SQLite Index Visualization

Anton Sukhachev

Content table

- About me
- SQLite
- SQLite index structure
- SQLite index dump/tracing tools
- Experiments
- Practical Tips

About me

Backend Developer (10+ years experience)

Author of 20+ technical articles

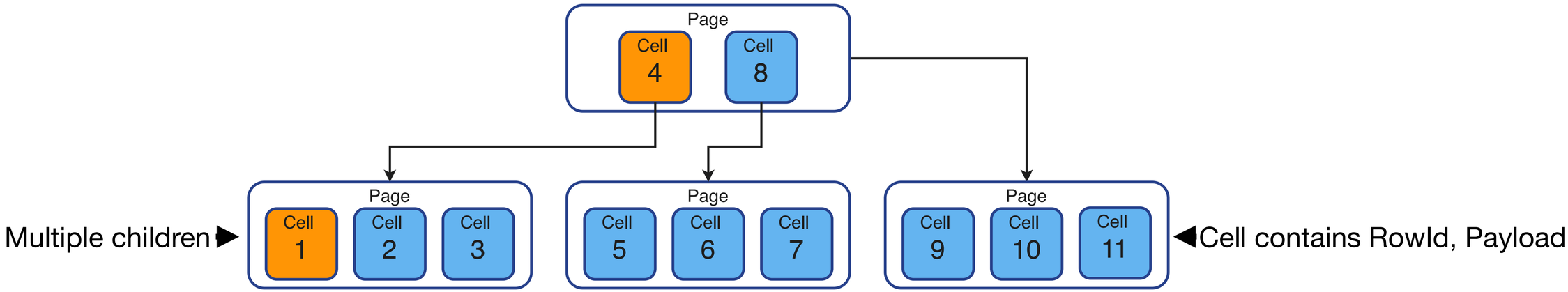
Key projects:

- PHP Generics<T> - RFC style without annotations
- PHP Bison Skeleton - parser generator (PHP Engine)
- PHP var_sizeof() - use FFI to calculate real variable size

I enjoy exploring how things work under the hood

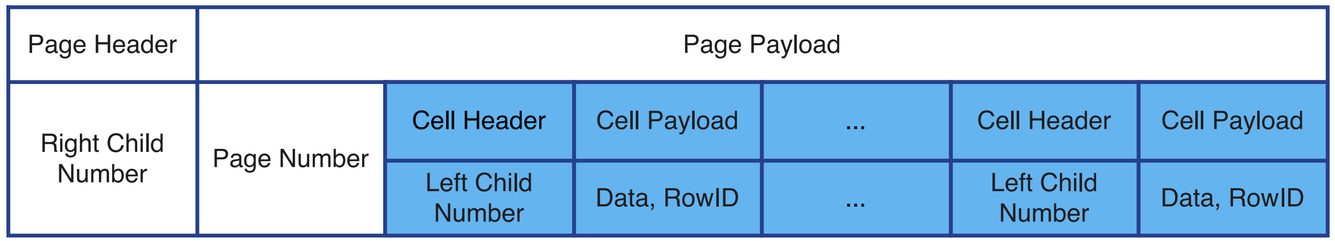
https://www.sqlite.org/famous.html

For index and data ▶ B-tree structure



Memory structure

Memory friendly ▶



Code structure

Page

sqlite/src/btreeInt.h

```
struct MemPage {
  Pgno pgno;           /* Page number for this page */
  u16 nCell;           /* Number of cells on this page, local and ovfl */
  u8 *aCellIdx;        /* The cell index area */
  u8 *aData;           /* Pointer to disk image of the page data */
  ...
};
```

Cell

sqlite/src/btreeInt.h

```
struct CellInfo {
  u8 *pPayload;  /* Pointer to the start of payload */
  ...
};
```

Index information (sqlite3 analyzer)

Meta data only▶

```
sqlite3_analyzer database.sqlite
...
Page size in bytes.................................. 4096
...
*** Index IDX of table TABLE_TEST ******************************************
Number of entries.................................. 1000
B-tree depth....................................... 2
Total pages used................................... 4
..
```

Index data dump tool

```
char *sqlite3DebugGetMemoryPayload(Mem *mem);

char **sqlite3DebugGetCellPayloadAndRowId(BtCursor *pCur, MemPage *pPage, int cellIndex);

void sqlite3DebugBtreeIndexDump(BtCursor *pCur, int pageNumber);
```

Dump current index ▶

```
SQL query -> selected index -> stdout
```

```
sqlite3BtreeIndexDump: page, number=3, rightChildPageNumber=99
sqlite3BtreeIndexDump: cell, number=0, leftChildPageNumber=7, payload=384, rowId=384
sqlite3BtreeIndexDump: cell, number=1, leftChildPageNumber=8, payload=742, rowId=742
...
```

Index data dump tool usage

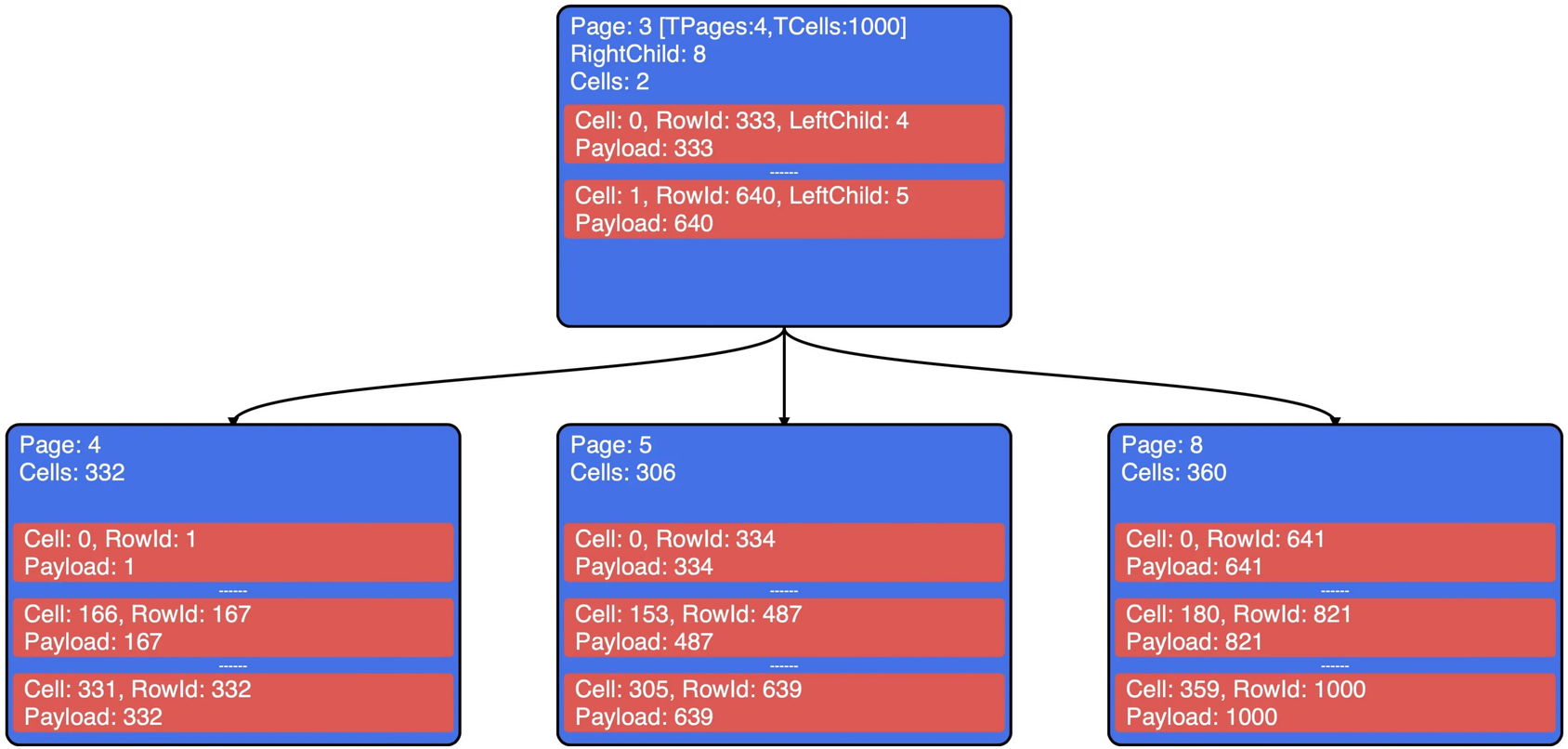
```
docker run -it --rm -v "$PWD":/app/data --platform linux/x86_64 mrsuh/sqlite-index bash
```

```
sh bin/dump-index.sh database.sqlite "SELECT * FROM table INDEXED BY index WHERE column=1" dump.txt
```

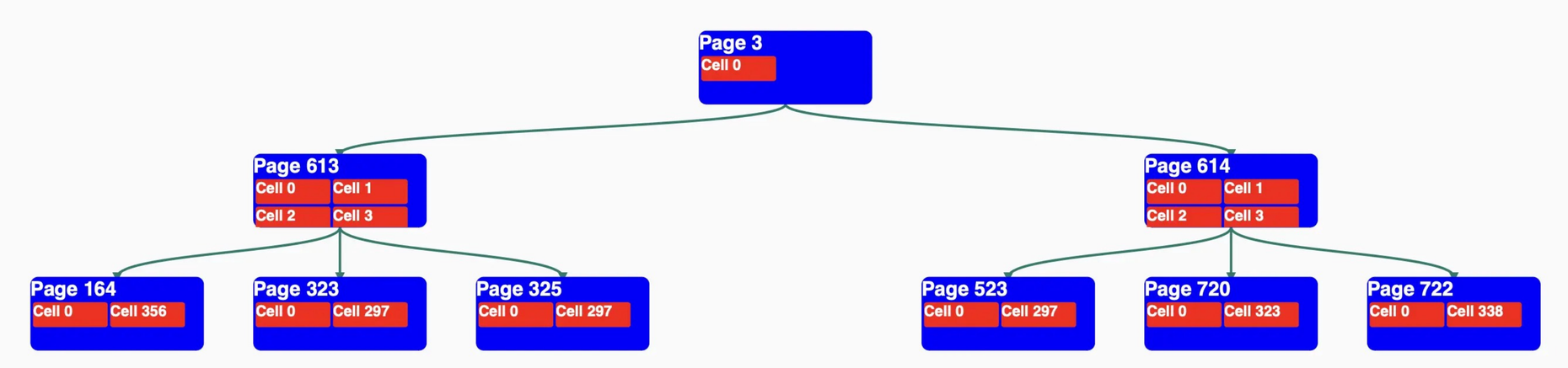
STDOUT▶

```
sqlite3BtreeIndexDump: page, number=3, rightChildPageNumber=99
sqlite3BtreeIndexDump: cell, number=0, leftChildPageNumber=7, payload=384, rowId=384
sqlite3BtreeIndexDump: cell, number=1, leftChildPageNumber=8, payload=742, rowId=742
...
sqlite3BtreeIndexDump: page, number=99, rightChildPageNumber=-1
sqlite3BtreeIndexDump: cell, number=0, leftChildPageNumber=-1, payload=9642, rowId=9642
sqlite3BtreeIndexDump: cell, number=1, leftChildPageNumber=-1, payload=9643, rowId=9643
...
sqlite3BtreeIndexDump: page, number=7, rightChildPageNumber=-1
sqlite3BtreeIndexDump: cell, number=0, leftChildPageNumber=-1, payload=1, rowId=1
sqlite3BtreeIndexDump: cell, number=1, leftChildPageNumber=-1, payload=2, rowId=2
...
```

Index visualization v1 (JS D3 library)



Index visualization v1 (JS D3 library)

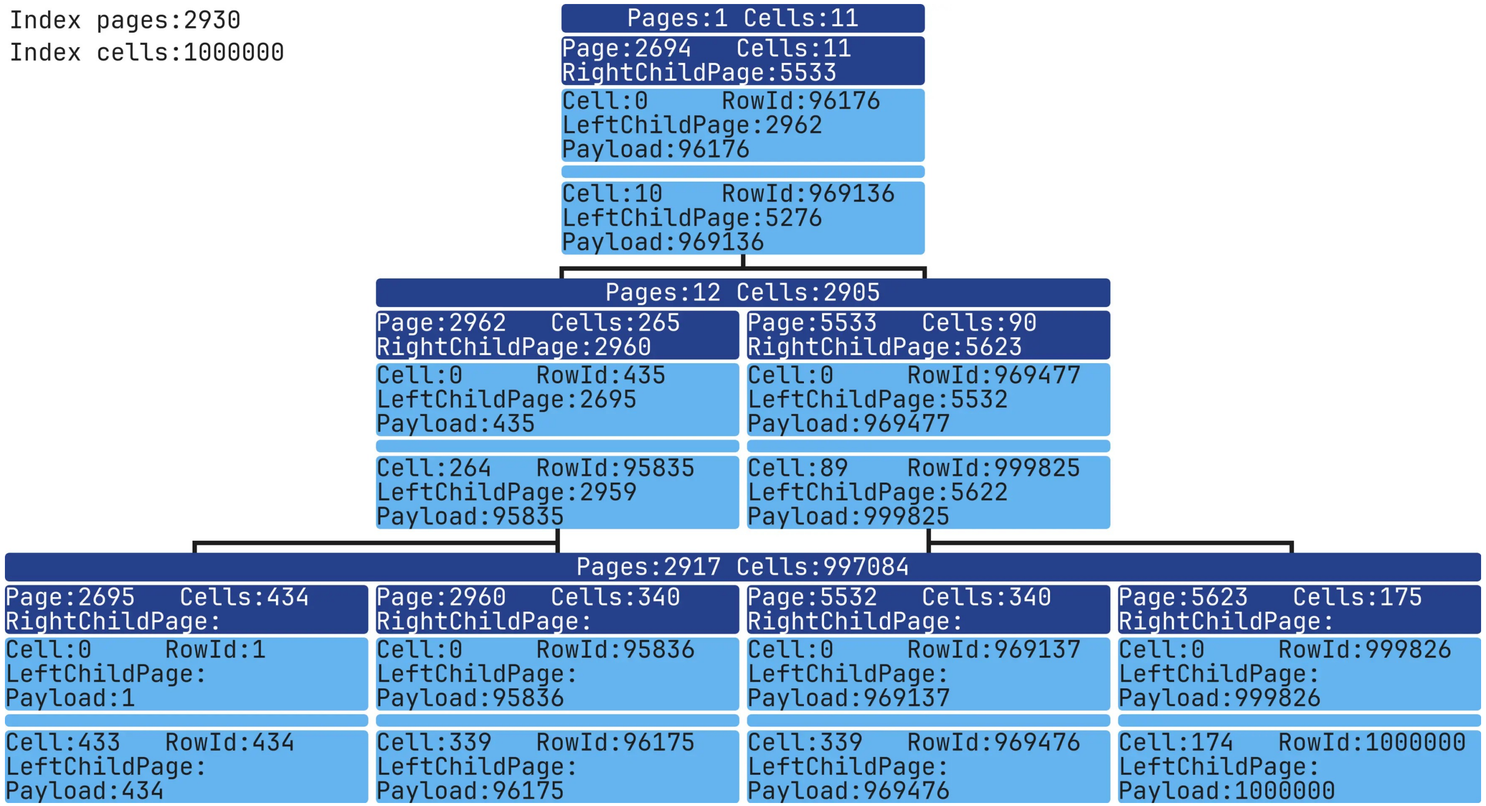


Index visualization v2 (Text)

```
-----------------------------------------------------------------------------------------------------------------------
 Total Pages: 29
 Total Cells: 1000
-----------------------------------------------------------------------------------------------------------------------
 Level: 1                          |============================================|
 Pages: 1                          |Page: 3   | RightChildPage: 53  | Cells: 27 |
 Cells: 29                         |============================================|
                                   |Cell: 0   | LeftChildPage: 47   | RowId: 1  |
                                   |Payload: 00000000000000000000000000000000     |
                                   |--------------------------------------------|
                                   |                  * * *                     |
                                   |--------------------------------------------|
                                   |Cell: 26   | LeftChildPage: 78   | RowId: 5 |
                                   |Payload: 00000000000000000000000000000000     |
                                   |============================================|
-----------------------------------------------------------------------------------------------------------------------
 Level: 2      |============================================| |============================================|
 Pages: 50     |Page: 3   | RightChildPage: 53  | Cells: 27 | |Page: 3   | RightChildPage: 53  | Cells: 27 |
 Cells: 400    |============================================| |============================================|
               |Cell: 0   | LeftChildPage: 47   | RowId: 1  | |Cell: 0   | LeftChildPage: 47   | RowId: 1  |
               |Payload: 00000000000000000000000000000000     | |Payload: 00000000000000000000000000000000     |
               |--------------------------------------------| |--------------------------------------------|
               |                  * * *                     | |                  * * *                     |
               |--------------------------------------------| |--------------------------------------------|
               |Cell: 26   | LeftChildPage: 78   | RowId: 5 | |Cell: 26   | LeftChildPage: 78   | RowId: 5 |
               |Payload: 00000000000000000000000000000000     | |Payload: 00000000000000000000000000000000     |
               |============================================| |============================================|
-----------------------------------------------------------------------------------------------------------------------
```

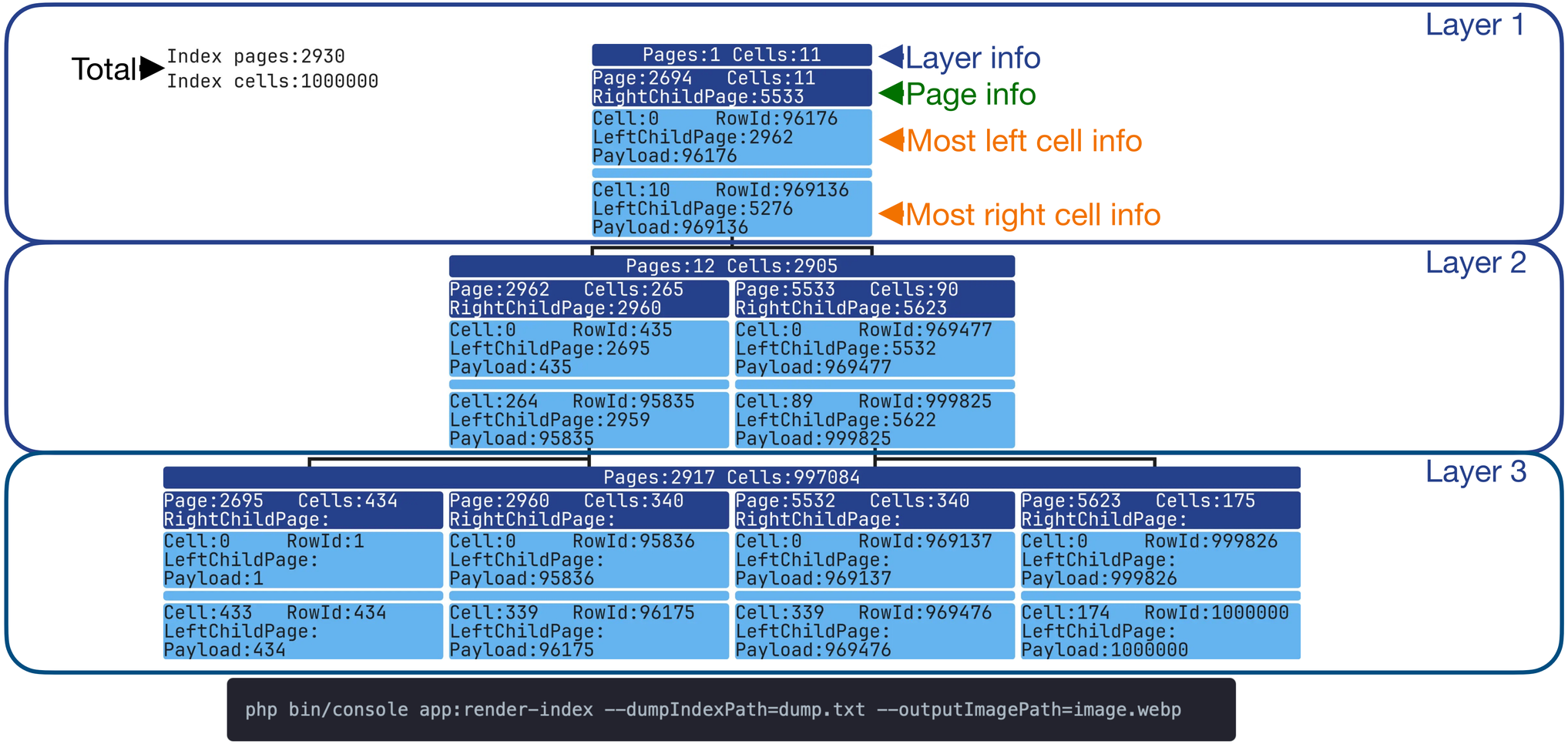
Index visualization v3 (PHP ImageMagick)

Index pages:2930
Index cells:1000000




```
php bin/console app:render-index --dumpIndexPath=dump.txt --outputImagePath=image.webp
```

Index visualization v3 (PHP ImageMagick)




```
php bin/console app:render-index --dumpIndexPath=dump.txt --outputImagePath=image.webp
```

Index search

VM OPcodes▶

```
EXPLAIN SELECT rowId, column1 FROM table_test INDEXED BY idx WHERE column1 = 1;
addr  opcode         p1    p2    p3    p4             p5  comment
----  -------------  ----  ----  ----  -------------  --  -------------
0     Init           0     11    0                    0   Start at 11
1     OpenRead       1     2694  0     k(2,,)         2   root=2694 iDb=0; idx
2     Explain        2     0     0     SEARCH table_test USING COVERING INDEX idx (column1=
3     Integer        1     1     0                    0   r[1]=1
4     SeekGE         1     10    1     1              0   key=r[1]
5       IdxGT          1     10    1     1              0   key=r[1]
6       IdxRowid       1     2     0                    0   r[2]=rowid; table_test.rowid
7       Column         1     0     3                    0   r[3]= cursor 1 column 0
8       ResultRow      2     2     0                    0   output=r[2..3]
9     Next           1     5     1                    0
10    Halt           0     0     0                    0
11    Transaction    0     0     3     0              1   usesStmtJournal=0
12    TableLock      0     2     0     table_test     0   iDb=0 root=2 write=0
13    Goto           0     1     0                    0
```

Index query tracing tool

```
if (sqlite3DebugIsBtreeIndexSeekEnabled()) {
  char **payload = sqlite3DebugGetCellPayloadAndRowId(pCur, pPage, idx);
  printf(
      "sqlite3DebugBtreeIndexMoveto: pageNumber=%d, cellNumber=%d, payload=%s, rowId=%s\n",
      pPage->pgno,
      idx,
      payload[0],
      payload[1]
  );
  sqlite3DebugFreeCellPayloadAndRowId(payload);
}
```

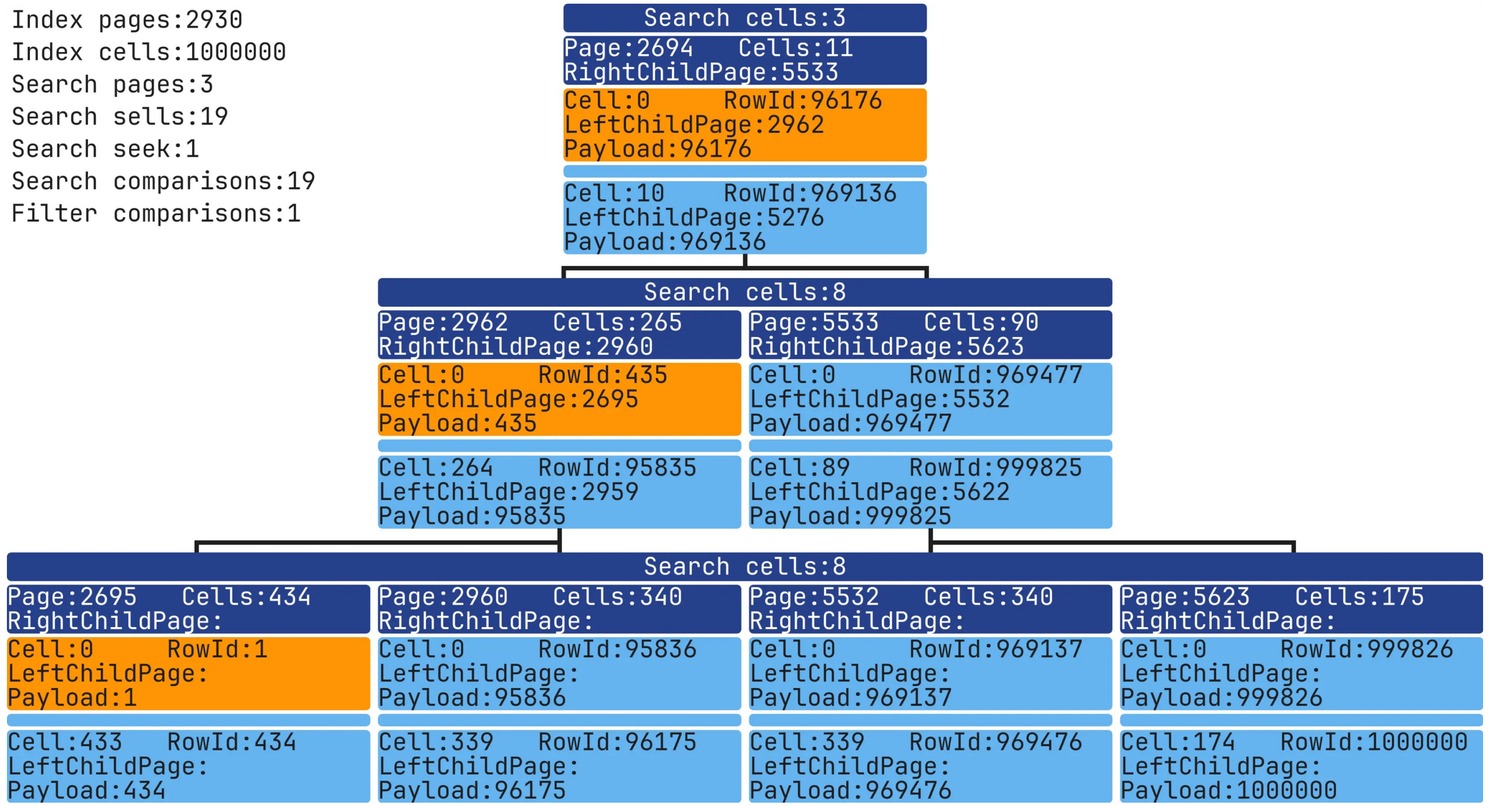
STDOUT▶

```
sqlite3DebugMoveToRoot:
sqlite3DebugBtreeIndexMoveto: pageNumber=2694, cellNumber=5, payload=532656, rowId=532656
sqlite3DebugBtreeIndexCompare: index=1, type=int, value=1
sqlite3DebugBtreeIndexMoveto: pageNumber=2694, cellNumber=2, payload=270768, rowId=270768
sqlite3DebugBtreeIndexCompare: index=1, type=int, value=1
...
sqlite3DebugBtreeIndexMoveto: pageNumber=2695, cellNumber=0, payload=1, rowId=1
sqlite3DebugBtreeIndexCompare: index=1, type=int, value=1
sqlite3DebugResultRow:
1|1
```

Index query tracing tool usage

Query info ▶

```
Index pages:2930
Index cells:1000000
Search pages:3
Search sells:19
Search seek:1
Search comparisons:19
Filter comparisons:1
```

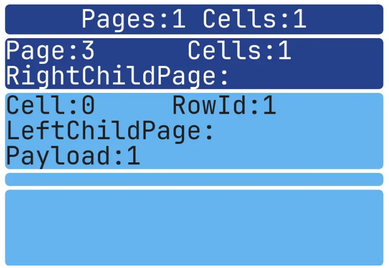


Result Cell ▶

```
docker run -it --rm -v "$PWD":/app/data --platform linux/x86_64 mrsuh/sqlite-index bash
sh bin/dump-index.sh database.sqlite "SELECT column1 FROM table_test INDEXED BY idx WHERE column1 = 1;" dump-index.txt
sh bin/dump-search.sh database.sqlite "SELECT column1 FROM table_test INDEXED BY idx WHERE column1 = 1;" dump-search.txt
php bin/console app:render-search --dumpIndexPath=dump-index.txt \
  --dumpSearchPath=dump-search.txt --outputImagePath=image.webp
```

Index with 1 record

```
Index pages:1
Index cells:1
```

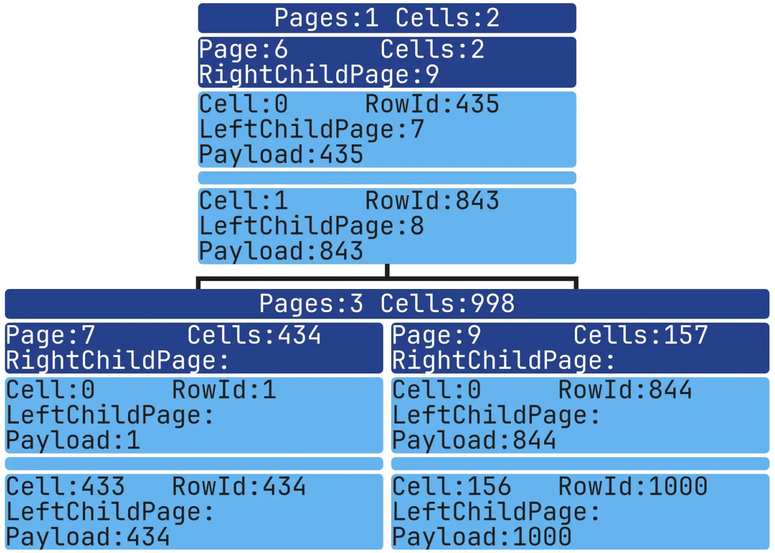


◀ First 2 pages with data

```sql
CREATE TABLE table_test (column1 INT NOT NULL);
INSERT INTO table_test (column1) VALUES (1);
CREATE INDEX idx ON table_test (column1 ASC);
```

Index with 1.000 records

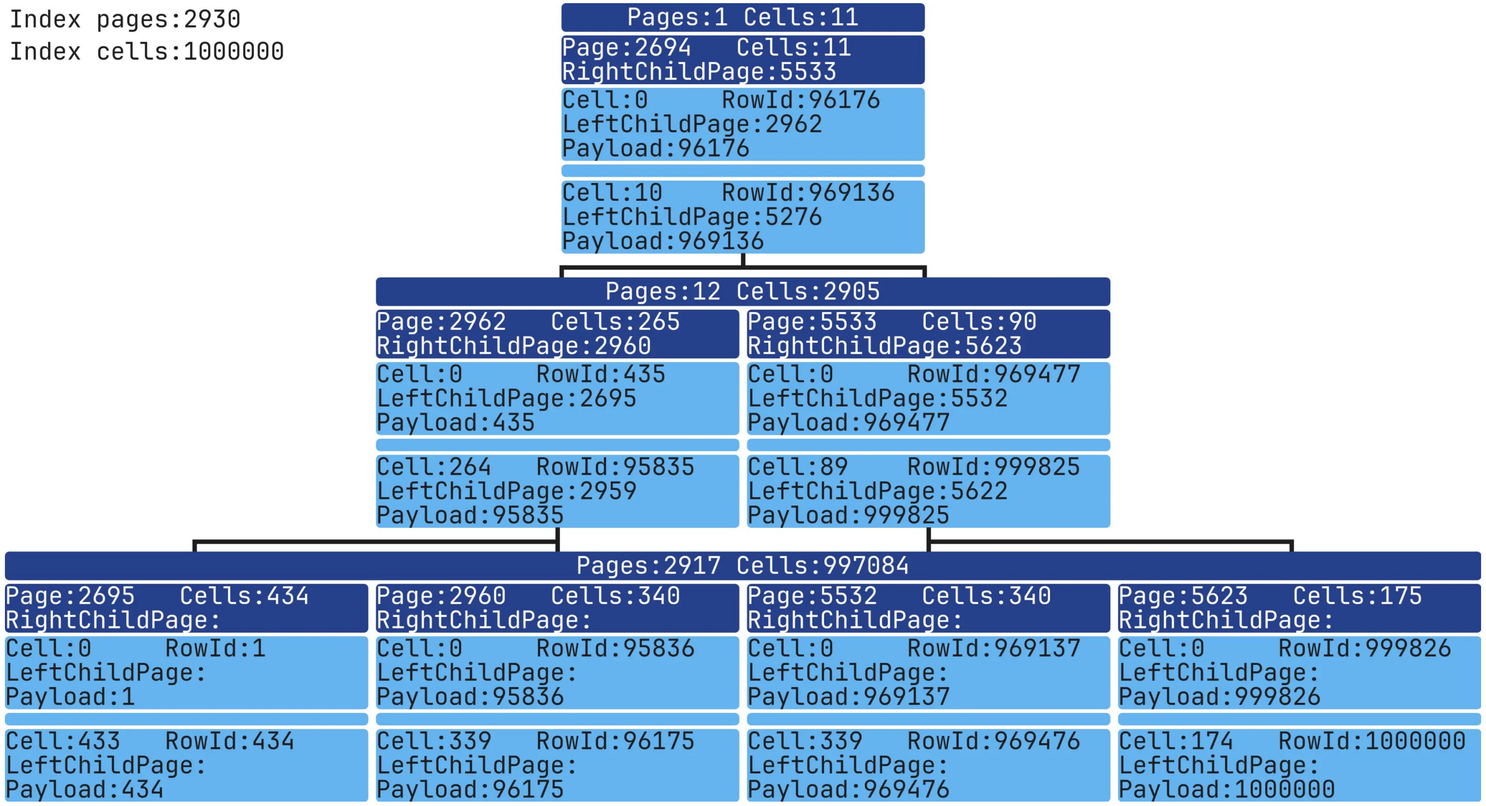
Index pages:4
Index cells:1000




```
CREATE TABLE table_test (column1 INT NOT NULL);
INSERT INTO table_test (column1) VALUES (1),(2),(3),...,(998),(999),(1000);
CREATE INDEX idx ON table_test (column1 ASC);
```

Index with 1.000.000 records

Index pages:2930
Index cells:1000000

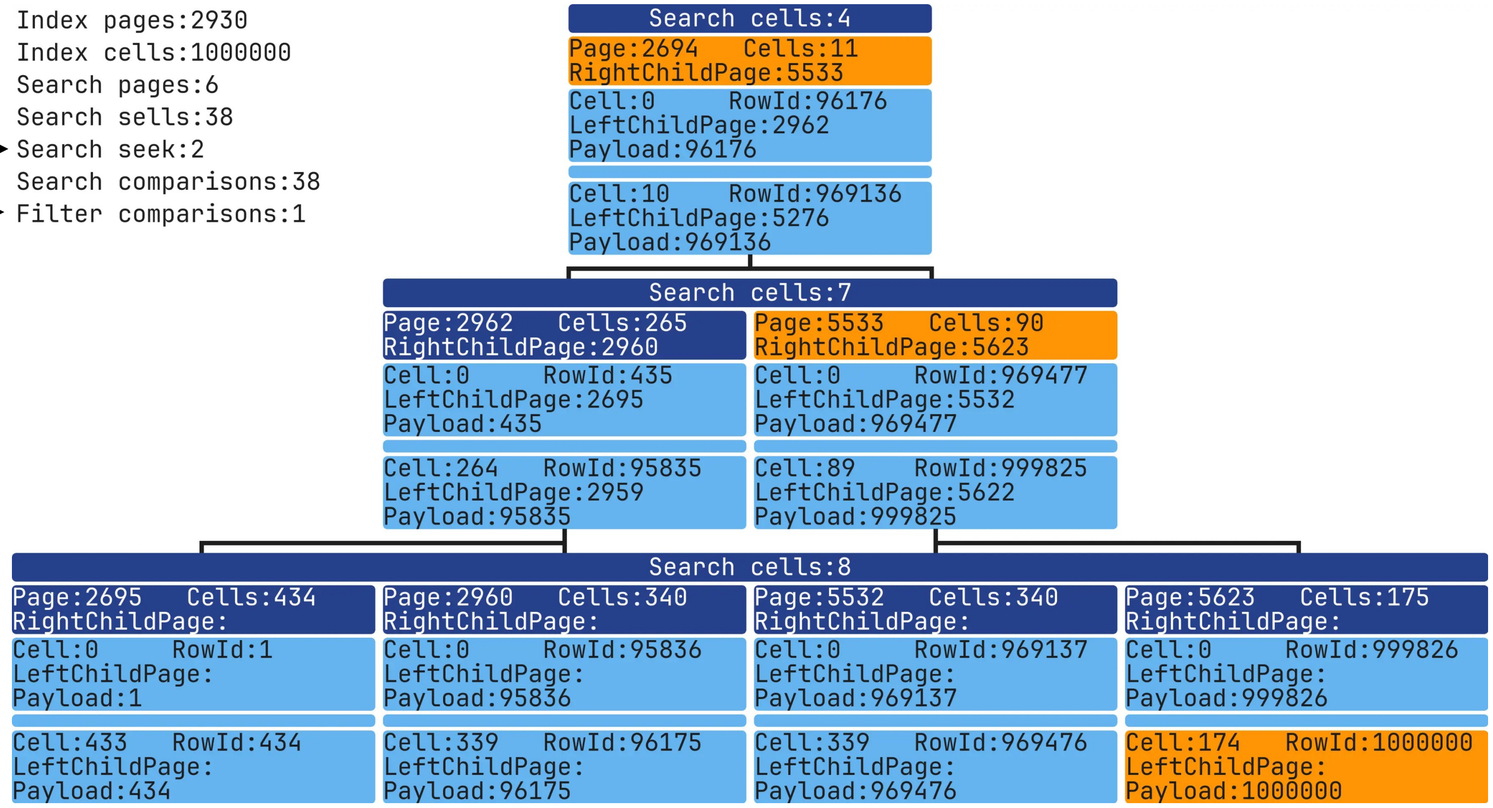



```
CREATE TABLE table_test (column1 INT NOT NULL);
INSERT INTO table_test (column1) VALUES (1),(2),(3),...,(999998),(999999),(1000000);
CREATE INDEX idx ON table_test (column1 ASC);
```

Query with multiple values in IN()

Multiple seeks ▶
No LIMIT ▶

```
Index pages:2930
Index cells:1000000
Search pages:6
Search sells:38
Search seek:2
Search comparisons:38
Filter comparisons:1
```

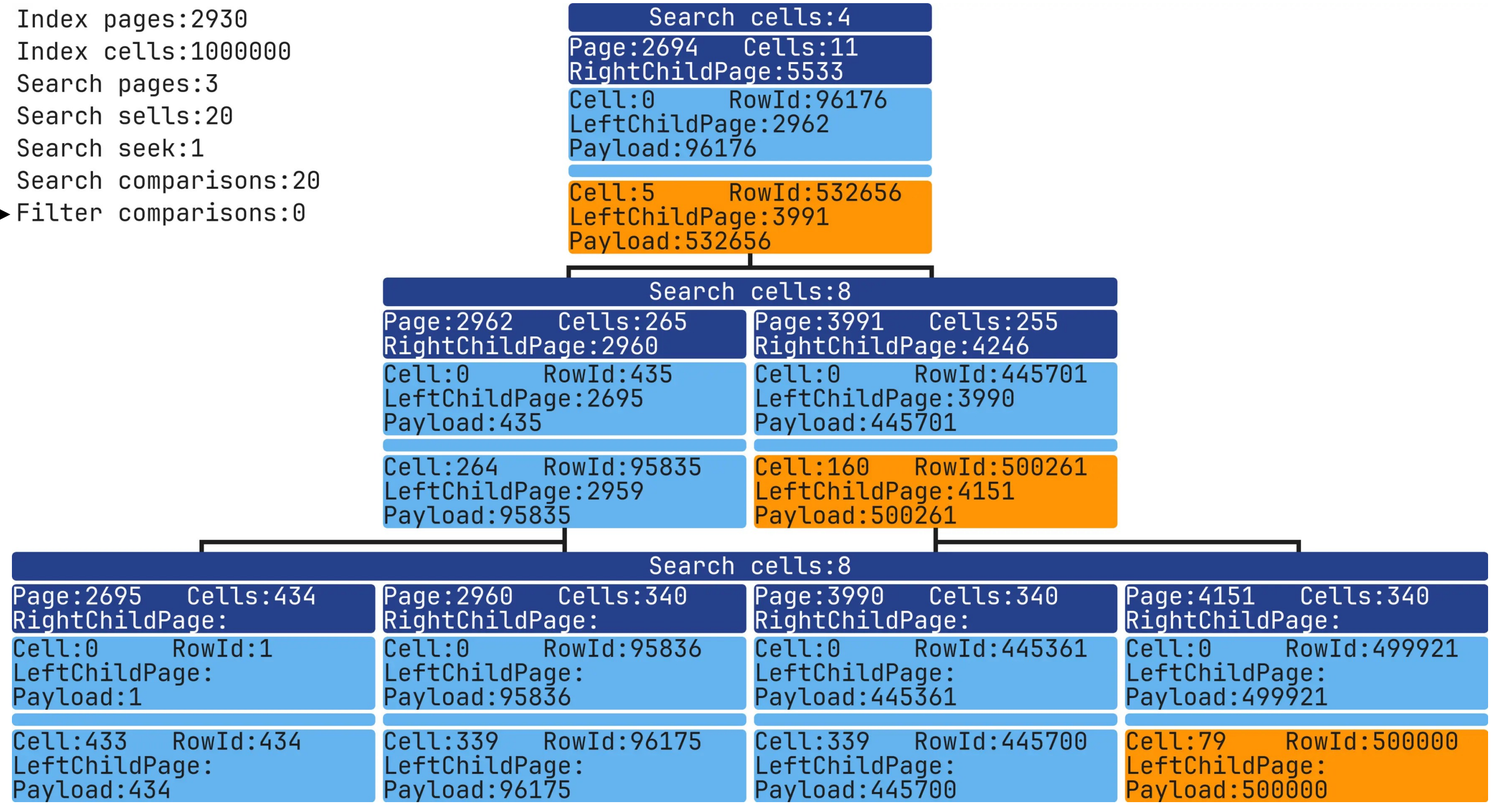



```sql
CREATE TABLE table_test (column1 INT NOT NULL);
INSERT INTO table_test (column1) VALUES (1),(2),(3),...,(999998),(999999),(1000000);
CREATE INDEX idx ON table_test (column1 ASC);
SELECT rowId, column1 FROM table_test INDEXED BY idx WHERE column1 IN (1,1000000);
```

Range searches

```
Index pages:2930
Index cells:1000000
Search pages:3
Search sells:20
Search seek:1
Search comparisons:20
Filter comparisons:0
```

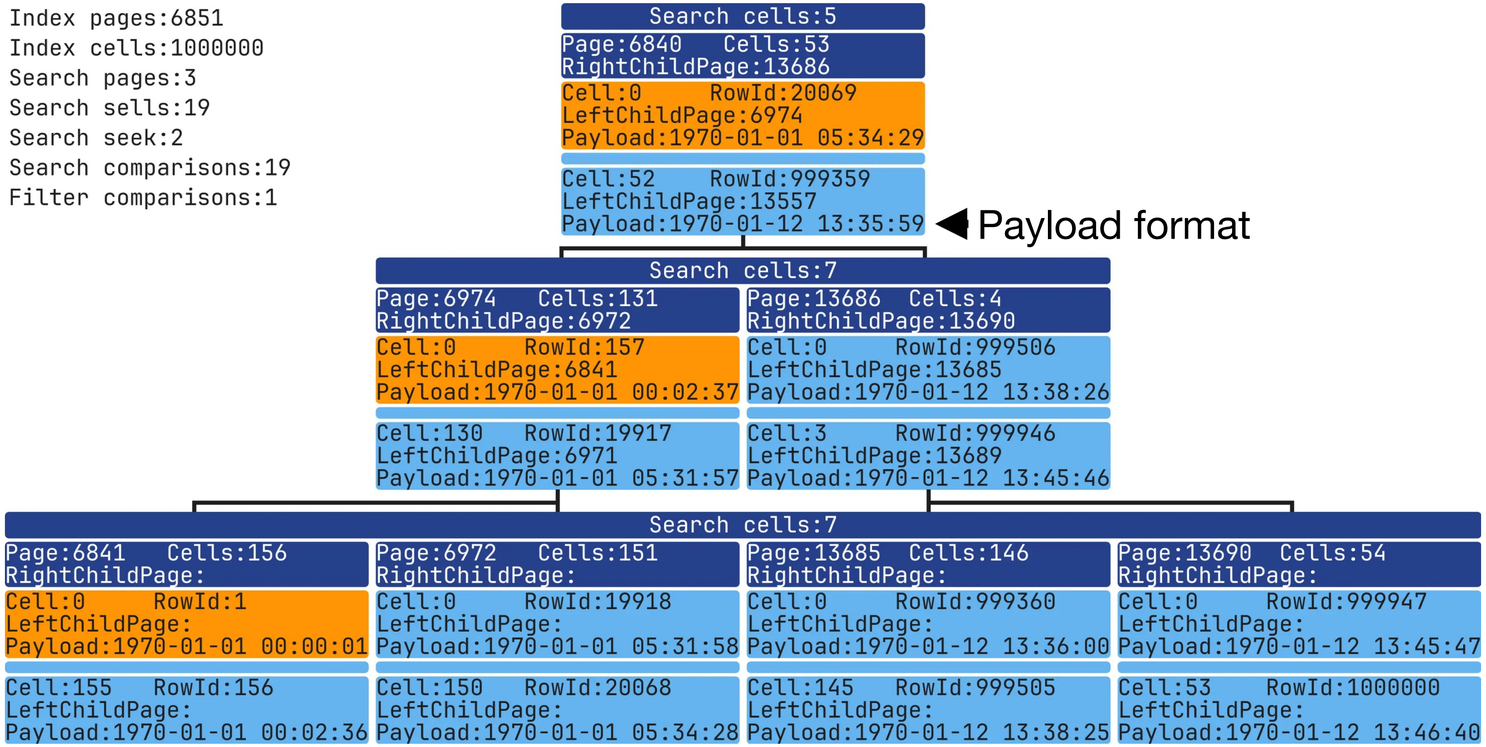
No filtering ▶




```sql
CREATE TABLE table_test (column1 INT NOT NULL);
INSERT INTO table_test (column1) VALUES (1),(2),(3),...,(999998),(999999),(1000000);
CREATE INDEX idx ON table_test (column1 ASC);
SELECT rowId, column1 FROM table_test INDEXED BY idx WHERE column1 >= 500000 LIMIT 5;
```

Expression-based searches

```
Index pages:6851
Index cells:1000000
Search pages:3
Search sells:19
Search seek:2
Search comparisons:19
Filter comparisons:1
```




```sql
CREATE TABLE table_test (column1 TEXT NOT NULL);
INSERT INTO table_test (column1) VALUES ('{"timestamp":1}'),...,('{"timestamp":1000000}');
CREATE INDEX idx ON table_test (strftime('%Y-%m-%d %H:%M:%S', json_extract(column1, '$.timestamp'), 'unixepoch') ASC);
SELECT rowId FROM table_test INDEXED BY idx
WHERE strftime('%Y-%m-%d %H:%M:%S',json_extract(column1, '$.timestamp'), 'unixepoch') = '1970-01-01 00:00:01';
```

Expression-based searches

For example, if we have an Index like this:

```
CREATE INDEX idx ON table_test (column1 + column2);
```

SQLite will not use the Index for this query:

Logically same ▶

```
SELECT * FROM table_test WHERE (column2 + column1) = 1
```

The expressions are the same in meaning but different in how they are written. You must use the exact expression as it is in the Index:

Syntactically seeks ▶

```
SELECT * FROM table_test WHERE (column1 + column2) = 1
```

Searching within a unique Index where nearly all values are filled with NULL

NULLable cells ▶

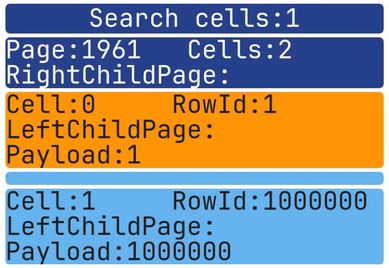
```
Index pages:2198
Index cells:1000000
Search pages:3
Search sells:20
Search seek:1
Search comparisons:20
Filter comparisons:0
```

```sql
CREATE TABLE table_test (column1 INT)
INSERT INTO table_test (column1) VALUES (1),(NULL),(NULL),...,(NULL),(NULL),(1000000);
CREATE UNIQUE INDEX idx ON table_test (column1 ASC);
SELECT rowId, column1 FROM table_test INDEXED BY idx WHERE column1 = 1;
```

Searching in an Index without NULL values

Valuable cells ▶

```
Index pages:1
Index cells:2
Search pages:1
Search sells:1
Search seek:1
Search comparisons:1
Filter comparisons:1
```

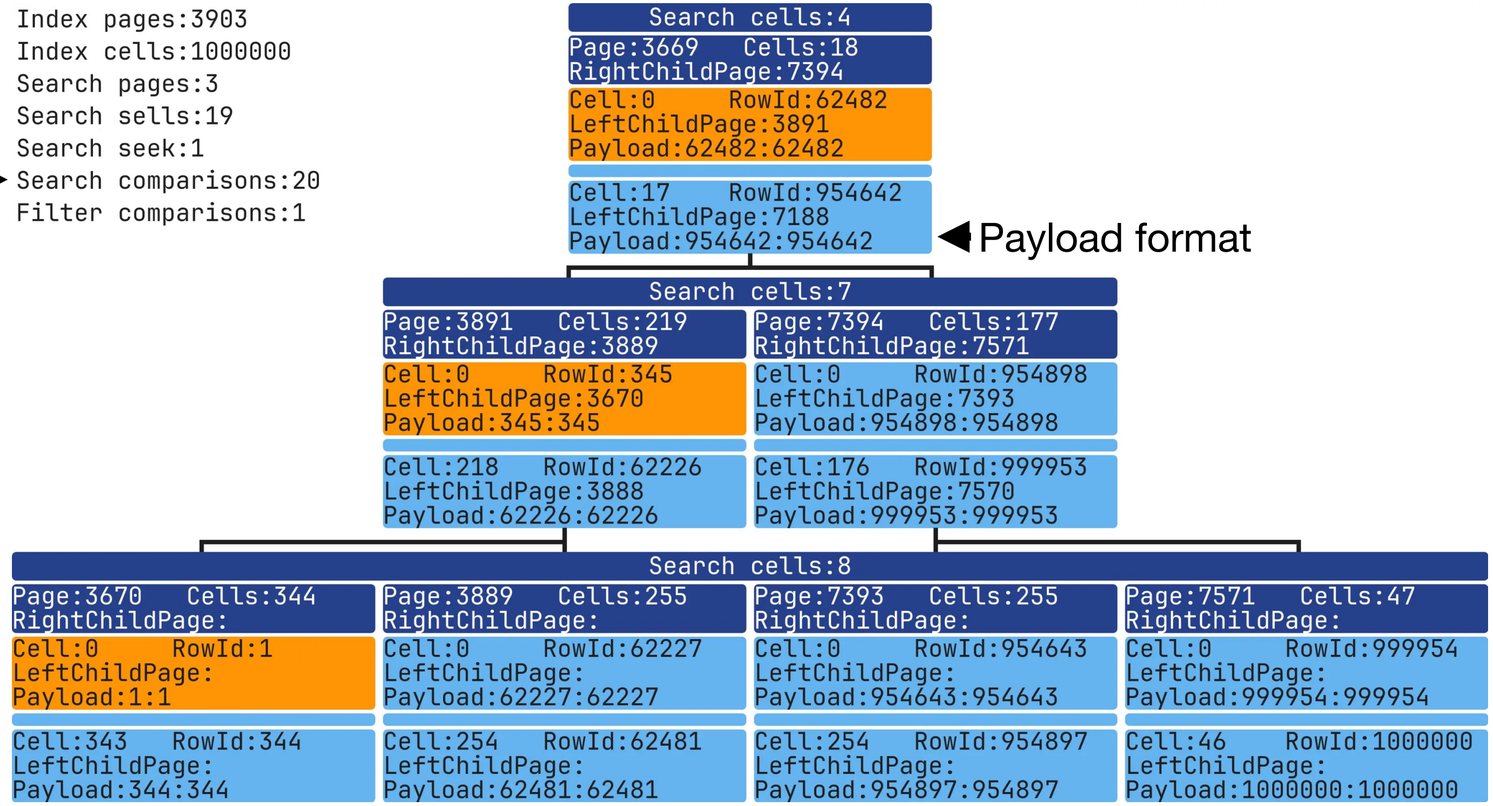



```sql
CREATE TABLE table_test (column1 INT)
INSERT INTO table_test (column1) VALUES (1),(NULL),(NULL),...,(NULL),(NULL),(1000000);
CREATE INDEX idx ON table_test (column1 ASC) WHERE column1 IS NOT NULL;
SELECT rowId, column1 FROM table_test INDEXED BY idx WHERE column1 = 1;
```

Searching in a two-column Index

```
Index pages:3903
Index cells:1000000
Search pages:3
Search sells:19
Search seek:1
Search comparisons:20
Filter comparisons:1
```

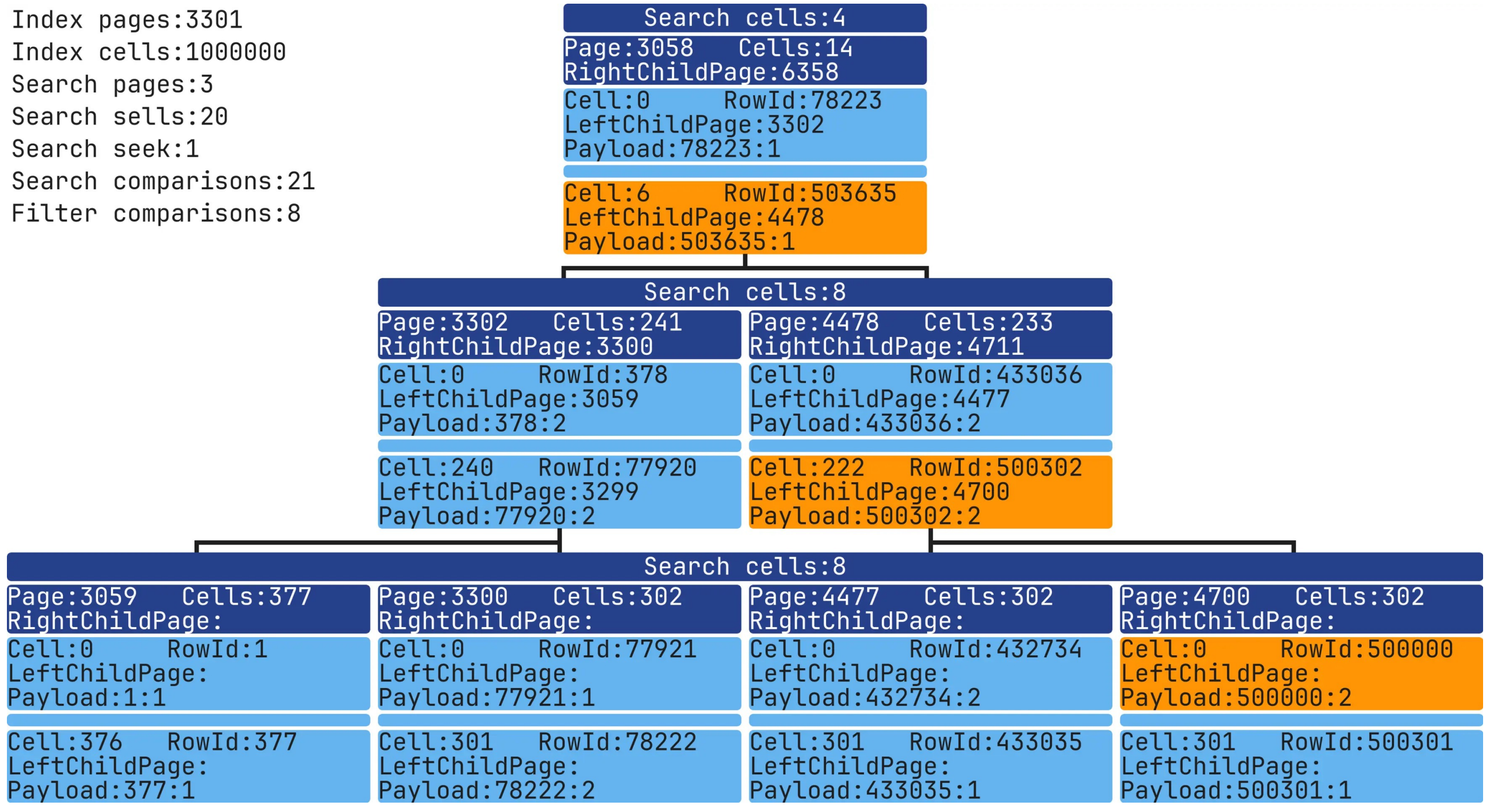
19 + 1 comparisons




```sql
CREATE TABLE table_test (column1 INT NOT NULL, column2 INT NOT NULL);
INSERT INTO table_test (column1, column2) VALUES (1,1),...,(1000000,1000000);
CREATE INDEX idx ON table_test (column1 ASC, column2 ASC);
SELECT rowId FROM table_test INDEXED BY idx WHERE column1 = 1 AND column2 = 1;
```

Searching in a two-column Index with different cardinality (idx1)

```
Index pages:3301
Index cells:1000000
Search pages:3
Search sells:20
Search seek:1
Search comparisons:21
Filter comparisons:8
```



Different indexes ▶

Use idx1 ▶

```sql
CREATE TABLE table_test (column1 INT NOT NULL, column2 INT NOT NULL);
INSERT INTO table_test (column1, column2) VALUES (1,1),...,(1000000,2);
CREATE INDEX idx_column1_column2 ON table_test (column1 ASC, column2 ASC);
CREATE INDEX idx_column2_column1 ON table_test (column2 ASC, column1 ASC);
SELECT rowId FROM table_test INDEXED BY idx_column1_column2
WHERE column1 >= 500000 AND column2 = 2 LIMIT 5;
```

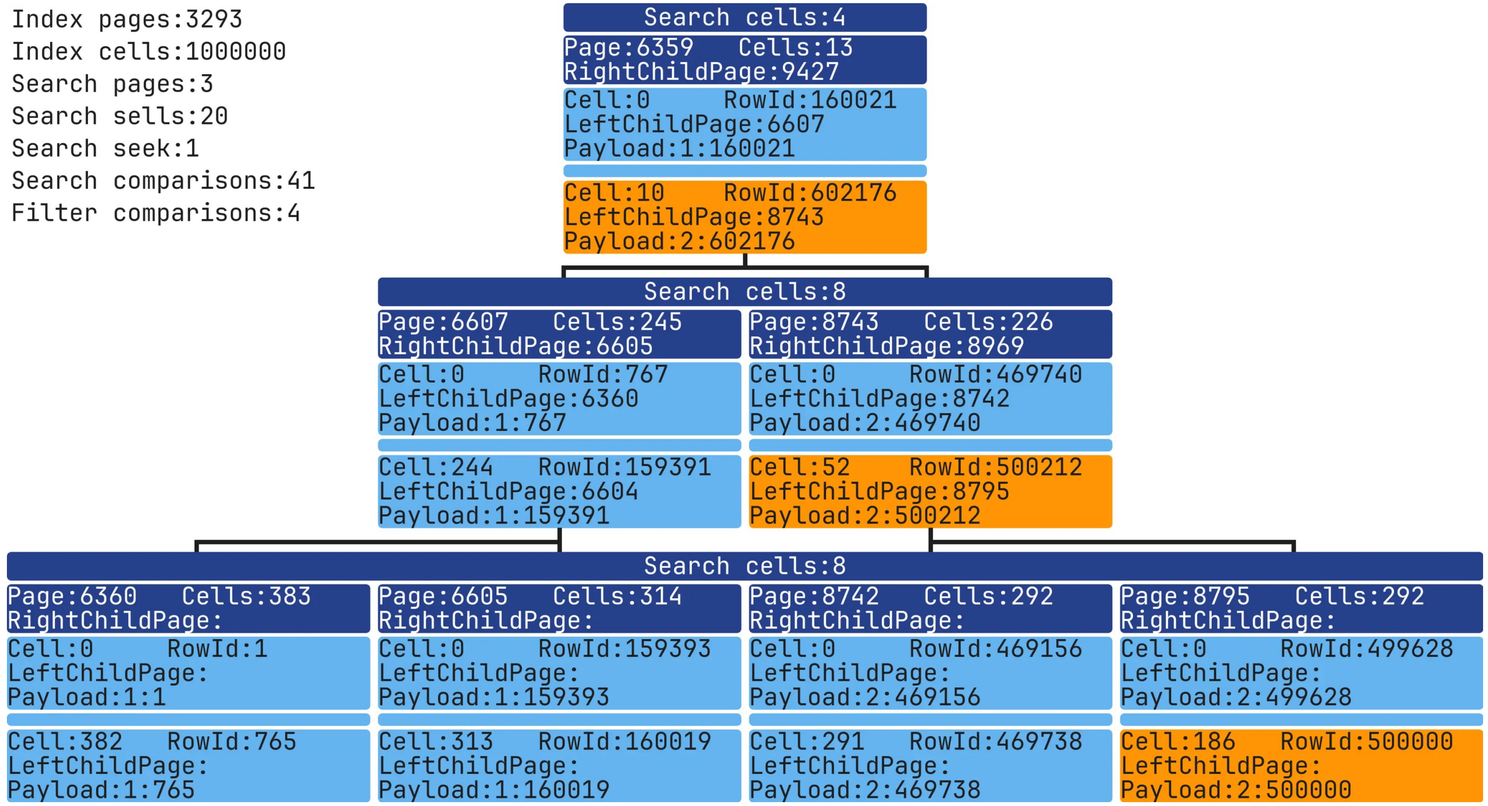
Searching in a two-column Index with different cardinality (idx1)

Unnecessary filtering ▶

column1 (>= 5000)	column2 (=2)	comments
...	...	search: 19 comparisons on column1
500000	2	search: 2 comparisons on column1 and column2
500001	1	filter: 1 comparison on column2
500002	2	filter: 1 comparison on column2
500003	1	filter: 1 comparison on column2
500004	2	filter: 1 comparison on column2
500005	1	filter: 1 comparison on column2
500006	2	filter: 1 comparison on column2
500007	1	filter: 1 comparison on column2
500008	2	filter: 1 comparison on column2
...	...	

Searching in a two-column Index with different cardinality (idx2)

```
Index pages:3293
Index cells:1000000
Search pages:3
Search sells:20
Search seek:1
Search comparisons:41
Filter comparisons:4
```



Use idx2 ▶

```sql
CREATE TABLE table_test (column1 INT NOT NULL, column2 INT NOT NULL);
INSERT INTO table_test (column1, column2) VALUES (1,1),...,(1000000,2);
CREATE INDEX idx_column1_column2 ON table_test (column1 ASC, column2 ASC);
CREATE INDEX idx_column2_column1 ON table_test (column2 ASC, column1 ASC);
SELECT rowId FROM table_test INDEXED BY idx_column2_column1
WHERE column1 >= 500000 AND column2 = 2 LIMIT 5;
```

Searching in a two-column Index with different cardinality (idx2)

Valuable data only ▶

column2 (=2)	column1 (>= 5000)	comments
...	...	39 comparisons on column1 and column2
2	500000	2 comparisons on column1 and column2
2	500002	filter: 1 comparison on column1
2	500004	filter: 1 comparison on column1
2	500006	filter: 1 comparison on column1
2	500008	filter: 1 comparison on column1
...	...	

Practical Tips

- Query with multiple values in IN() - multiple seeks
- Range searches => - without filtering
- Expression-based searches - same format, same query
- Searching in an NULL Index - valuable data only
- Complex indexes - depends on data

Anton Sukhachev
mrsuh6@gmail.com
mrsuh.com