SQLite Record Recovery

zurum

heros86@korea.ac.kr

DFRC@CIST@KU

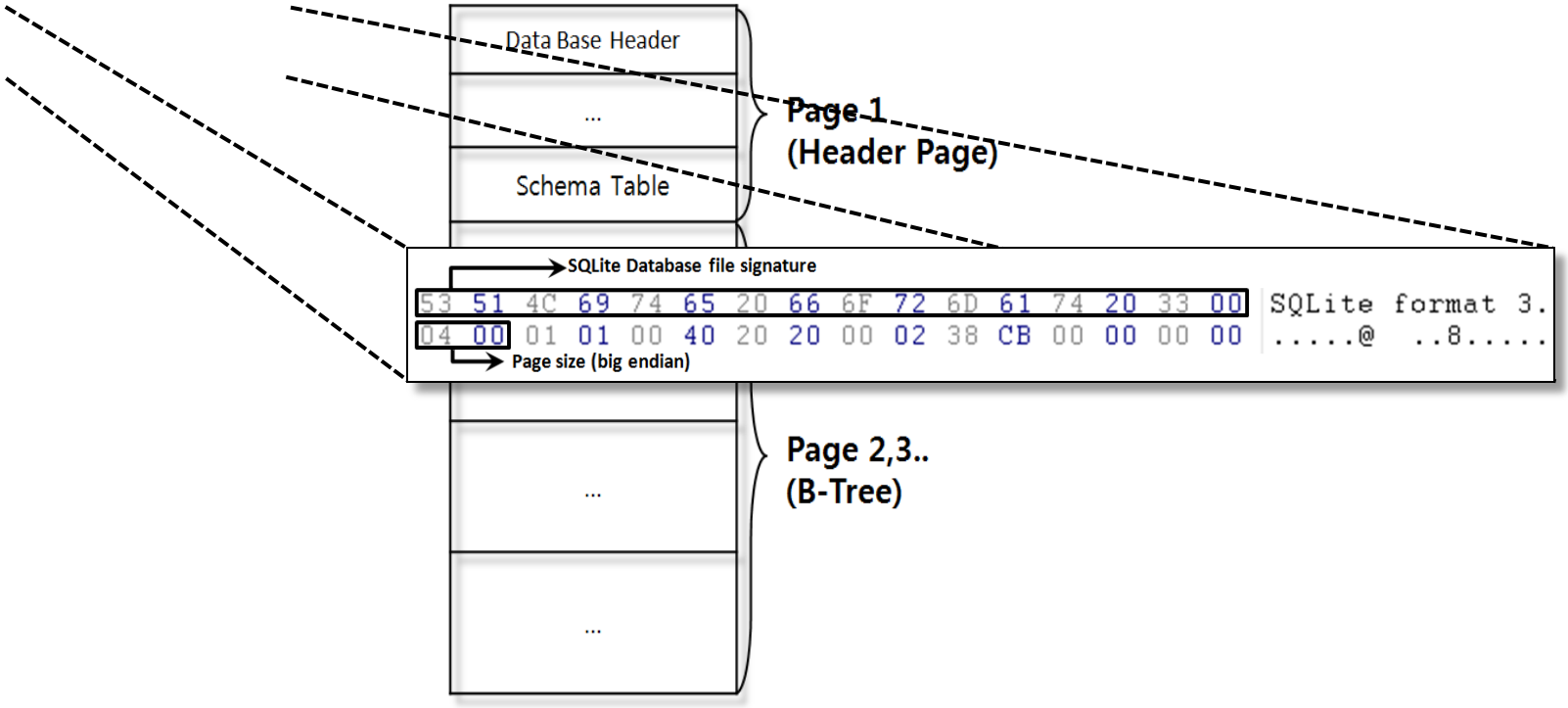
Contents

Structure of SQLite DataBase

- File & Page Structure

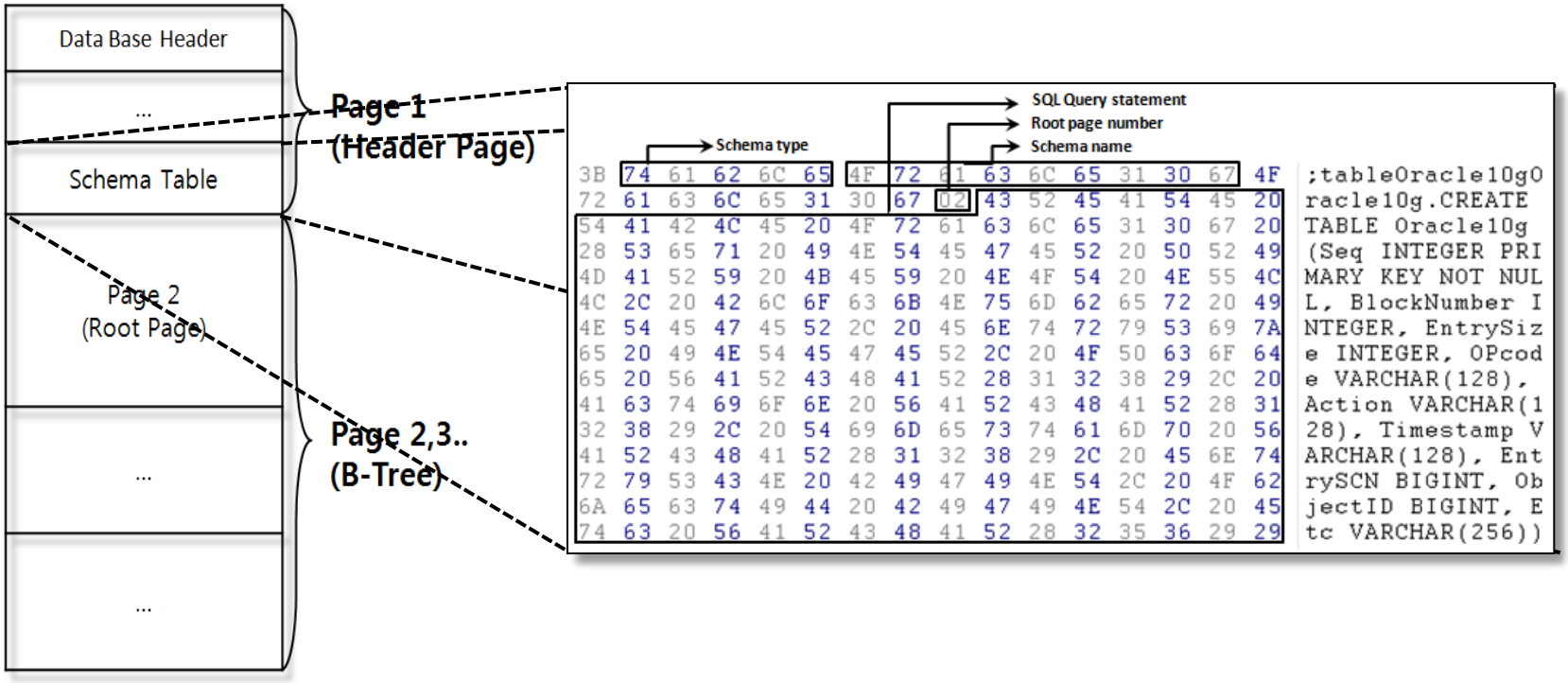
File Structure

- 전체 구조 및 파일 헤더



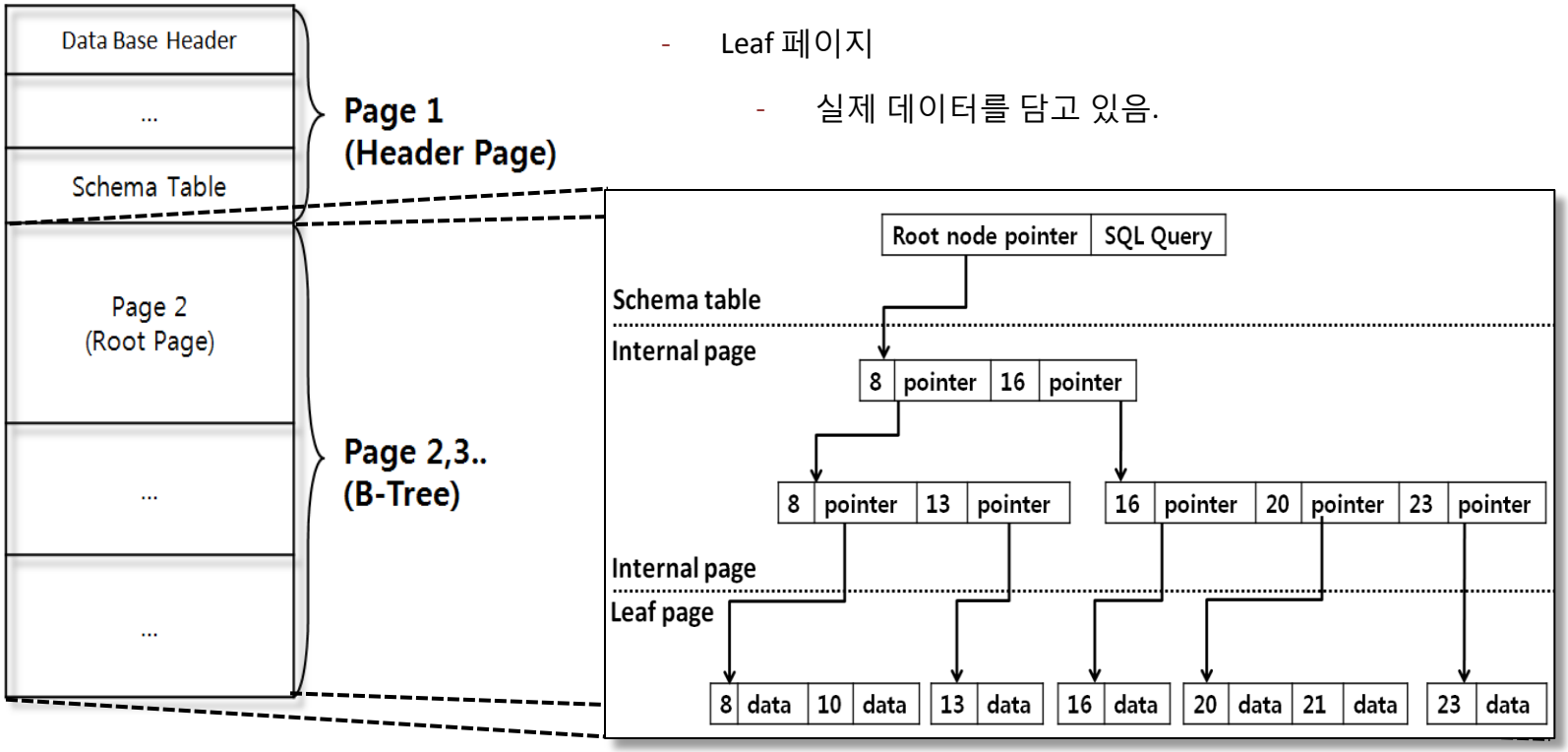
File Structure

- 스키마 테이블



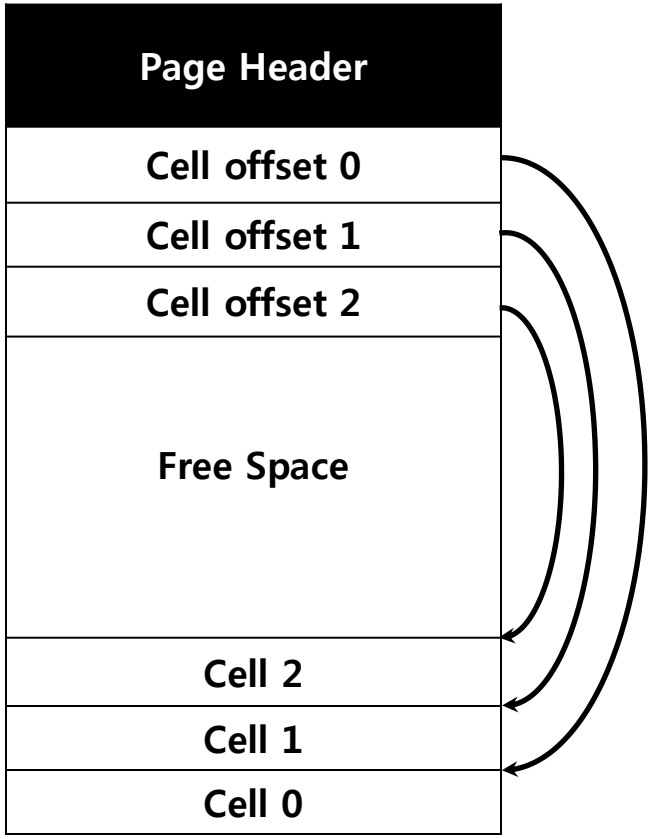
File Structure

- **Table B-tree**



- Internal 페이지
  - 하위 페이지의 포인터를 가짐.
- Leaf 페이지
  - 실제 데이터를 담고 있음.

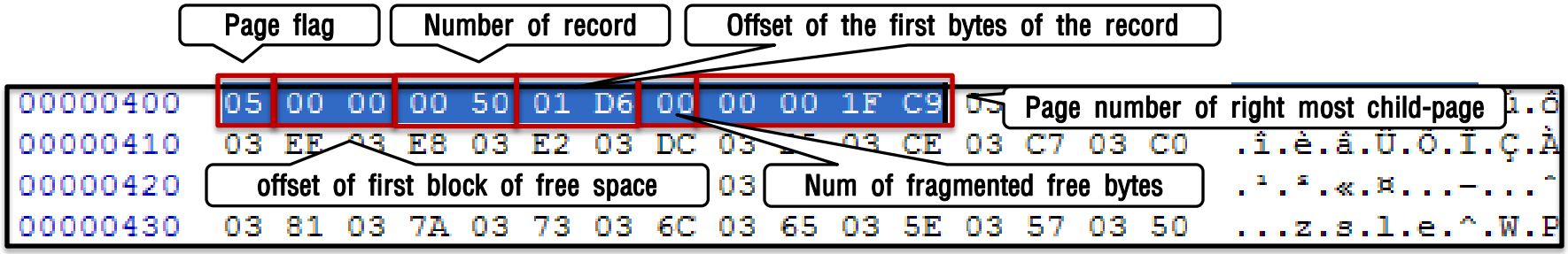
Page Structure



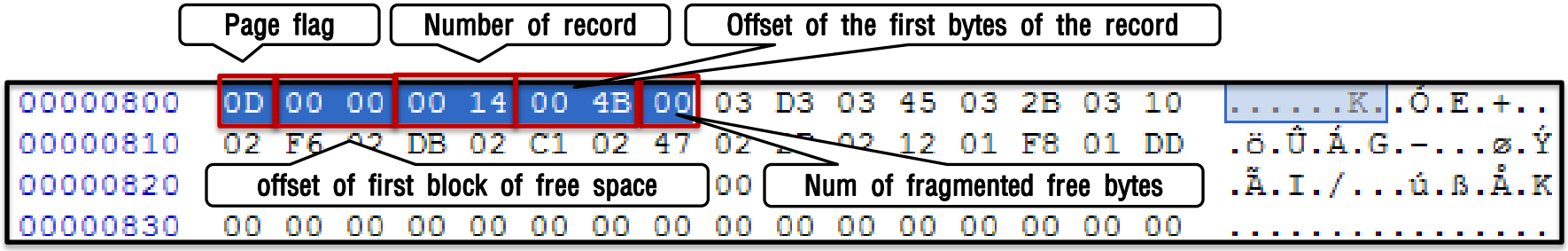
- Page Header
  - Table – b-tree
    - Offset 0
      - 0x05 - Internal Nodeleaf
      - 0x0D – Leaf Node
    - Size
      - 12 Byte – Internal Node pages
      - 8 Byte - Leaf node pages
- Cell Offset
  - 2byte Big endian integer

Page Structure

- Internal Page header



- Leaf Page header



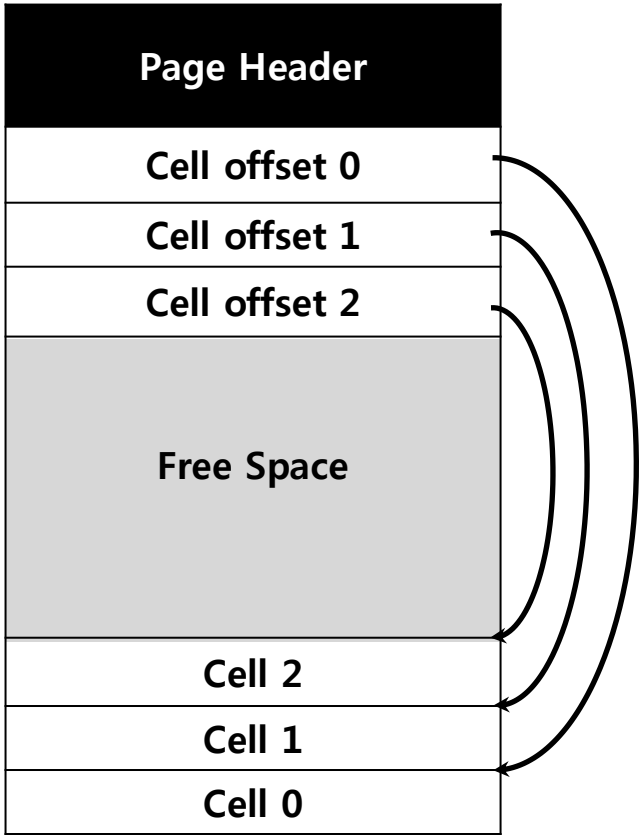
Deleted area in database file

- **Unallocated Area(Free space & Free block)**

Unallocated Area

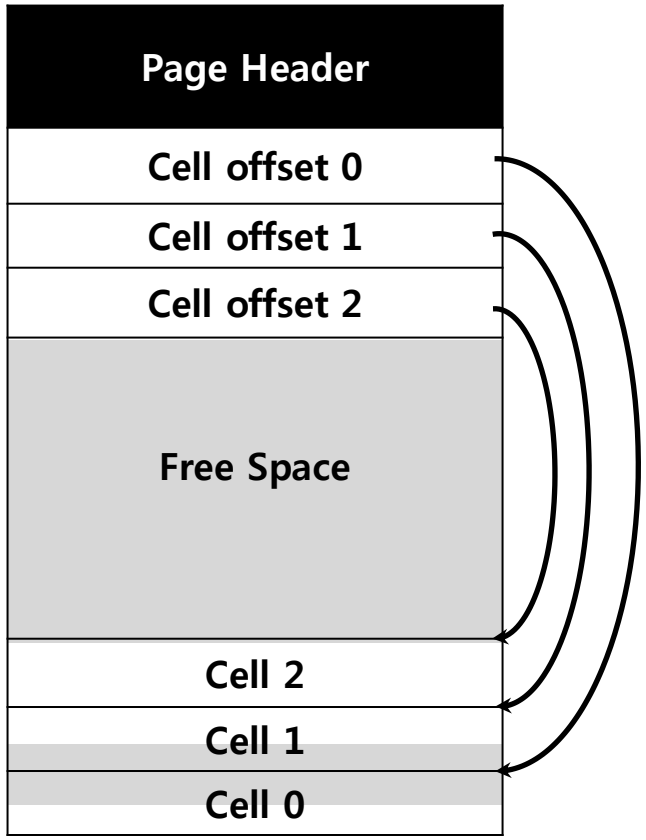
- **Free Space(비할당 공간)**

  - 아직 셀(레코드)이 할당되지 않은 영역
  - 셀 오프셋과 마지막 셀 사이의 공간



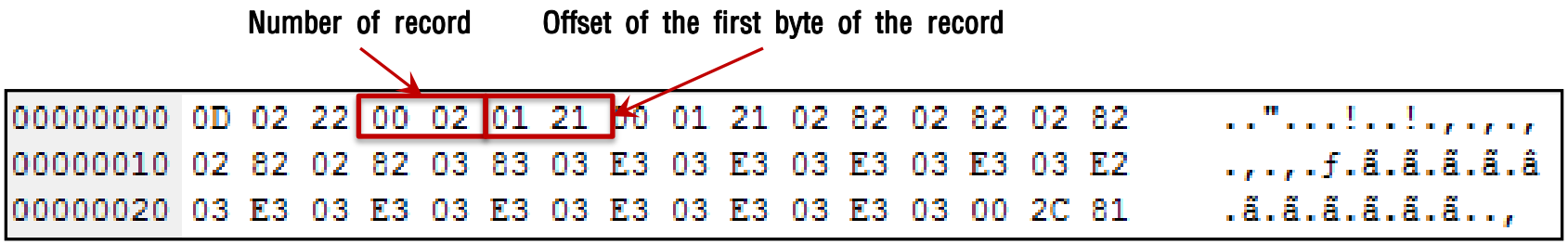
Unallocated Space

- **Free Block**
  - 셀이 할당되었다가 삭제된 영역
  - 셀과 셀 사이에 존재



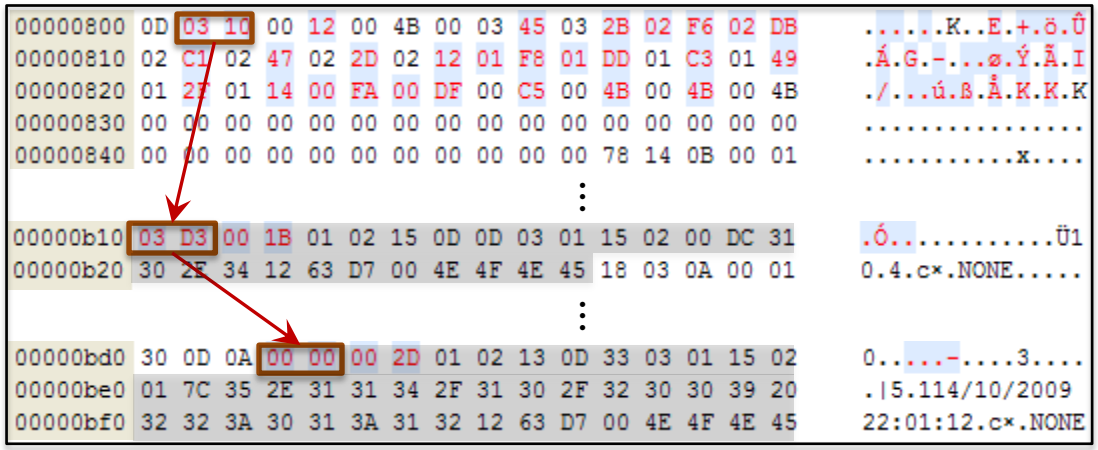
Deleted area

- 삭제된 영역 수집
  - 비할당 영역(Free Area)의 삭제 데이터
    - ✓ 헤더와 오프셋 배열을 건너 뛴 후 페이지의 첫 번째 레코드가 나올 때까지 탐색
      - 기본적으로 Free Area는 0으로 세팅되어 있음
      - 따라서 0이 아닌 값이 존재하는 경우 삭제된 데이터로 판단 가능.



Deleted area

- 삭제된 영역 수집
  - 비할당 블록(Free Block)의 삭제 데이터
    - ✓ 비할당 블록의 데이터는 무조건 삭제된 데이터로 판단 가능
    - ✓ 비할당 블록 체인 (Chain of free block)

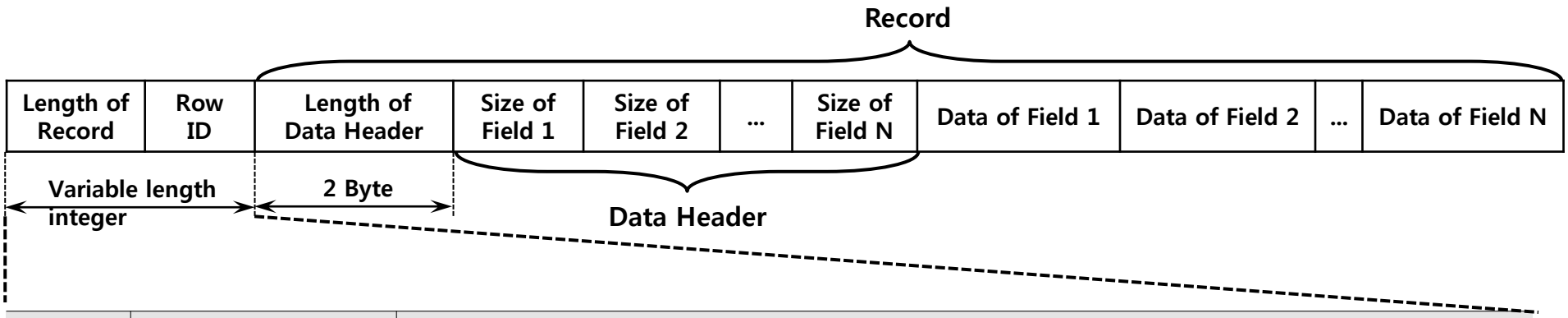


Data Recovery

- Normal Cell & Deleted Cell
- Record Recovery

Cell structure

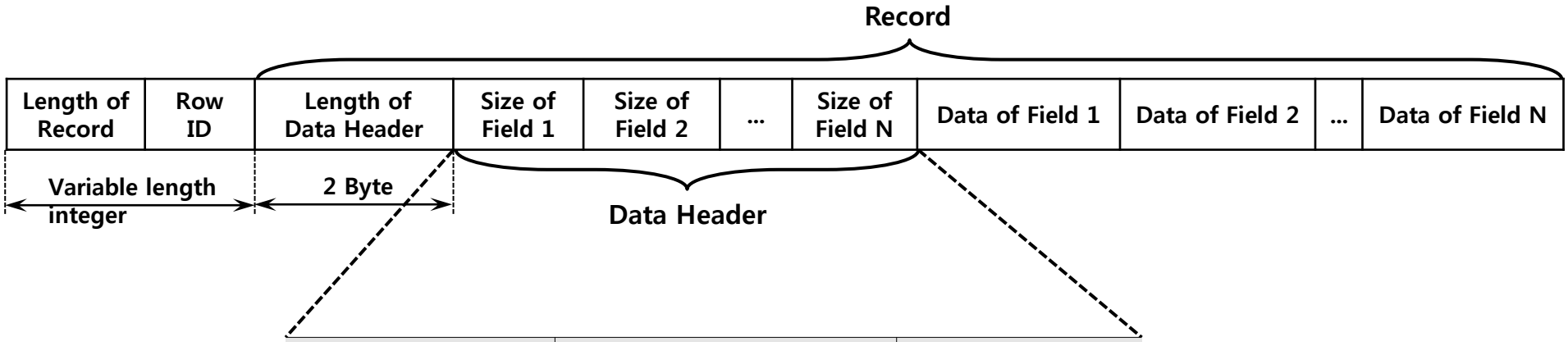
- **Variable length integer**



Byte	Value range	Bit pattern
1	7 bit	0XXXXXXX
2	14 bit	1XXXXXXX 0XXXXXXX
3	21 bit	1XXXXXXX 1XXXXXXX 0XXXXXXX
4	28 bit	1XXXXXXX 1XXXXXXX 1XXXXXXX 0XXXXXXX
5	35 bit	1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 0XXXXXXX
6	42 bit	1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 0XXXXXXX
7	49 bit	1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 0XXXXXXX
8	56 bit	1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 0XXXXXXX
9	64 bit	1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX 1XXXXXXX XXXXXXXX

Cell structure

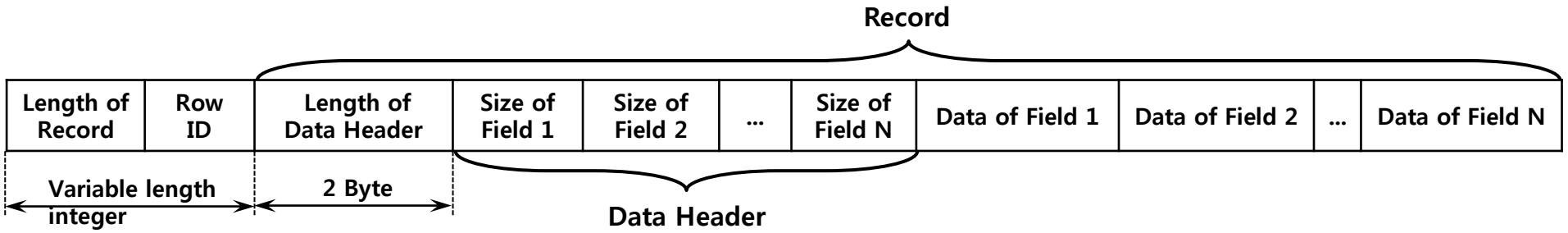
- **Data Header**



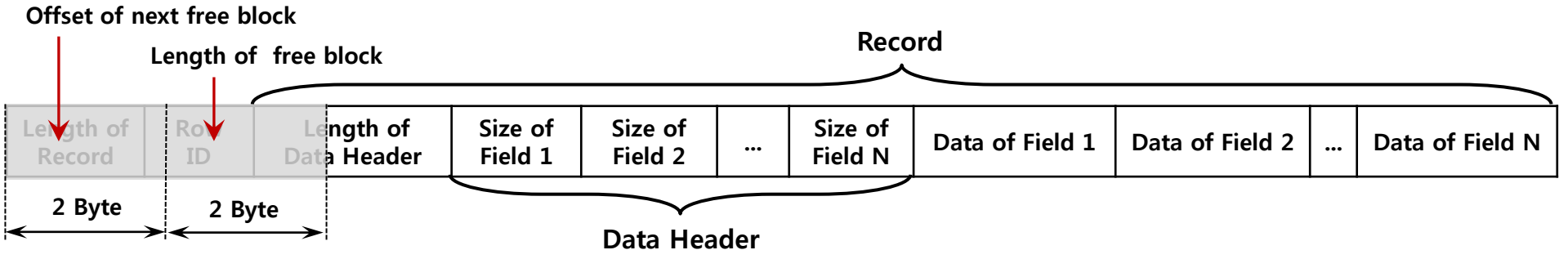
Value	Data Type	Data Size
0	NULL	0
N (N=1-4)	Signed Integer	N
5		6
6	[illegible]	8
7	IEEE float	8
8-11	Reserved	
N>12 (N:even)	BLOB	(N-12)/2
N>13 (N:odd)	TEXT	(N-13)/2

Cell structure

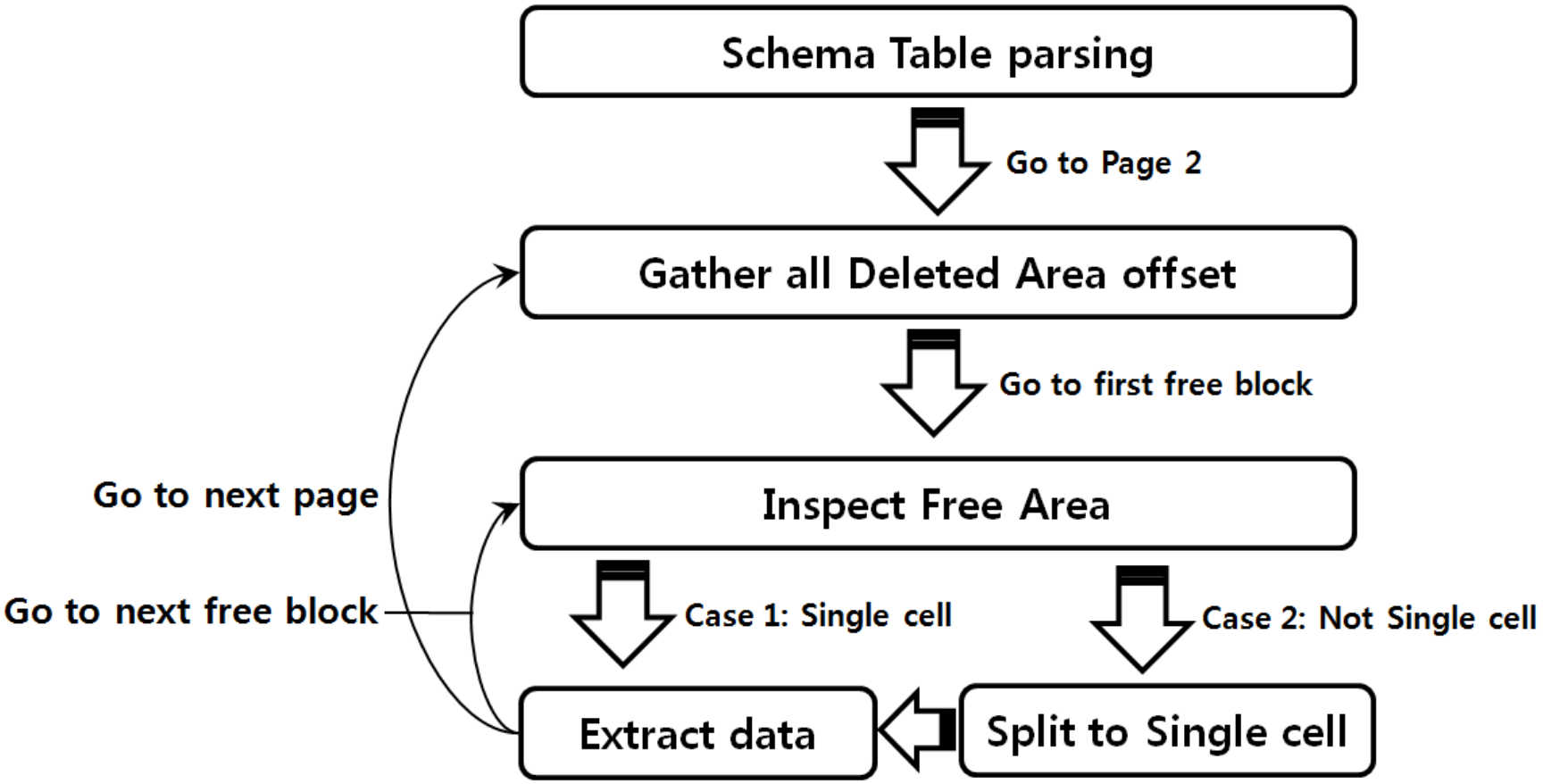
- **Normal Cell**



- **Deleted Cell**



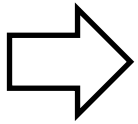
Record carving



Inspect Free Area

- simulation

A	B	C	D
Click here to define a filter			
1	10000000000	17	one
2	13245	34	two
3	9876854	51	three
4	99	51	four



A	B	C	D
Click here to define a filter			
2	13245	34	two
4	99	51	four

- Schema table

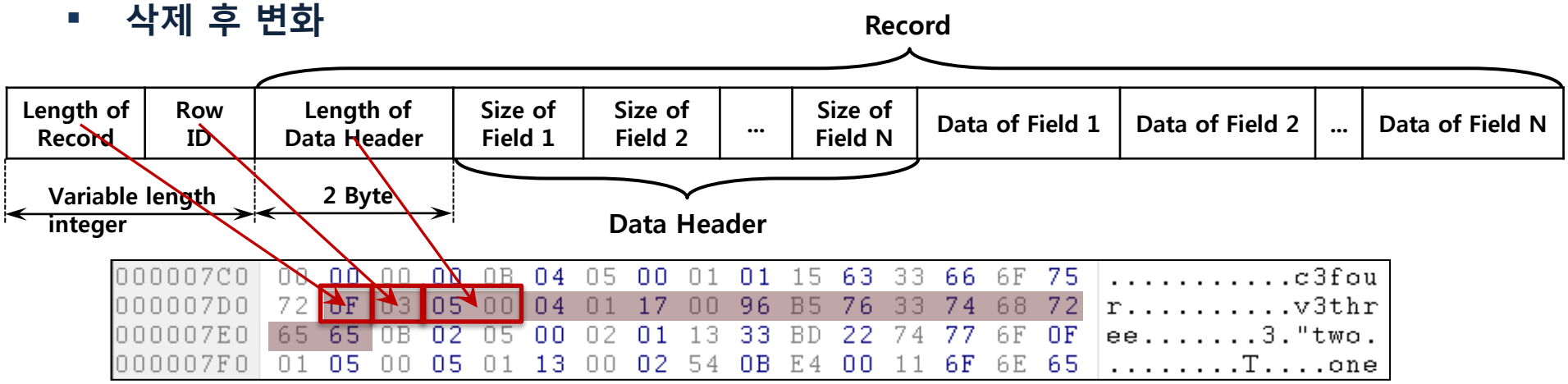
Index	Name	Declared Type
1	A	INTEGER
2	B	INTEGER
3	C	INTEGER
4	D	VARCHAR(256)

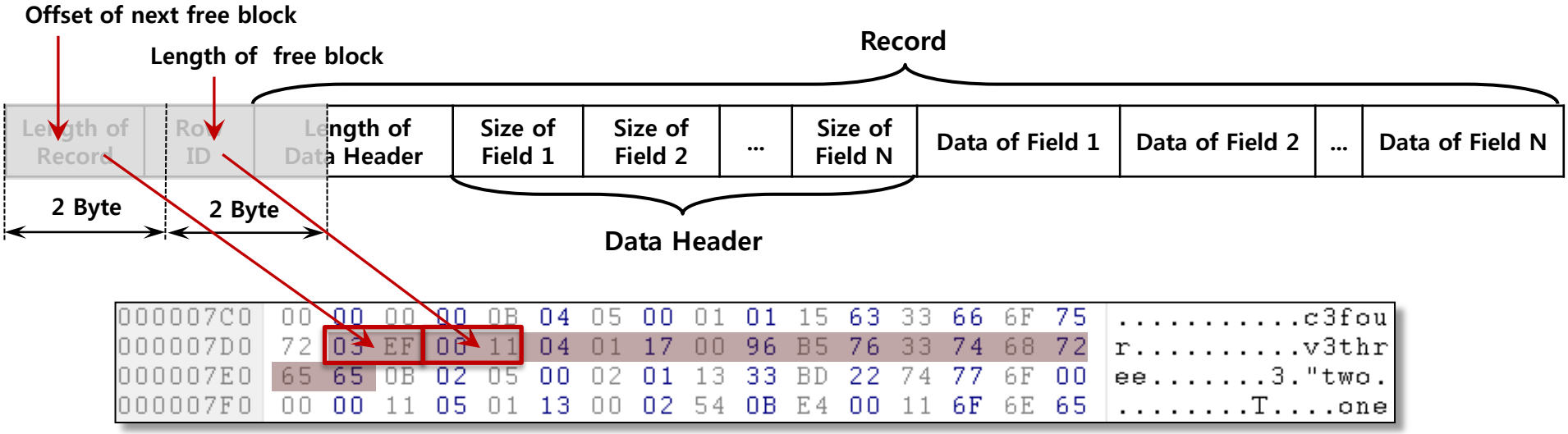
- Target data

```
000007C0  00 00 00 00 0B 04 05 00 01 01 15 63 33 66 6F 75  ...........c3fou
000007D0  72 03 EF 00 11 04 01 17 00 96 B5 76 33 74 68 72  r..........v3thr
000007E0  65 65 0B 02 05 00 02 01 13 33 BD 22 74 77 6F 00  ee.......3."two.
000007F0  00 00 11 05 01 13 00 02 54 0B E4 00 11 6F 6E 65  ........T....one
```

Inspect Free Area

- 삭제 후 변화



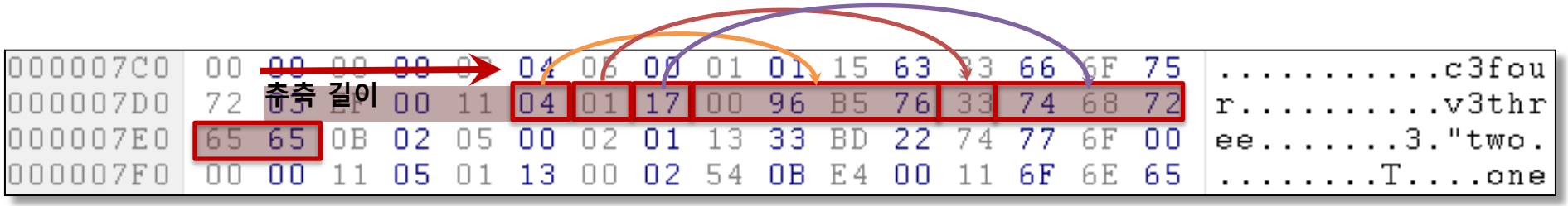


Inspect Free Area

- **Values of Data header**

Value	Data Type	Data Size
0	NULL	0
N (N=1-4)	Signed Integer	N
5	-- --- - ------ ---	6
6		8
7	IEEE float	8
8-11	Reserved	
N>12 (N:even)	BLOB	(N-12)/2
N>13 (N:odd)	TEXT	(N-13)/2

- **Target data**
  - 스키마의 데이터 헤더와 일치하는 점을 찾은 지점에서 복원 진행
  - 스키마가 복잡할 수록 오탐은 적어짐.



Thank you for listenig