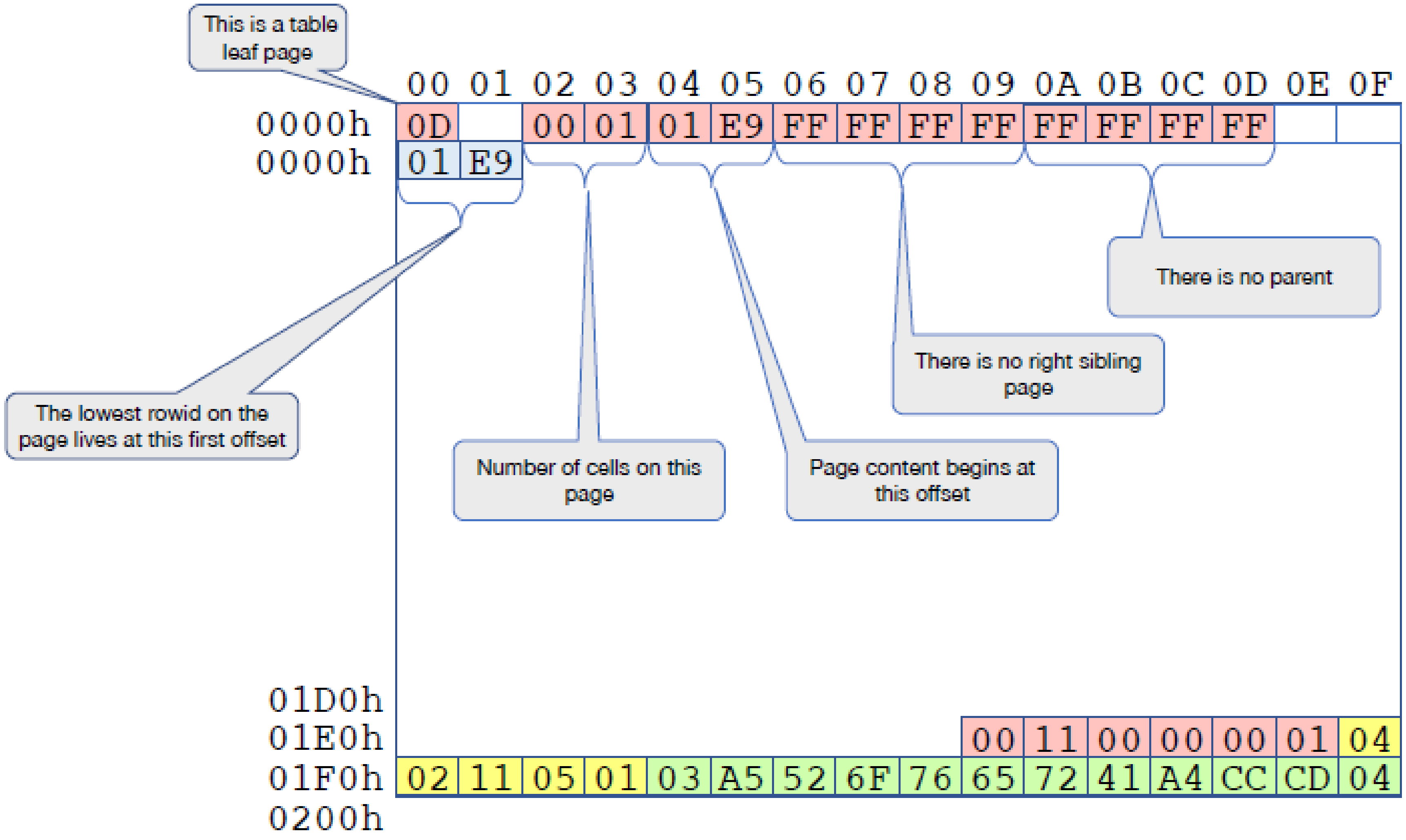
Database design guide & Storage mapping

Page Format



```
       00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0000h  0D    00 01 01 E9 FF FF FF FF FF FF FF FF
0000h  01 E9
01D0h
01E0h                             00 11 00 00 00 01 04
01F0h  02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04
0200h
```

This is a table leaf page

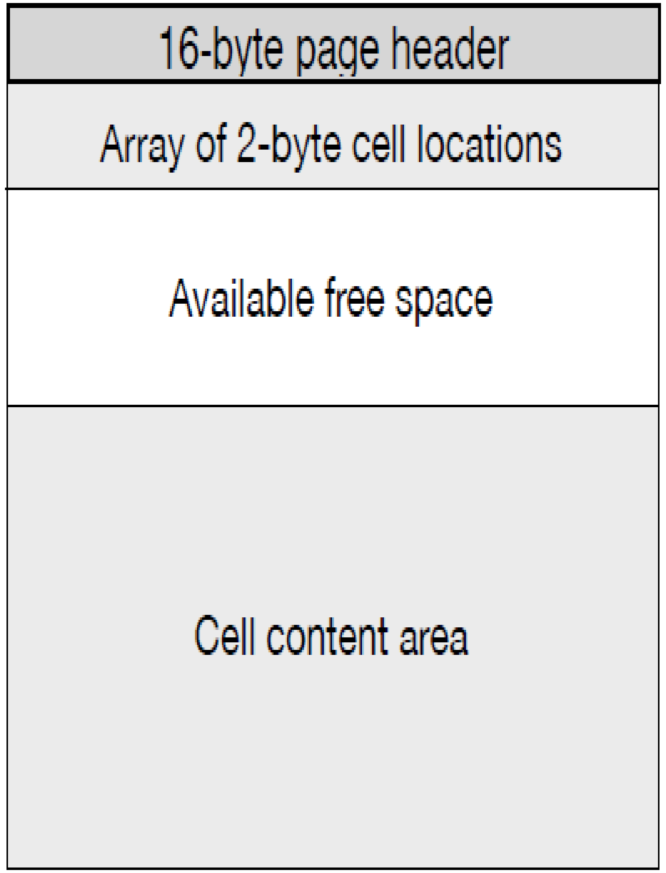
The lowest rowid on the page lives at this first offset

Number of cells on this page

Page content begins at this offset

There is no right sibling page

There is no parent

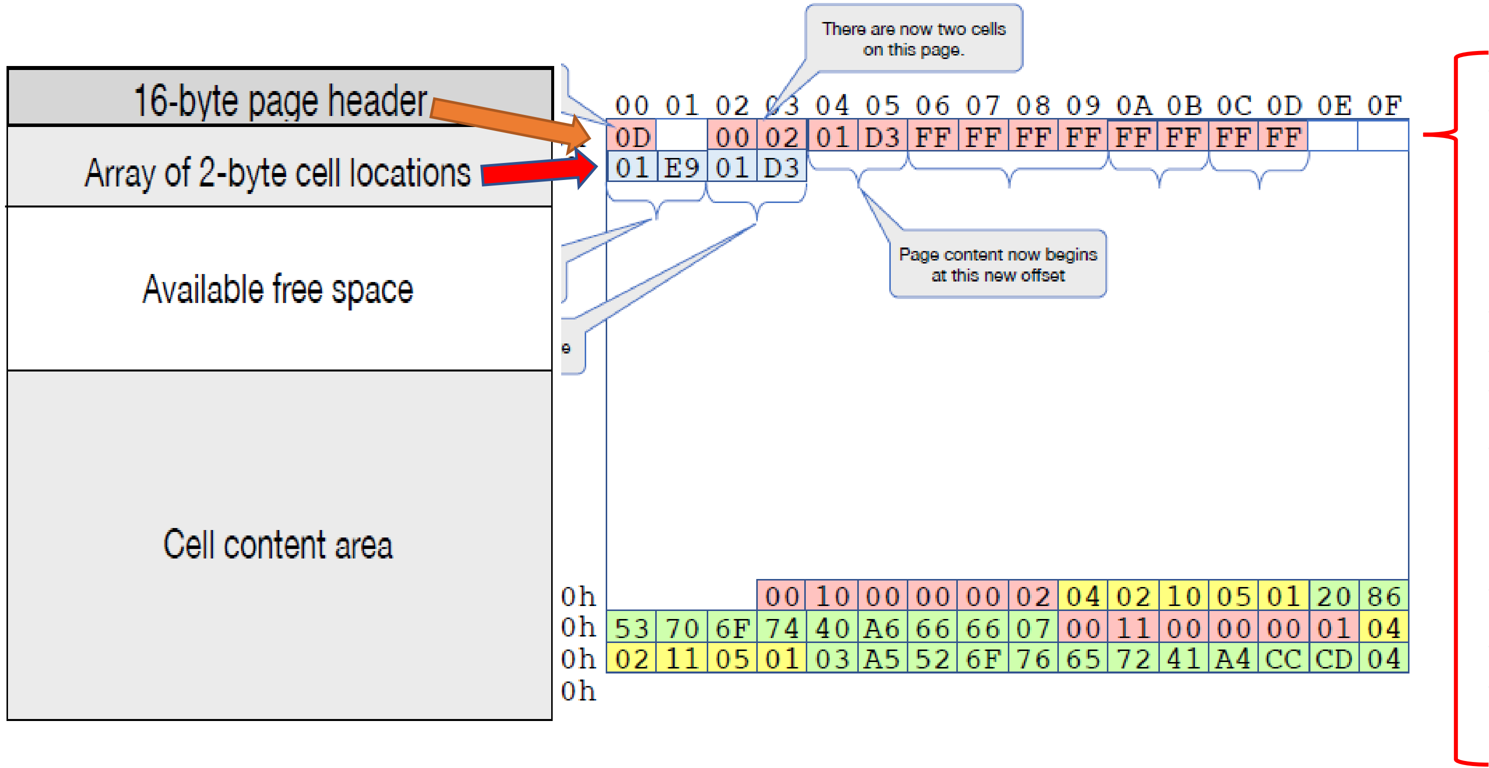


16-byte page header

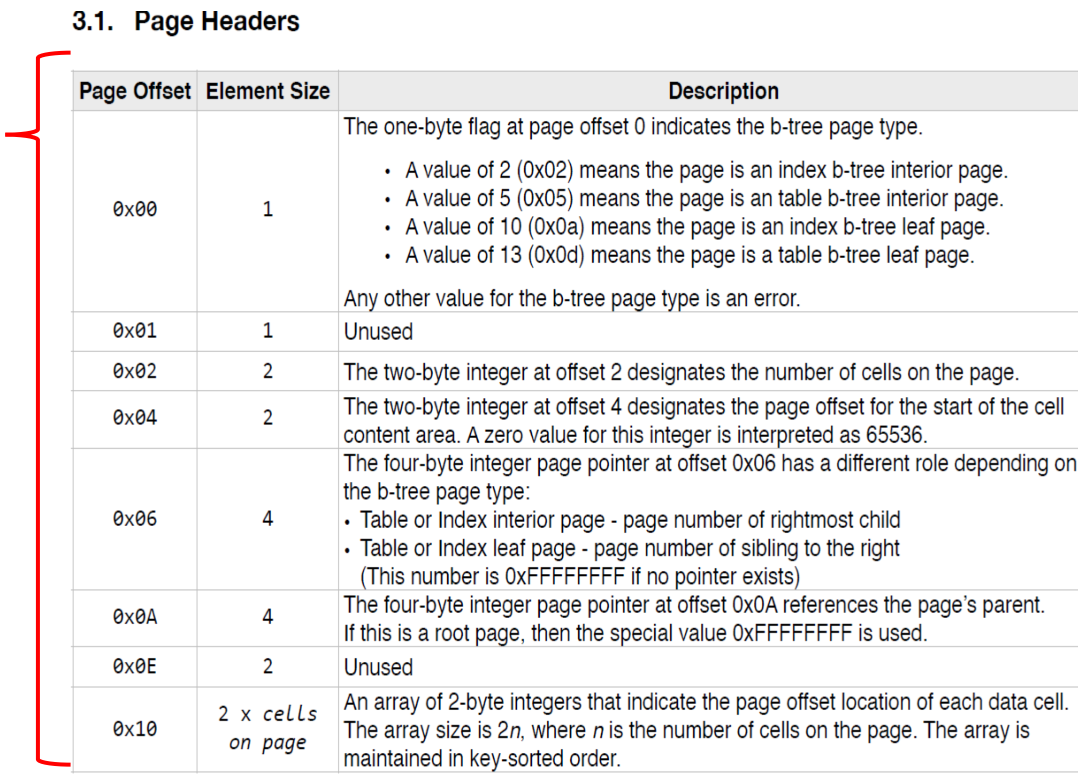
Array of 2-byte cell locations

Available free space

Cell content area


16-byte page header
Array of 2-byte cell locations
Available free space
Cell content area
There are now two cells on this page.
Page content now begins at this new offset
e

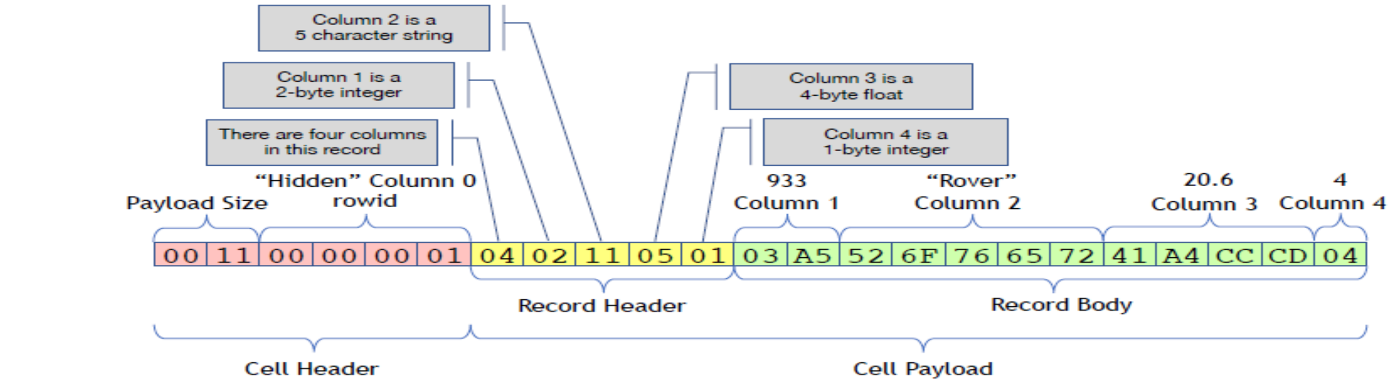
```
    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
    0D    00 02 01 D3 FF FF FF FF FF FF FF FF
    01 E9 01 D3

0h           00 10 00 00 00 02 04 02 10 05 01 20 86
0h  53 70 6F 74 40 A6 66 66 07 00 11 00 00 00 01 04
0h  02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04
0h
```



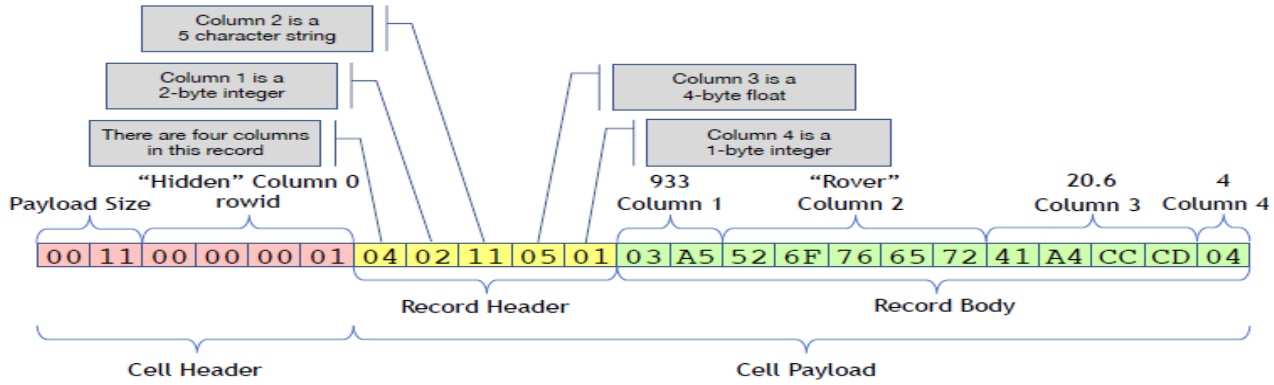
3.1. Page Headers

Page Offset	Element Size	Description
0x00	1	The one-byte flag at page offset 0 indicates the b-tree page type. • A value of 2 (0x02) means the page is an index b-tree interior page. • A value of 5 (0x05) means the page is an table b-tree interior page. • A value of 10 (0x0a) means the page is an index b-tree leaf page. • A value of 13 (0x0d) means the page is a table b-tree leaf page. Any other value for the b-tree page type is an error.
0x01	1	Unused
0x02	2	The two-byte integer at offset 2 designates the number of cells on the page.
0x04	2	The two-byte integer at offset 4 designates the page offset for the start of the cell content area. A zero value for this integer is interpreted as 65536.
0x06	4	The four-byte integer page pointer at offset 0x06 has a different role depending on the b-tree page type: • Table or Index interior page - page number of rightmost child • Table or Index leaf page - page number of sibling to the right (This number is 0xFFFFFFFF if no pointer exists)
0x0A	4	The four-byte integer page pointer at offset 0x0A references the page's parent. If this is a root page, then the special value 0xFFFFFFFF is used.
0x0E	2	Unused
0x10	2 x cells on page	An array of 2-byte integers that indicate the page offset location of each data cell. The array size is 2*n*, where *n* is the number of cells on the page. The array is maintained in key-sorted order.

First Record Format Insert


Column 2 is a 5 character string
Column 1 is a 2-byte integer
There are four columns in this record
Column 3 is a 4-byte float
Column 4 is a 1-byte integer
"Hidden" Column 0 rowid
Payload Size
933 Column 1
"Rover" Column 2
20.6 Column 3
4 Column 4
00 11 00 00 00 01 04 02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04
Record Header
Record Body
Cell Header
Cell Payload

First Record Format Insert



Column 2 is a 5 character string

Column 1 is a 2-byte integer

There are four columns in this record

Column 3 is a 4-byte float

Column 4 is a 1-byte integer

Payload Size

"Hidden" Column 0 rowid

933 Column 1

"Rover" Column 2

20.6 Column 3

4 Column 4

00 11 00 00 00 01 04 02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04

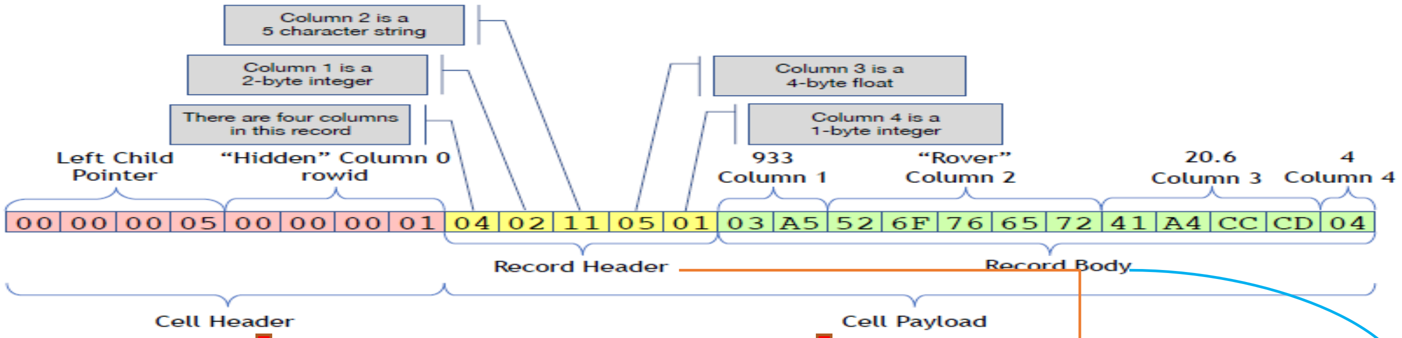
Record Header

Record Body

Cell Header

Cell Payload

Table Interior Cell



Column 2 is a 5 character string

Column 1 is a 2-byte integer

There are four columns in this record

Column 3 is a 4-byte float

Column 4 is a 1-byte integer

Left Child Pointer

"Hidden" Column 0 rowid

933 Column 1

"Rover" Column 2

20.6 Column 3

4 Column 4

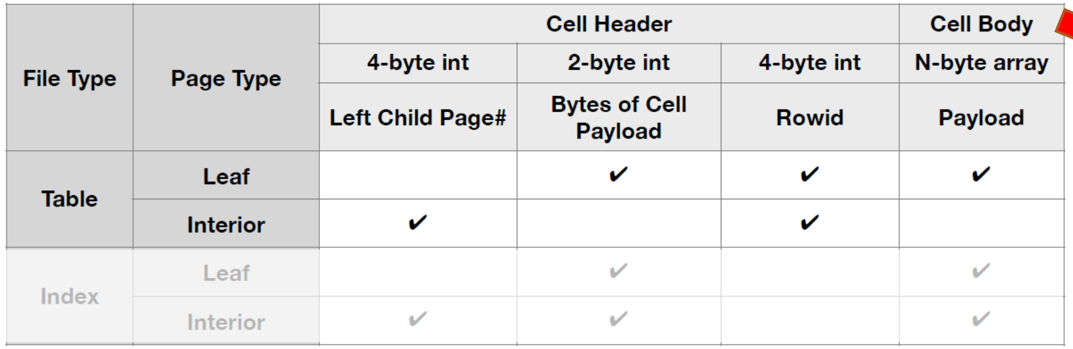
00 00 00 05 00 00 00 01 04 02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04

Record Header

Record Body

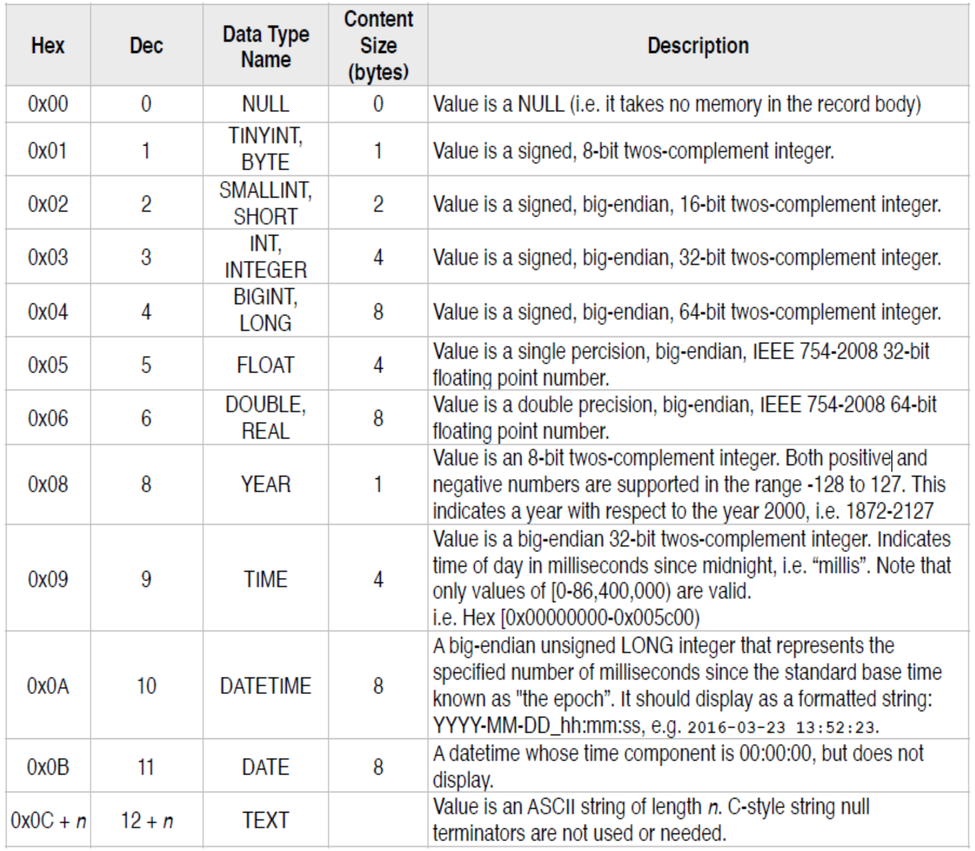
Cell Header

Cell Payload



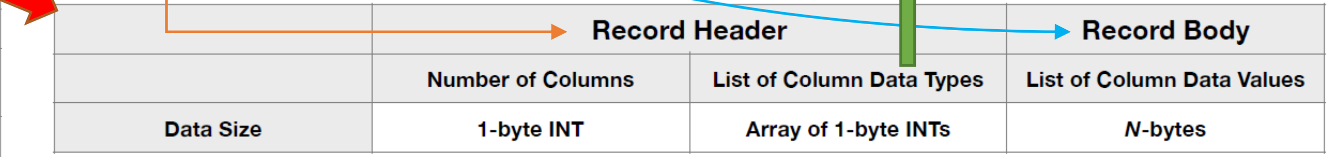
File Type	Page Type	Cell Header			Cell Body
		4-byte int	2-byte int	4-byte int	N-byte array
		Left Child Page#	Bytes of Cell Payload	Rowid	Payload
Table	Leaf		✔	✔	✔
	Interior	✔		✔	
Index	Leaf		✔		✔
	Interior	✔	✔		✔

Figure 2: Cell Formats



Hex	Dec	Data Type Name	Content Size (bytes)	Description
0x00	0	NULL	0	Value is a NULL (i.e. it takes no memory in the record body)
0x01	1	TINYINT, BYTE	1	Value is a signed, 8-bit twos-complement integer.
0x02	2	SMALLINT, SHORT	2	Value is a signed, big-endian, 16-bit twos-complement integer.
0x03	3	INT, INTEGER	4	Value is a signed, big-endian, 32-bit twos-complement integer.
0x04	4	BIGINT, LONG	8	Value is a signed, big-endian, 64-bit twos-complement integer.
0x05	5	FLOAT	4	Value is a single percision, big-endian, IEEE 754-2008 32-bit floating point number.
0x06	6	DOUBLE, REAL	8	Value is a double precision, big-endian, IEEE 754-2008 64-bit floating point number.
0x08	8	YEAR	1	Value is an 8-bit twos-complement integer. Both positive and negative numbers are supported in the range -128 to 127. This indicates a year with respect to the year 2000, i.e. 1872-2127
0x09	9	TIME	4	Value is a big-endian 32-bit twos-complement integer. Indicates time of day in milliseconds since midnight, i.e. "millis". Note that only values of [0-86,400,000) are valid. i.e. Hex [0x00000000-0x005c00)
0x0A	10	DATETIME	8	A big-endian unsigned LONG integer that represents the specified number of milliseconds since the standard base time known as "the epoch". It should display as a formatted string: YYYY-MM-DD_hh:mm:ss, e.g. `2016-03-23 13:52:23`.
0x0B	11	DATE	8	A datetime whose time component is 00:00:00, but does not display.
0x0C + *n*	12 + *n*	TEXT		Value is an ASCII string of length *n*. C-style string null terminators are not used or needed.

Figure 4 - Data Types, Their 1-byte Code, and Implementation

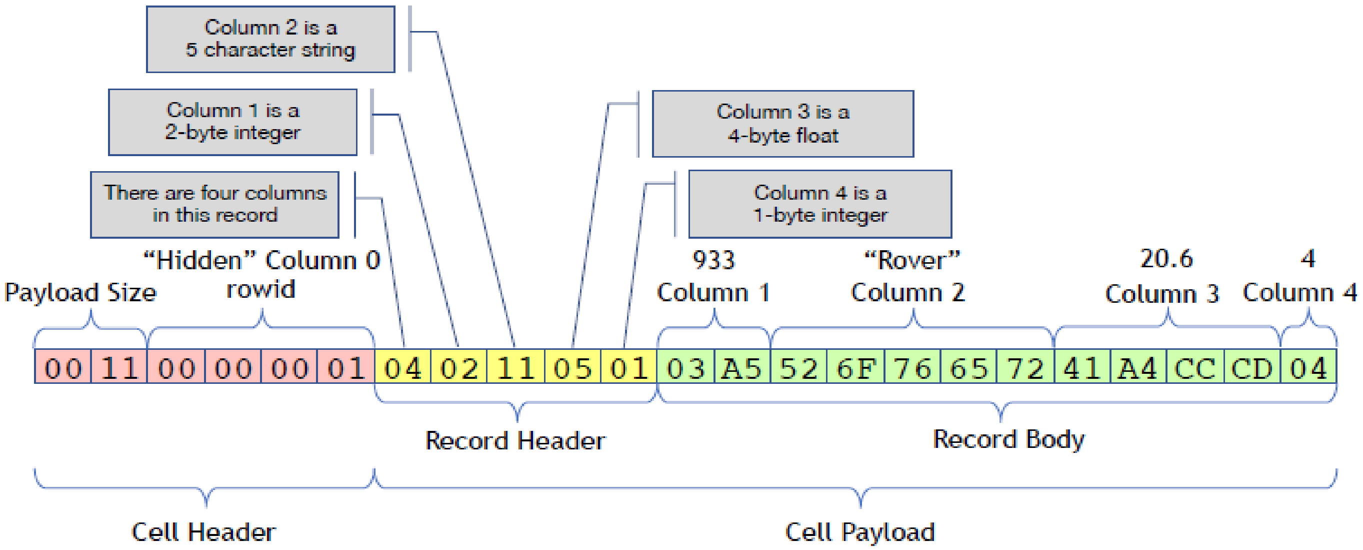


	Record Header		Record Body
	Number of Columns	List of Column Data Types	List of Column Data Values
Data Size	1-byte INT	Array of 1-byte INTs	*N*-bytes

Figure 3: Record Format (i.e. Payload of Table Leaf Cell)

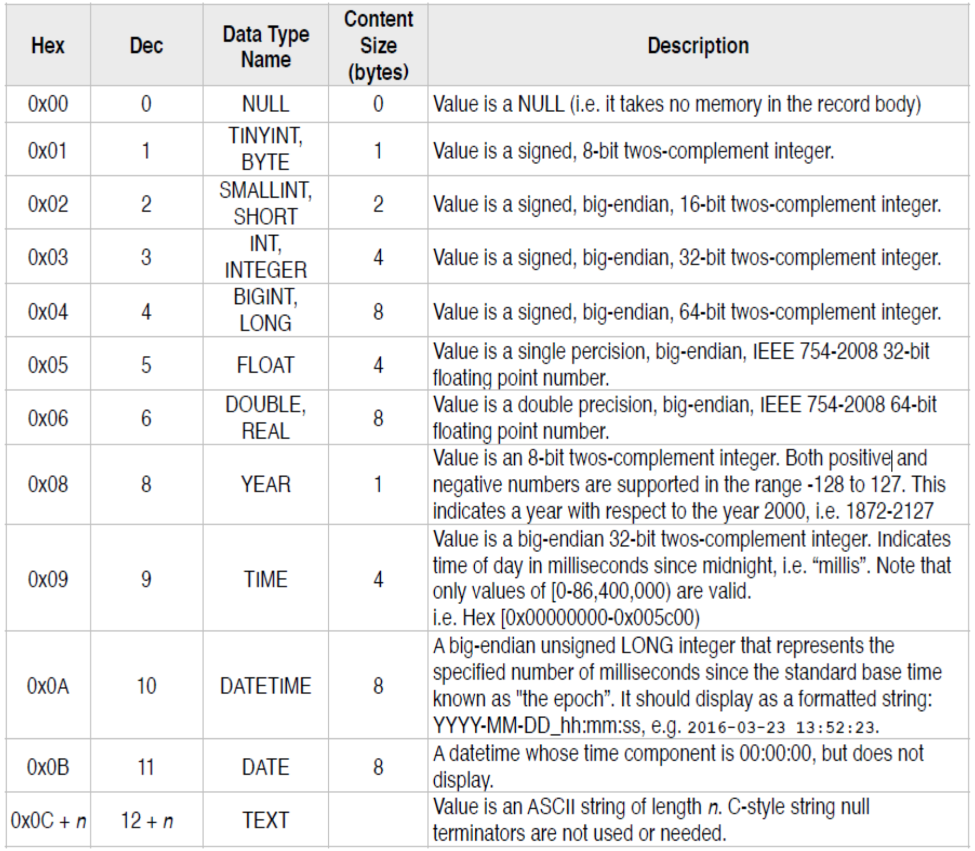
Example

```
• CREATE TABLE Dogs (
        Id         SHORT,
        Name       TEXT,
        Weight     FLOAT,
        Age        BYTE);

 INSERT INTO TABLE Dogs VALUES
 (933,"Rover",20.6,4);
```



```
00 11 00 00 00 01 04 02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04
```

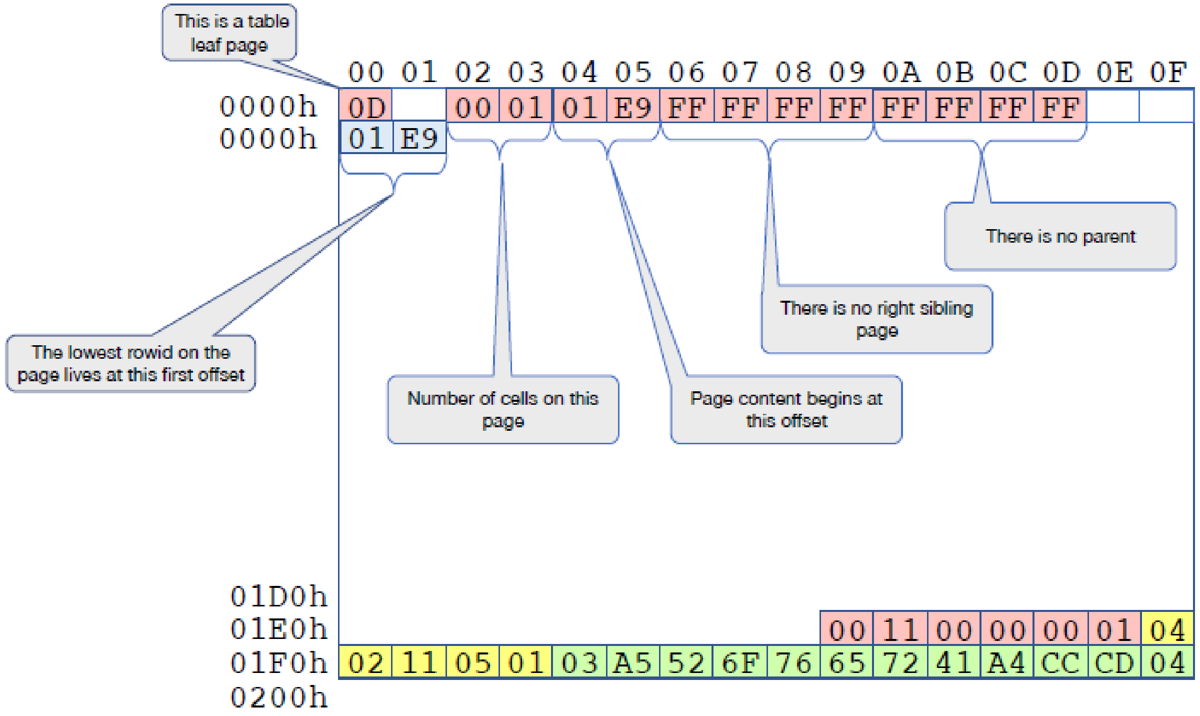
Column 2 is a 5 character string
Column 1 is a 2-byte integer
There are four columns in this record
Column 3 is a 4-byte float
Column 4 is a 1-byte integer
Payload Size
"Hidden" Column 0
rowid
933
Column 1
"Rover"
Column 2
20.6
Column 3
4
Column 4
Record Header
Record Body
Cell Header
Cell Payload



Hex	Dec	Data Type Name	Content Size (bytes)	Description
0x00	0	NULL	0	Value is a NULL (i.e. it takes no memory in the record body)
0x01	1	TINYINT, BYTE	1	Value is a signed, 8-bit twos-complement integer.
0x02	2	SMALLINT, SHORT	2	Value is a signed, big-endian, 16-bit twos-complement integer.
0x03	3	INT, INTEGER	4	Value is a signed, big-endian, 32-bit twos-complement integer.
0x04	4	BIGINT, LONG	8	Value is a signed, big-endian, 64-bit twos-complement integer.
0x05	5	FLOAT	4	Value is a single percision, big-endian, IEEE 754-2008 32-bit floating point number.
0x06	6	DOUBLE, REAL	8	Value is a double precision, big-endian, IEEE 754-2008 64-bit floating point number.
0x08	8	YEAR	1	Value is an 8-bit twos-complement integer. Both positive and negative numbers are supported in the range -128 to 127. This indicates a year with respect to the year 2000, i.e. 1872-2127
0x09	9	TIME	4	Value is a big-endian 32-bit twos-complement integer. Indicates time of day in milliseconds since midnight, i.e. "millis". Note that only values of [0-86,400,000) are valid. i.e. Hex [0x00000000-0x005c00)
0x0A	10	DATETIME	8	A big-endian unsigned LONG integer that represents the specified number of milliseconds since the standard base time known as "the epoch". It should display as a formatted string: YYYY-MM-DD_hh:mm:ss, e.g. `2016-03-23 13:52:23`.
0x0B	11	DATE	8	A datetime whose time component is 00:00:00, but does not display.
0x0C + *n*	12 + *n*	TEXT		Value is an ASCII string of length *n*. C-style string null terminators are not used or needed.

Figure 4 - Data Types, Their 1-byte Code, and Implementation

Example



	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000h	0D		00	01	01	E9	FF	FF	FF	FF	FF	FF	FF	FF		
0000h	01	E9														
01D0h																
01E0h										00	11	00	00	00	01	04
01F0h	02	11	05	01	03	A5	52	6F	76	65	72	41	A4	CC	CD	04
0200h																

This is a table leaf page

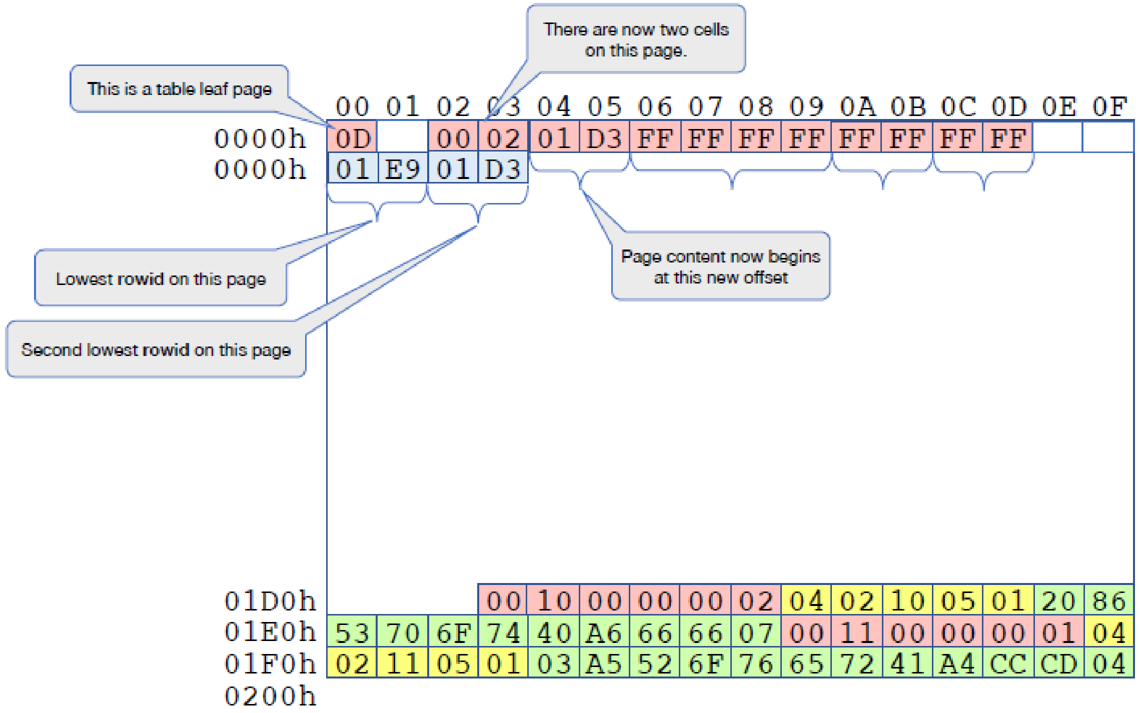
The lowest rowid on the page lives at this first offset

Number of cells on this page

Page content begins at this offset

There is no right sibling page

There is no parent



	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000h	0D		00	02	01	D3	FF	FF	FF	FF	FF	FF	FF	FF		
0000h	01	E9	01	D3												
01D0h				00	10	00	00	00	02	04	02	10	05	01	20	86
01E0h	53	70	6F	74	40	A6	66	66	07	00	11	00	00	00	01	04
01F0h	02	11	05	01	03	A5	52	6F	76	65	72	41	A4	CC	CD	04
0200h																

There are now two cells on this page.

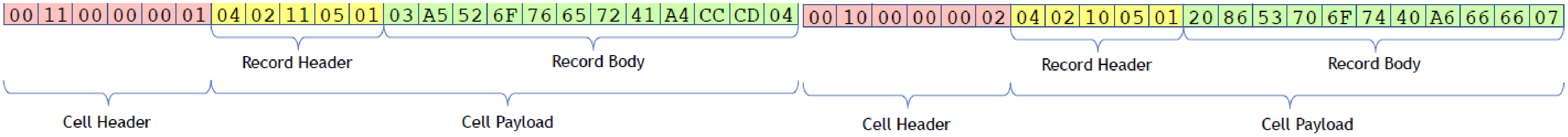
This is a table leaf page

Lowest rowid on this page

Second lowest rowid on this page

Page content now begins at this new offset

Similarity to Genes ?



```
00 11 00 00 00 01 04 02 11 05 01 03 A5 52 6F 76 65 72 41 A4 CC CD 04 00 10 00 00 00 02 04 02 10 05 01 20 86 53 70 6F 74 40 A6 66 66 07
```

Record Header

Record Body

Cell Header

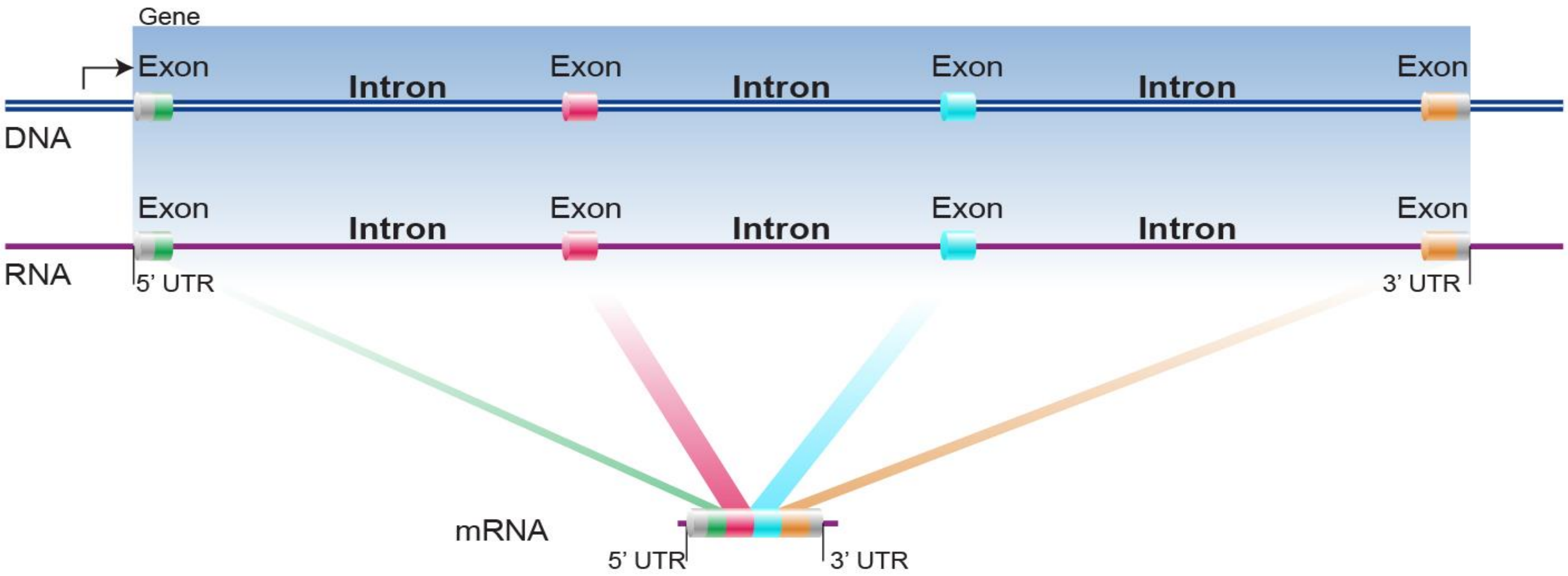
Cell Payload

Record Header

Record Body

Cell Header

Cell Payload



Gene
Exon
Intron
Exon
Intron
Exon
Intron
Exon
DNA
Exon
Intron
Exon
Intron
Exon
Intron
Exon
RNA
5' UTR
3' UTR
mRNA
5' UTR
3' UTR