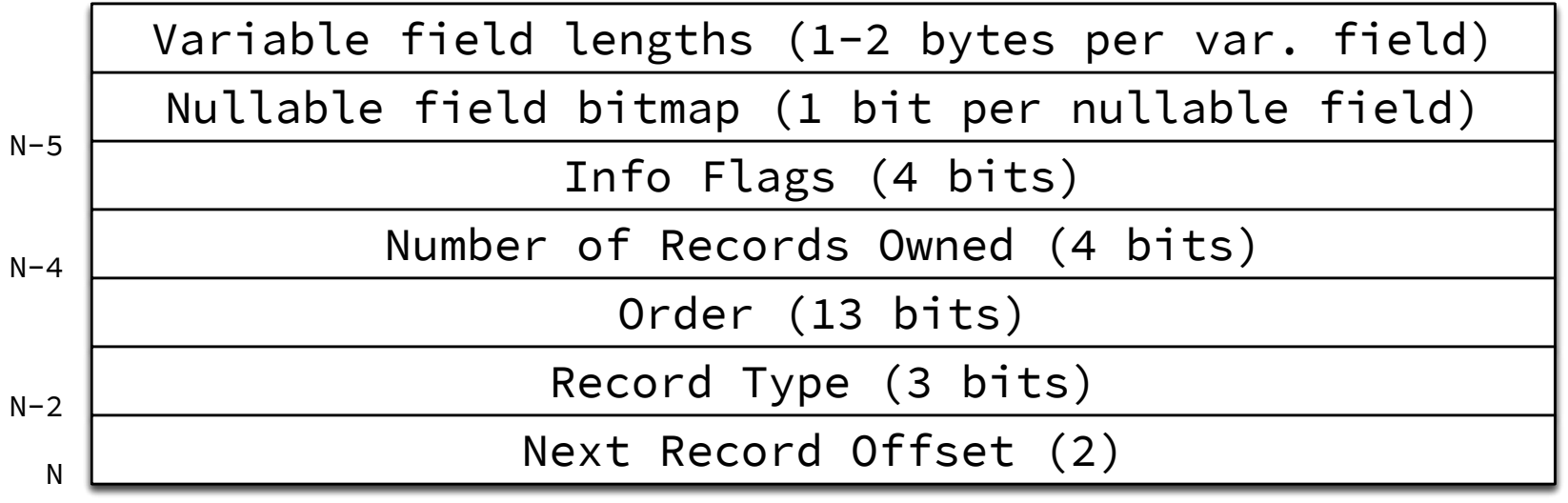



Rec. Offset ->
(Variable Length Record Header)
(Variable Length Record Data)

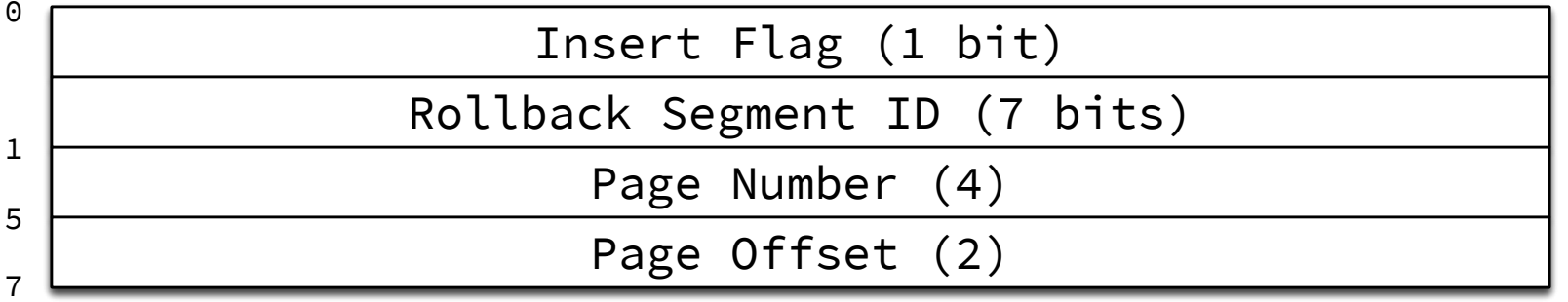
Record Format - Overview



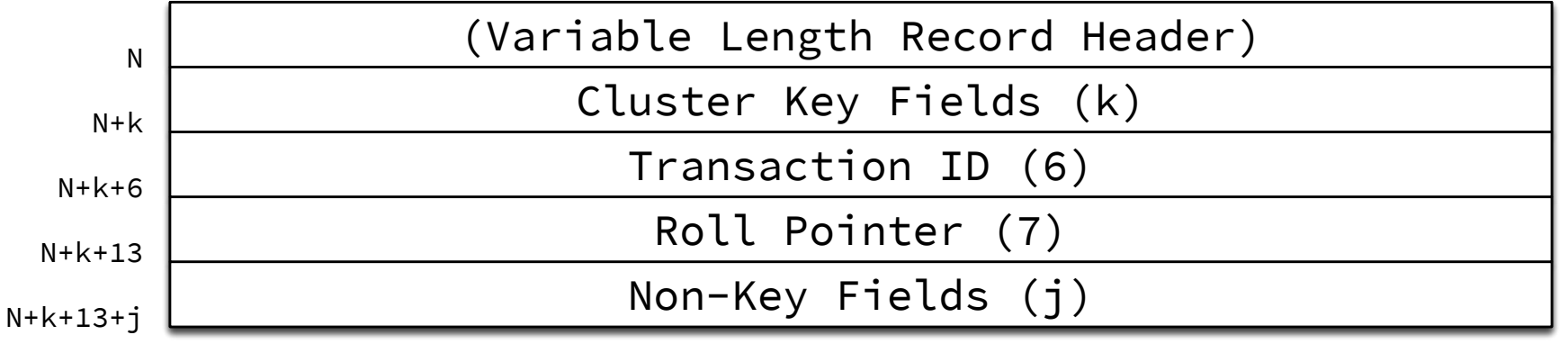
Record Format - Header



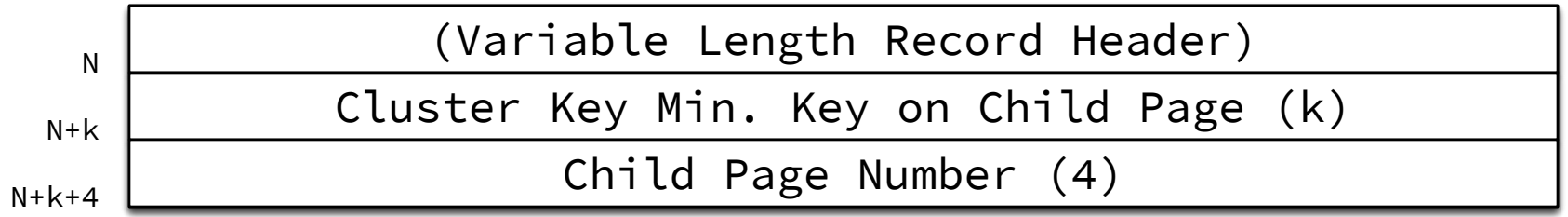
Rollback Pointer (ROLL_PTR)



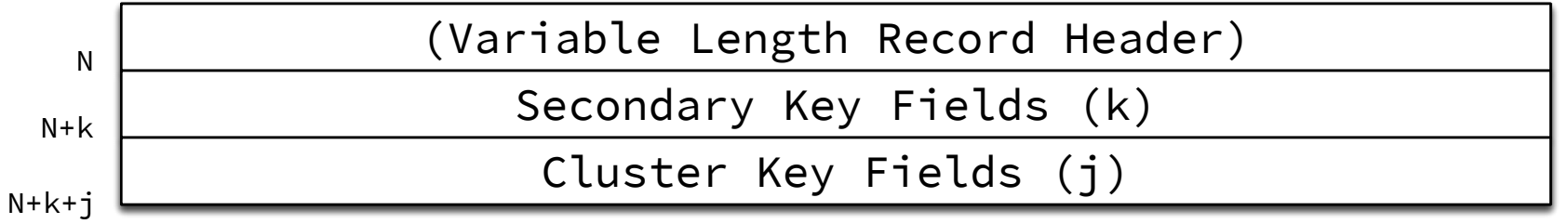
Record Format - Clustered Key - Leaf Pages



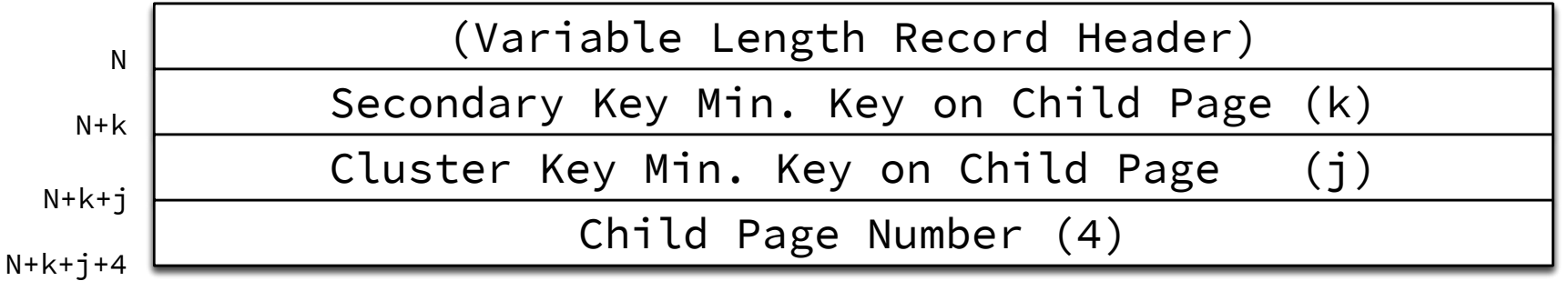
Record Format - Clustered Key - Non-Leaf Pages



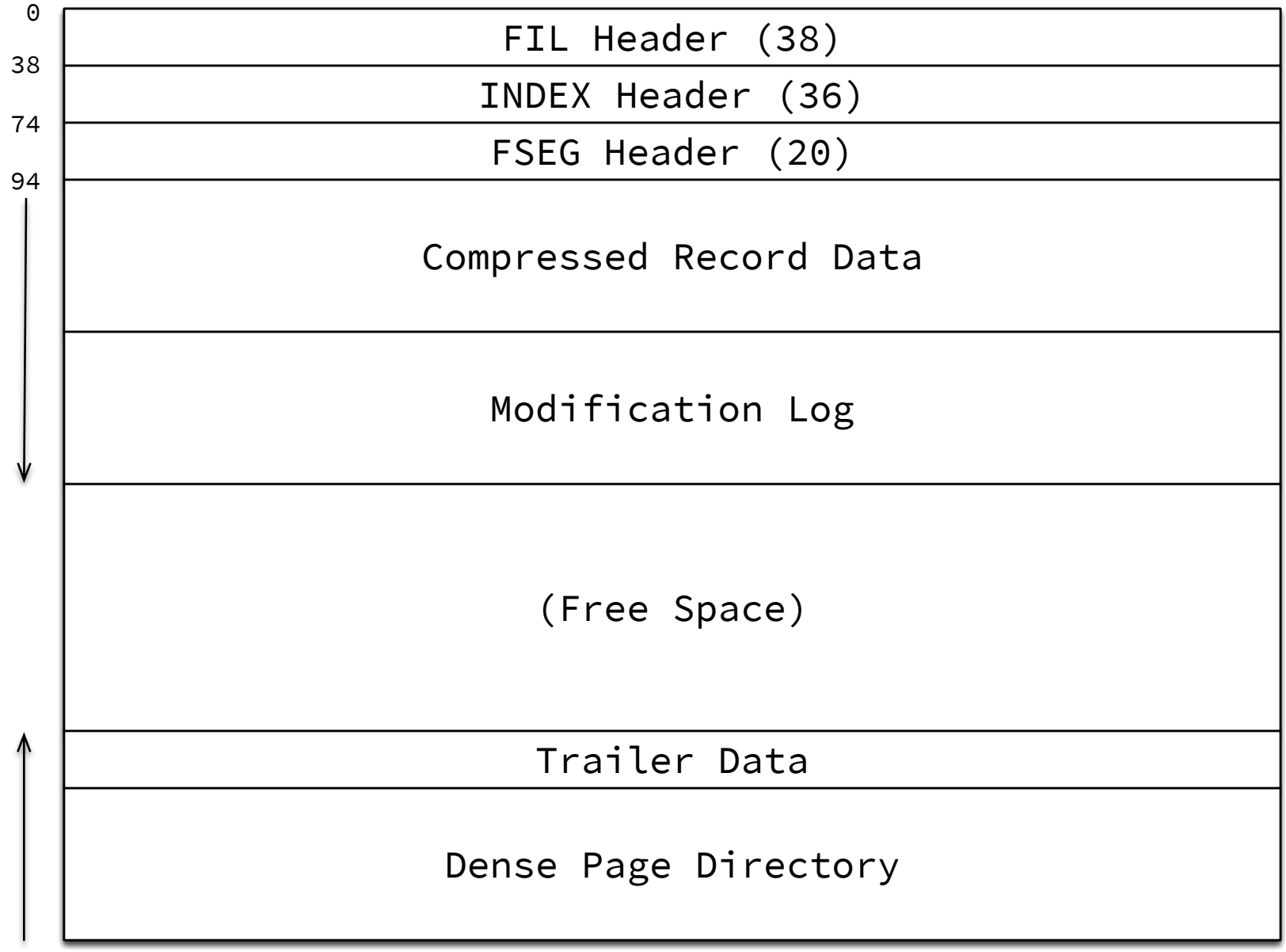
Record Format - Secondary Key - Leaf Pages



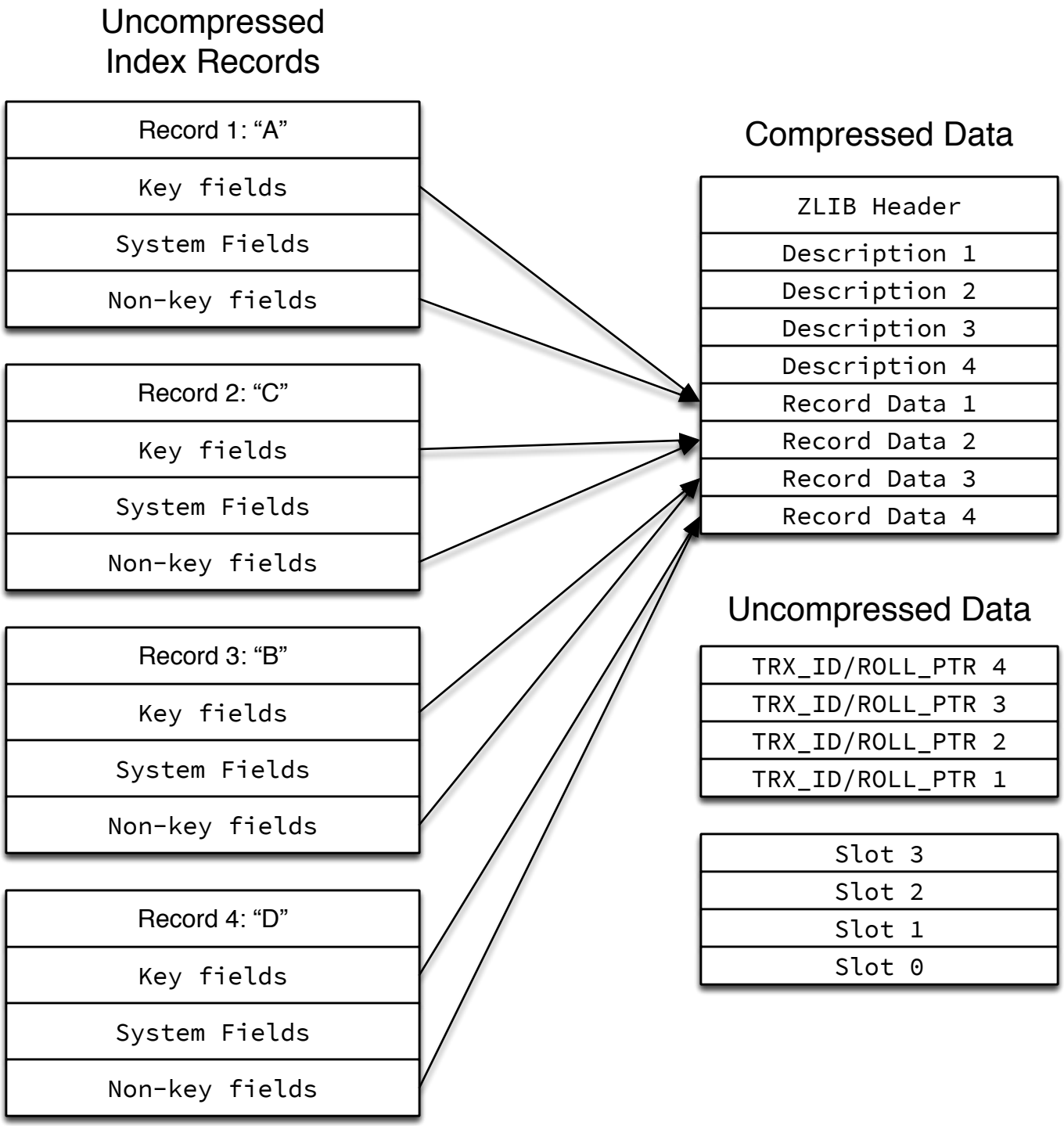
Record Format - Secondary Key - Non-Leaf Pages



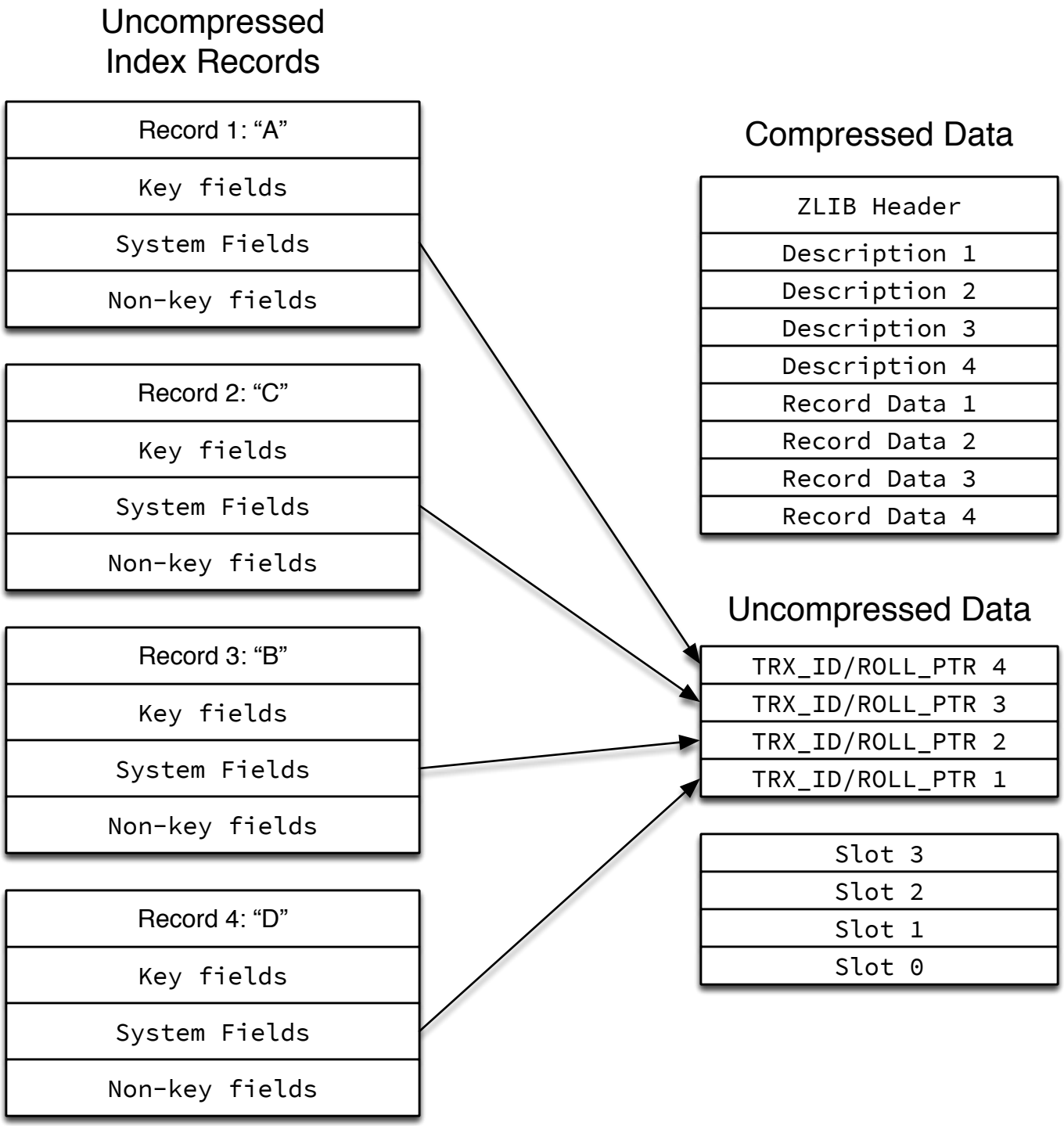
INDEX Overview



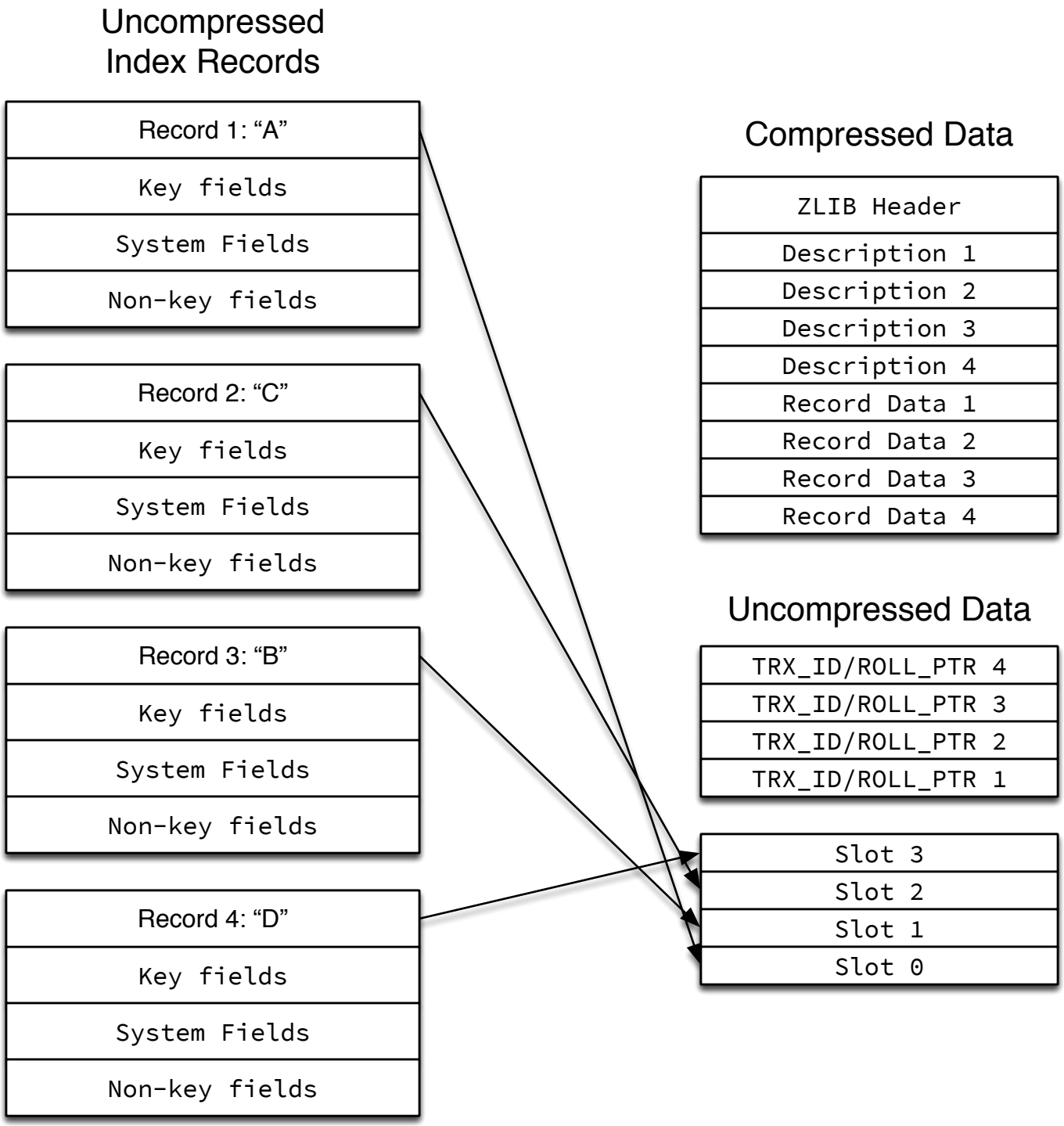
Compressed Index Data



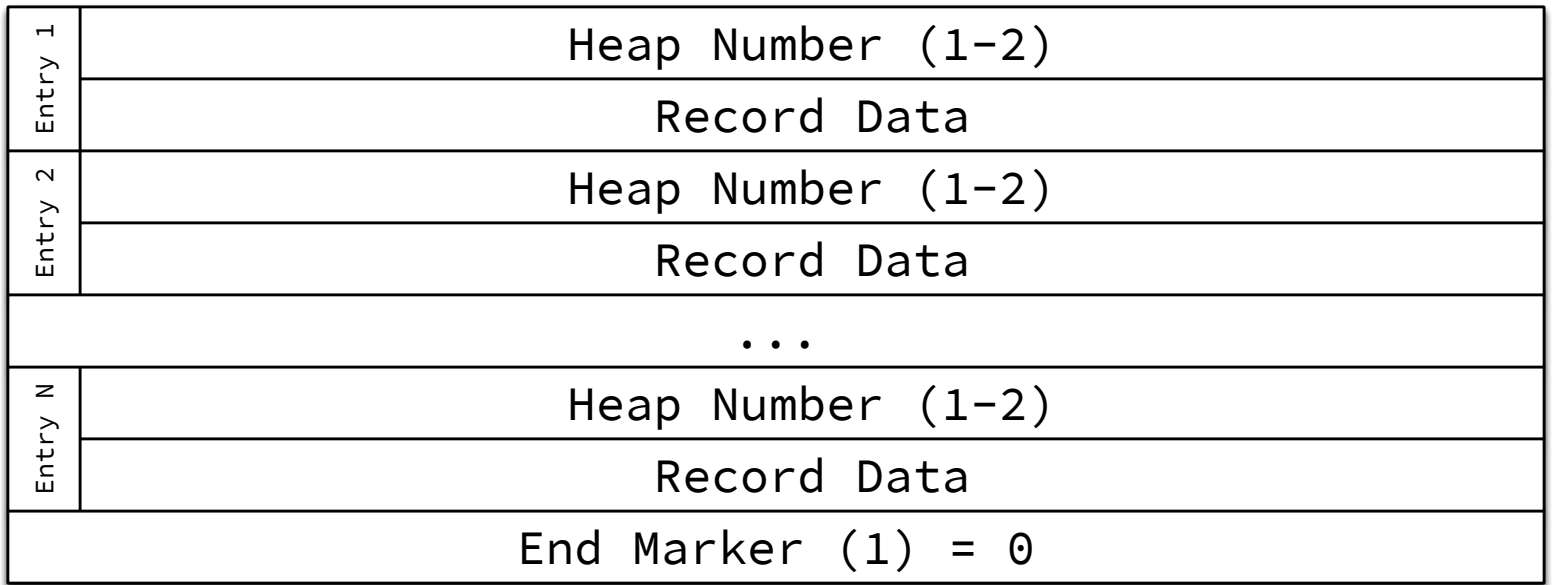
Compressed Index Data



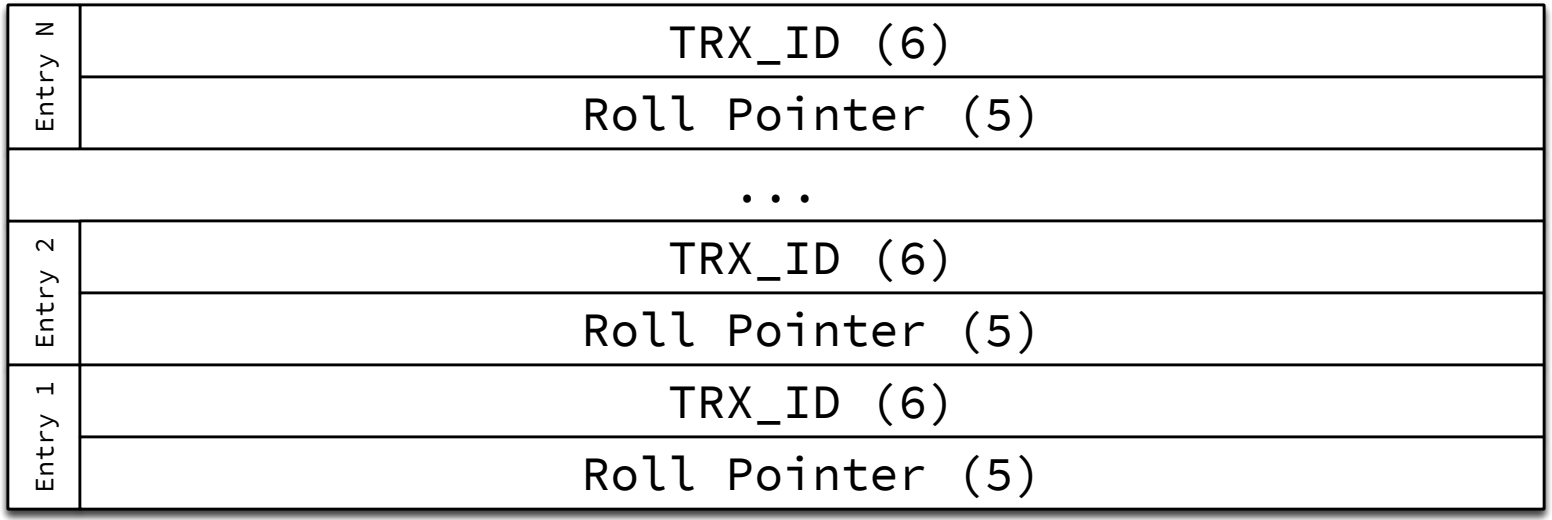
Compressed Index Data



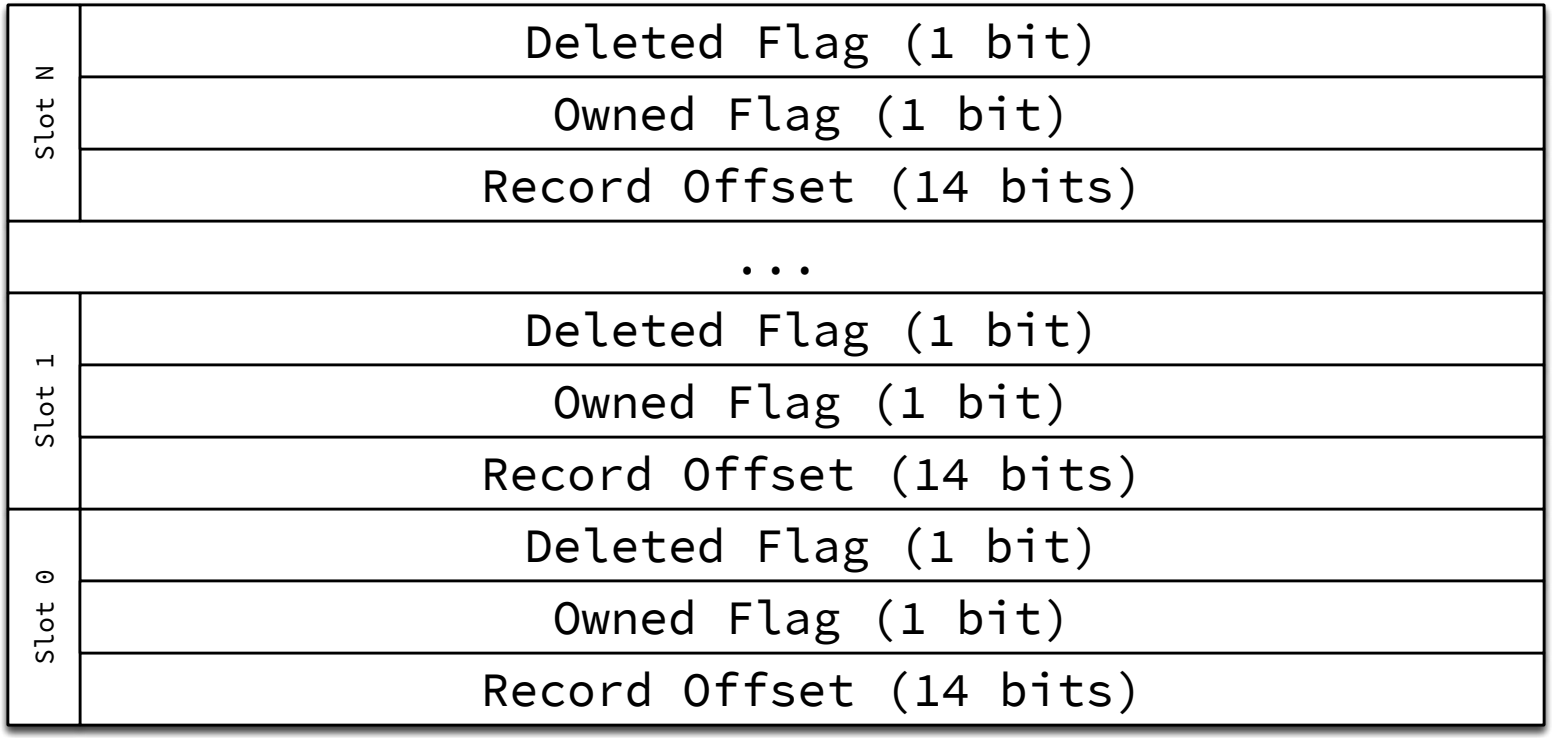
Modification Log



Trailer Data



Dense Page Directory



The dense page directory contains one entry per records, in the key's collation order.
All directory slots will own a minimum of 4 and maximum of 8 records.
The page directory grows "downwards" from the end of the page.

Record Format - Change Buffer - Leaf Pages