

SQL Server Databaseology: A Deep Dive Into Database Internals

John Morehouse
Consultant

Denny Cherry & Associates Consulting



john@dcac.com



<http://linkedin.com/in/sqlrus>



@SqlRUs



<http://www.sqlrus.com>

Who Am I?

- Leader of the Louisville SQL Server/Power BI User Group
- Organizer/Speaker of SQL Saturday's & other conferences
- Heavily involved with SQL PASS
- Microsoft Data Platform MVP
- Friend of Redgate 2015 - 2019
- Idera ACE 2016
- SentryOne Product Advisory Council



Where Am I?



The vetted and certified experts at Denny Cherry and Associates Consulting assist companies with attaining IT goals such as HA, scalability, SQL Server virtualization, migration, and acceleration reliably while finding ways to save on costs. With clients ranging from Fortune 50 corporations to small businesses, their commitment to each is the same: to provide a deft, high-speed IT environment that leverages every aspect of their platform: from architecture, infrastructure, to network.

Visit DCAC at <http://www.dcac.co>

FEEDBACK FORMS

PLEASE FILL OUT AND PASS TO YOUR ROOM
HELPER BEFORE YOU LEAVE THE SESSION



[This Photo by Unknown Author is licensed under CC BY-NC-ND](#)

GitHub

Slide deck and script demos
for this presentation can be
downloaded from Github

<http://bit.ly/databaseology>



GitHub

Agenda

1. SQL Server Databaseology
2. Records
3. Pages
4. Extents
5. Tools
6. Demo
7. Pulling it together



What is Databaseology?

data·base·ol·o·gy

noun

the science concerned with
the study of databases and
their structures.

Why? Why? Why?



Overall Structure

Records = Rows =
Slots

Records live in
Pages

Groups of 8 pages
is an Extent

Overall
Structure

Records =
Rows = Slots

Everything is stored in a record, somewhere..

Data

Forwarding

Index

Versioned Records

Ghost Records

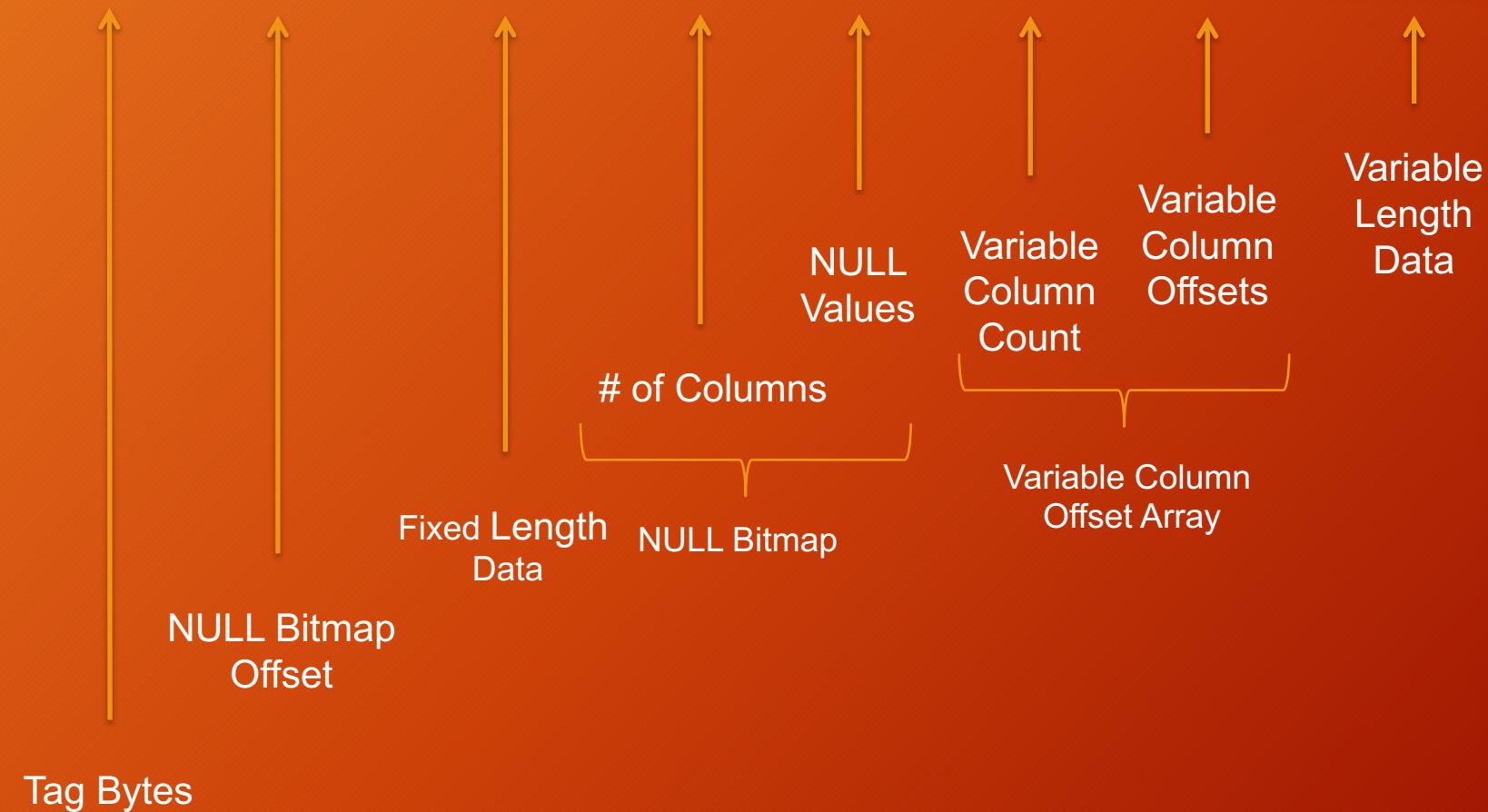
Large Object (LOB)

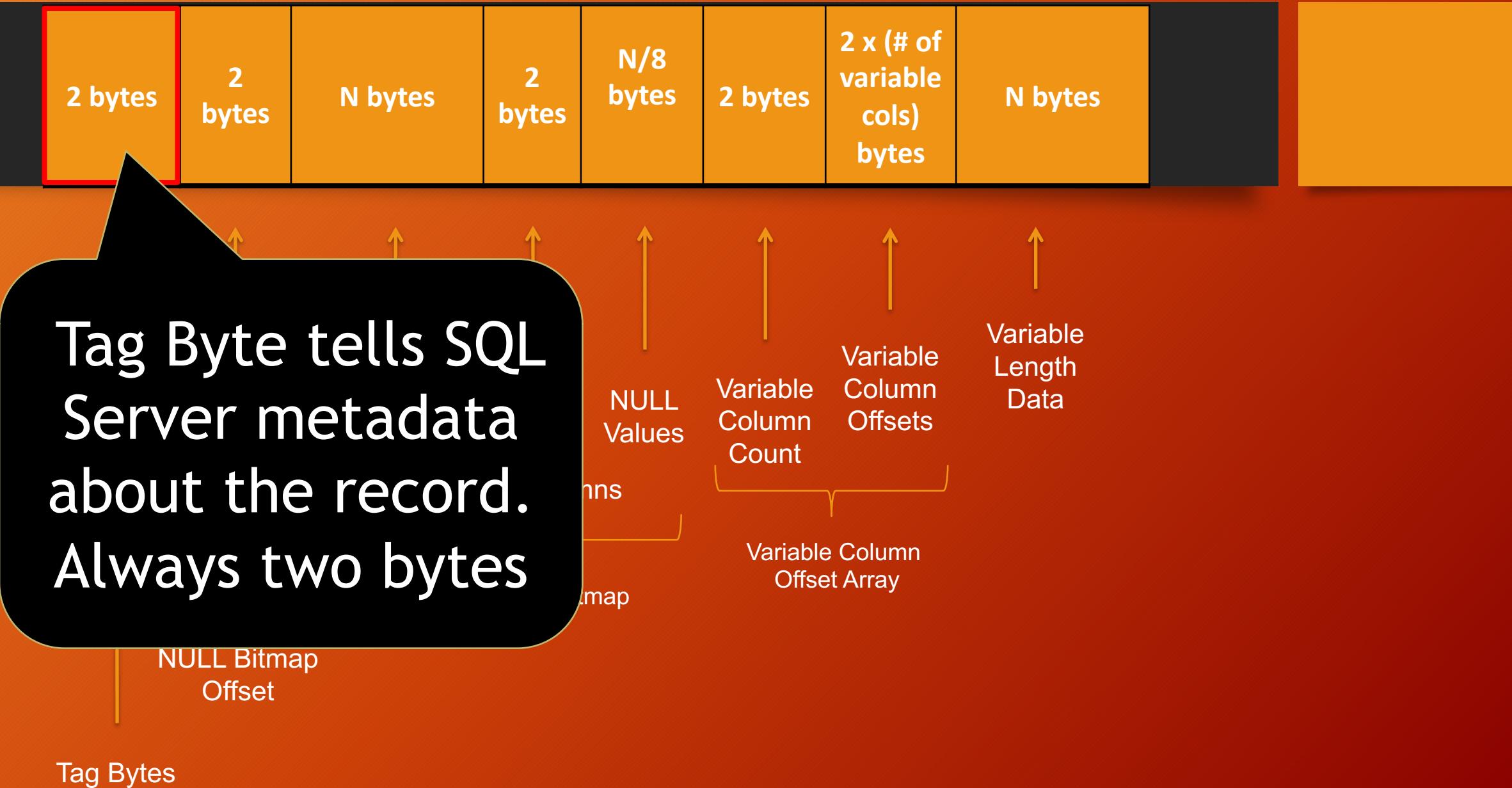
And so on...

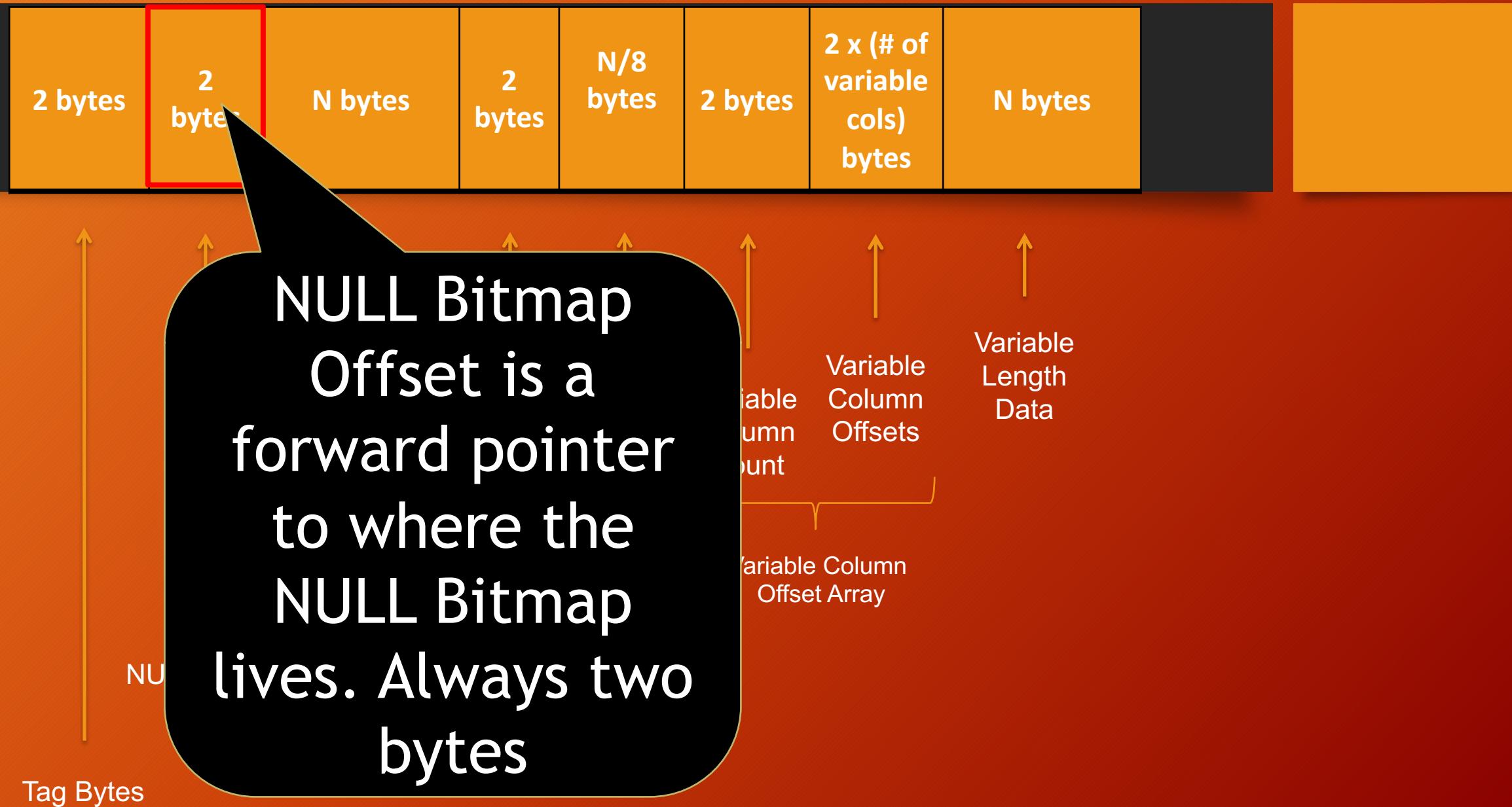
Data Records

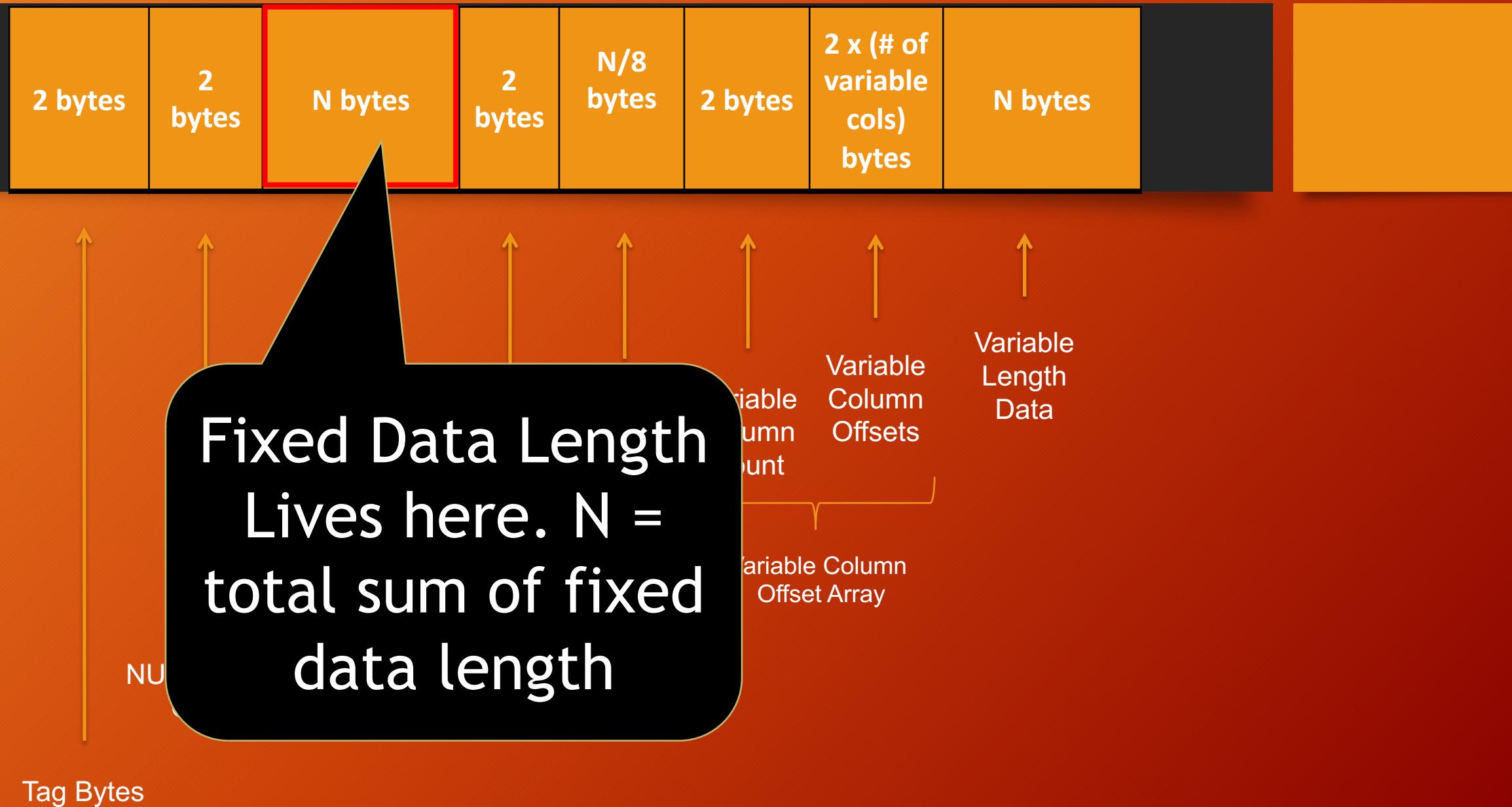
- Records that actually store your data!
- Data Records store two basic types of data
 - Fixed Length
 - Variable Length

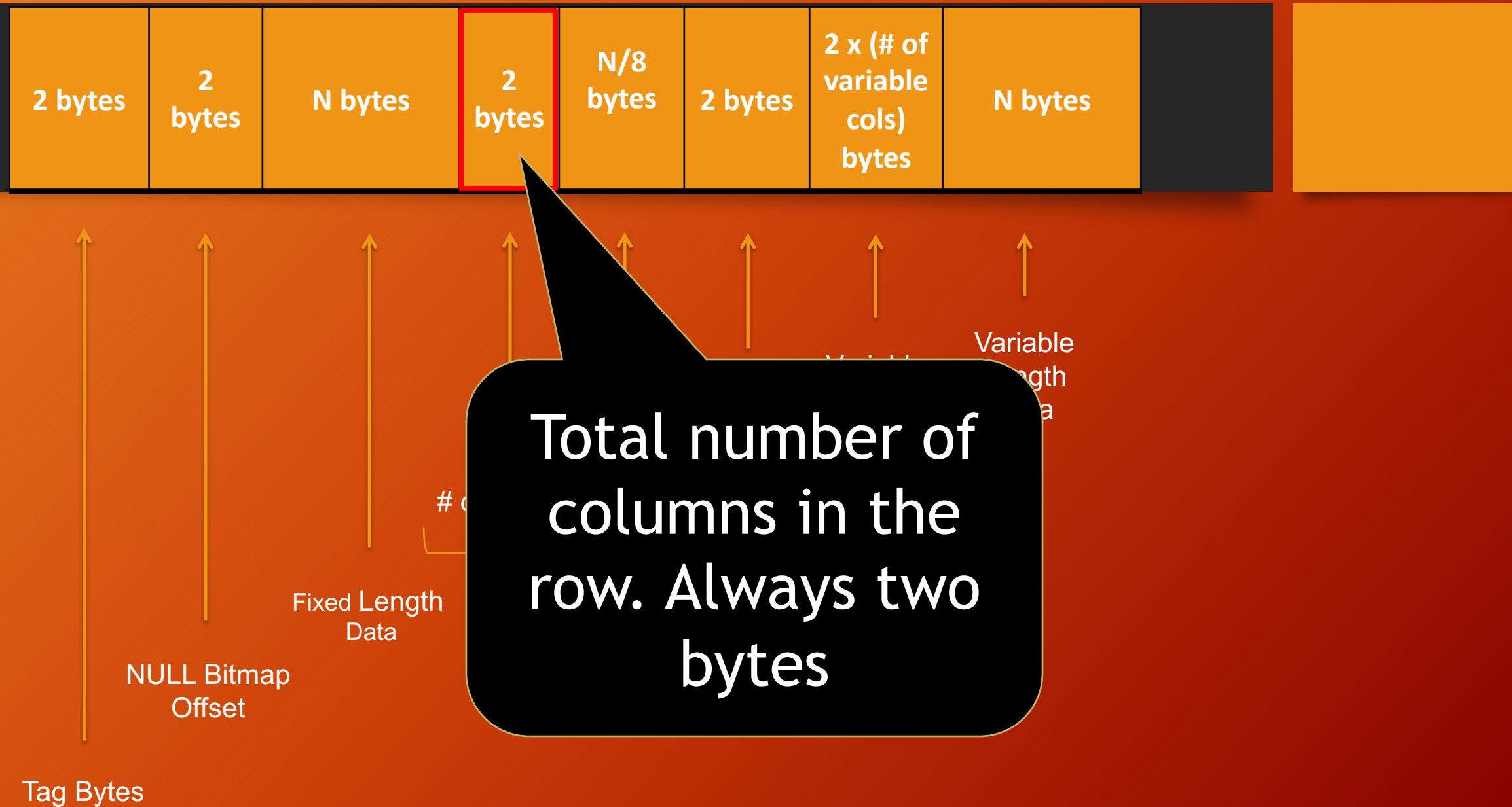


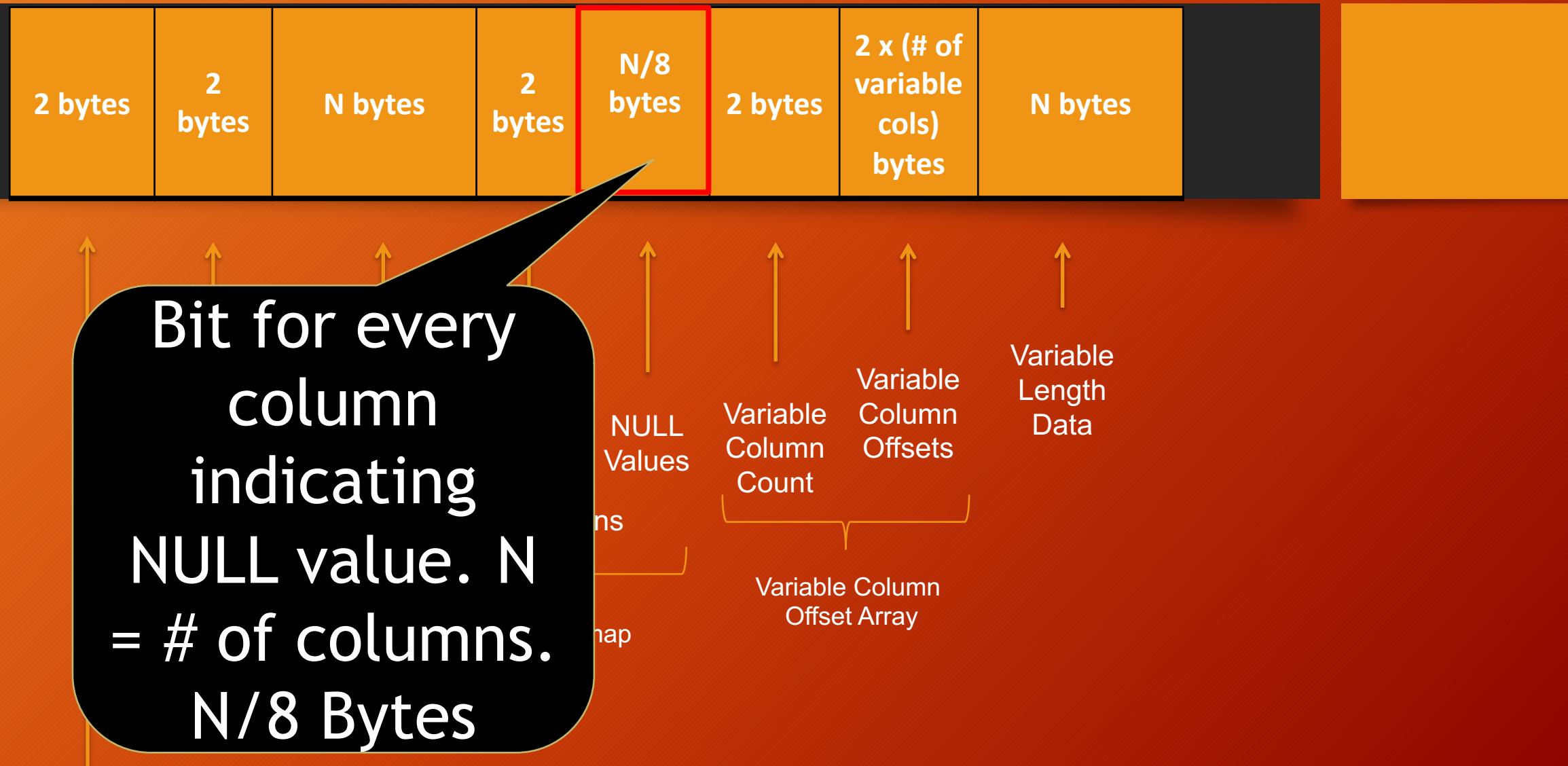




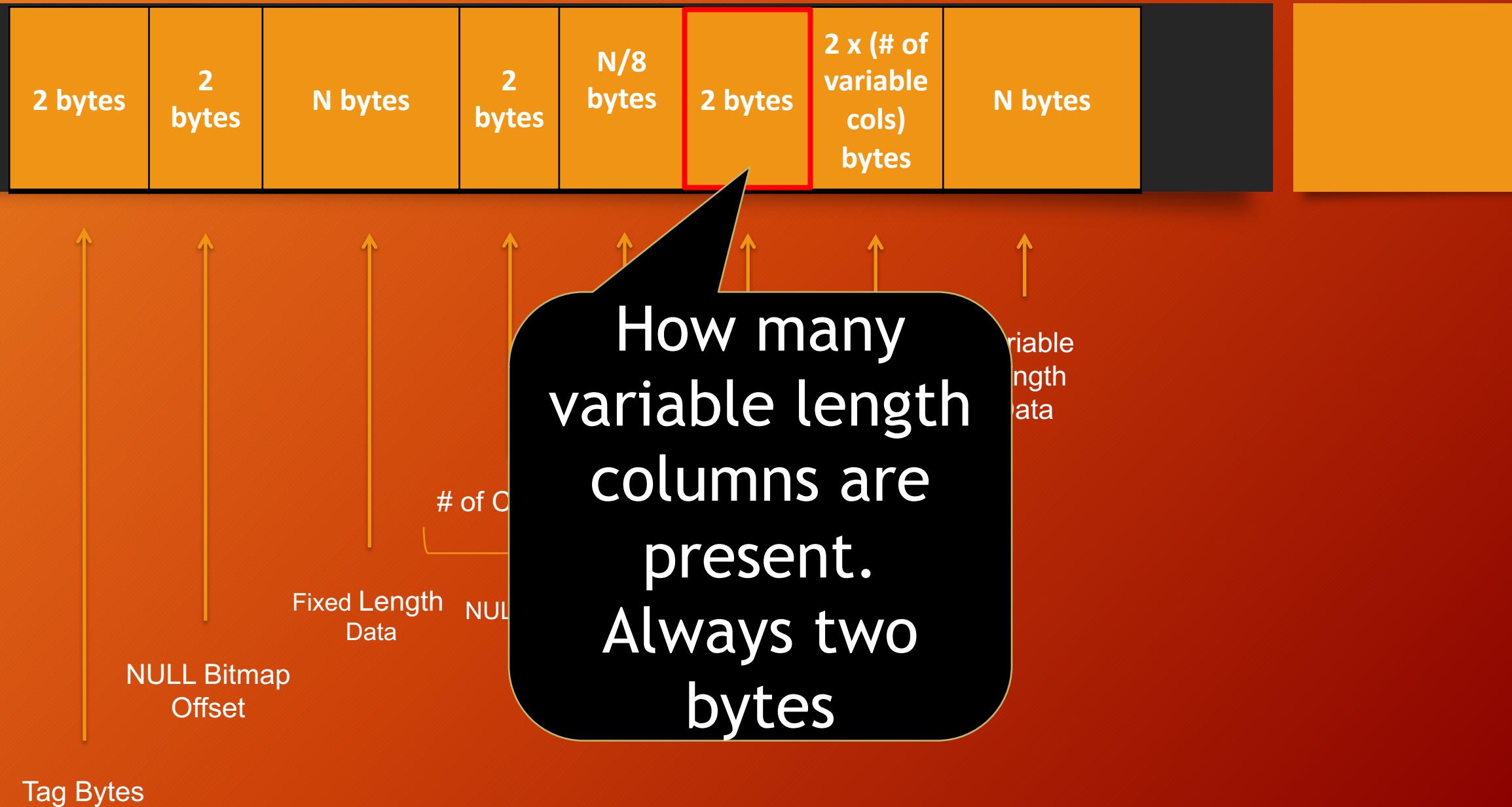


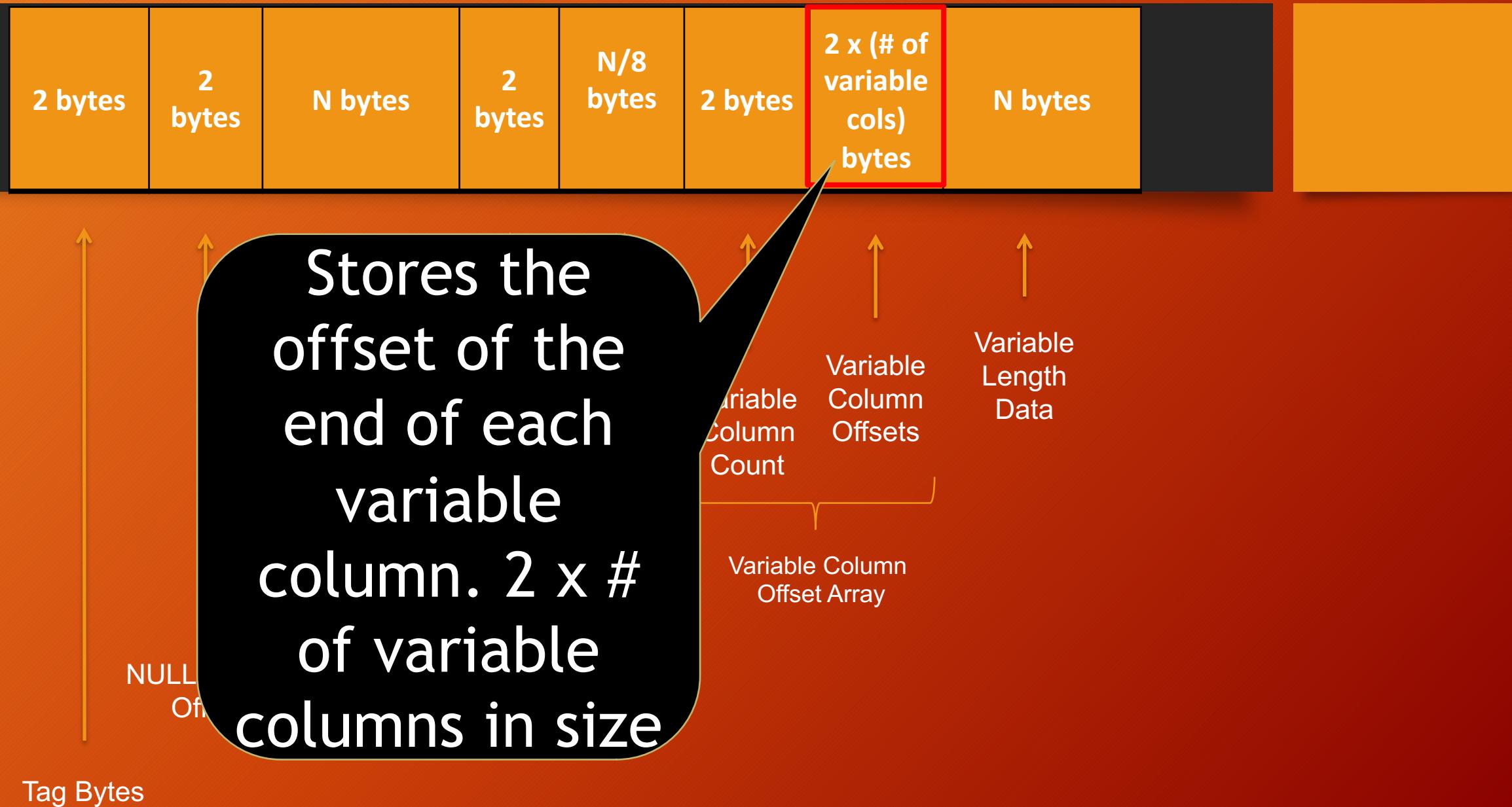






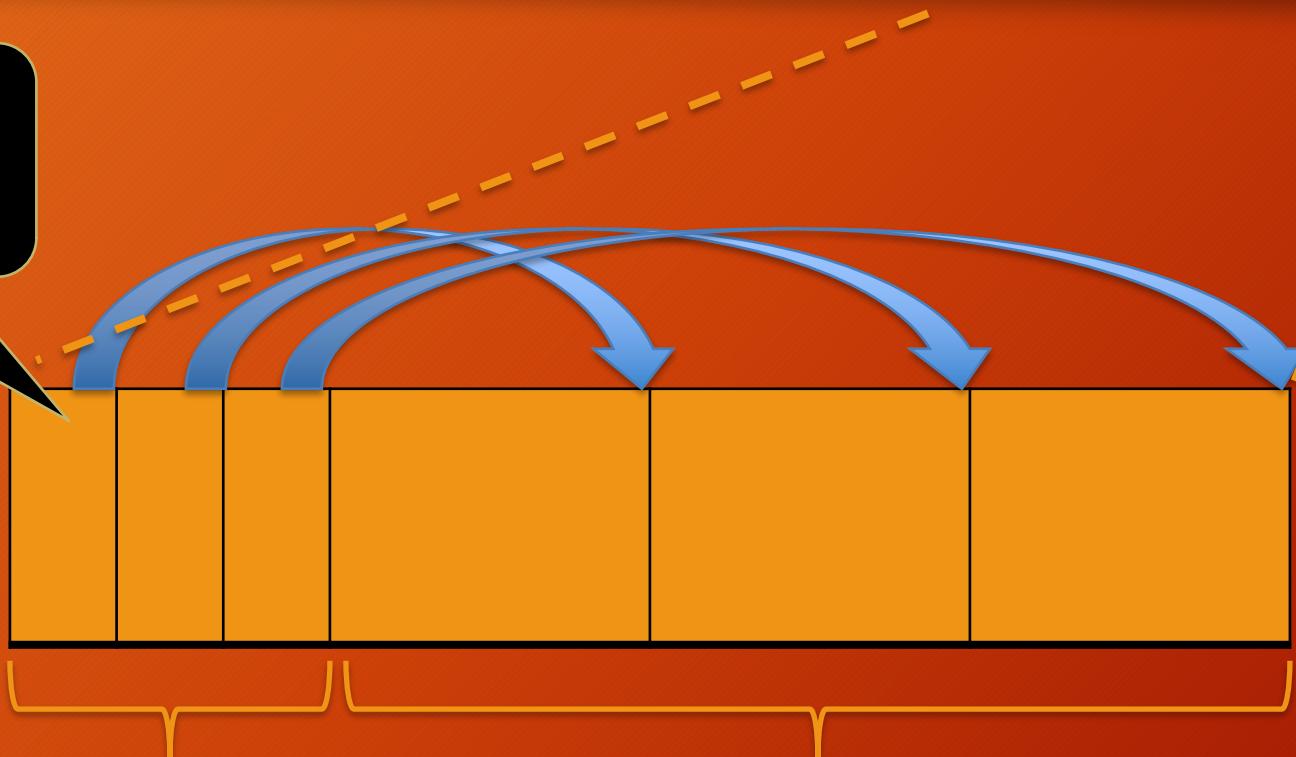
Tag Bytes







Stores END
offset



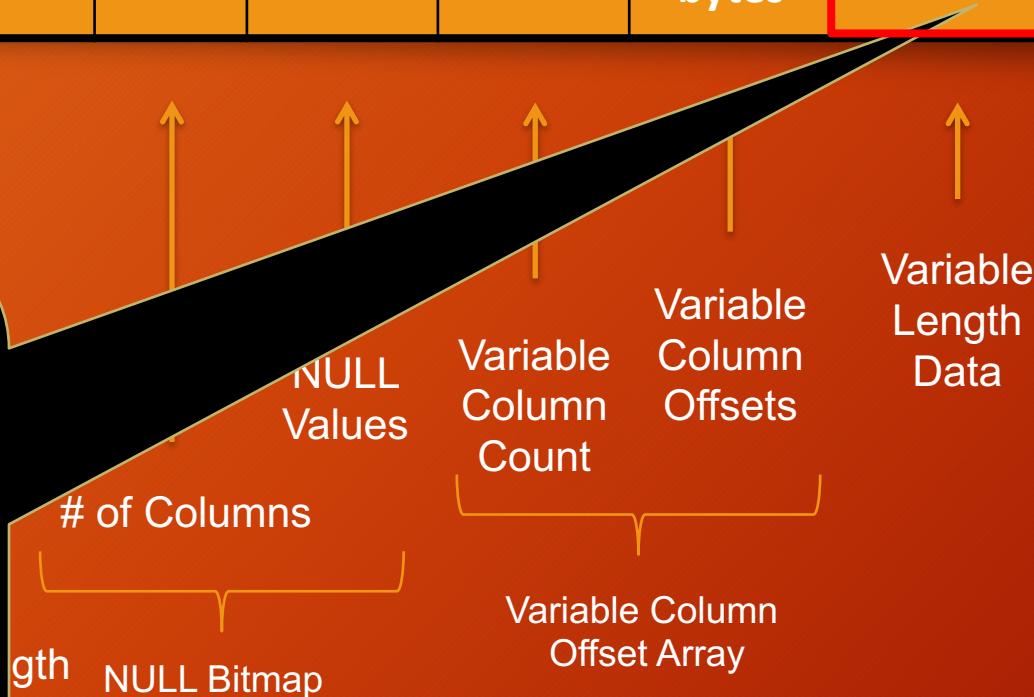
Variable Column
Offset Array

Variable Data



Variable
column data
lives here. N =
total sum of
variable data
length.

Tag Bytes



Records

```
CREATE TABLE Sharknado (
    CustomerID INT
    , CustomerName VARCHAR(50)
    , CustomerState CHAR(2)
) ;
```

How big can this row be?

Records

```
CREATE TABLE Sharknado(  
CustomerID INT  
, CustomerName  
    VARCHAR(50)  
, CustomerState CHAR(2)  
);
```

For the:	Bytes
Tag	?
NULL Offset Bitmap	?
Fixed Data Length	?
# of columns	?
Null Columns (N/8)	?
Variable Column Count	?
Variable Column Offset	?
Variable Length Columns	?
Total:	? bytes

Records

```
CREATE TABLE Sharknado(  
CustomerID INT  
, CustomerName  
    VARCHAR(50)  
, CustomerState CHAR(2)  
);
```

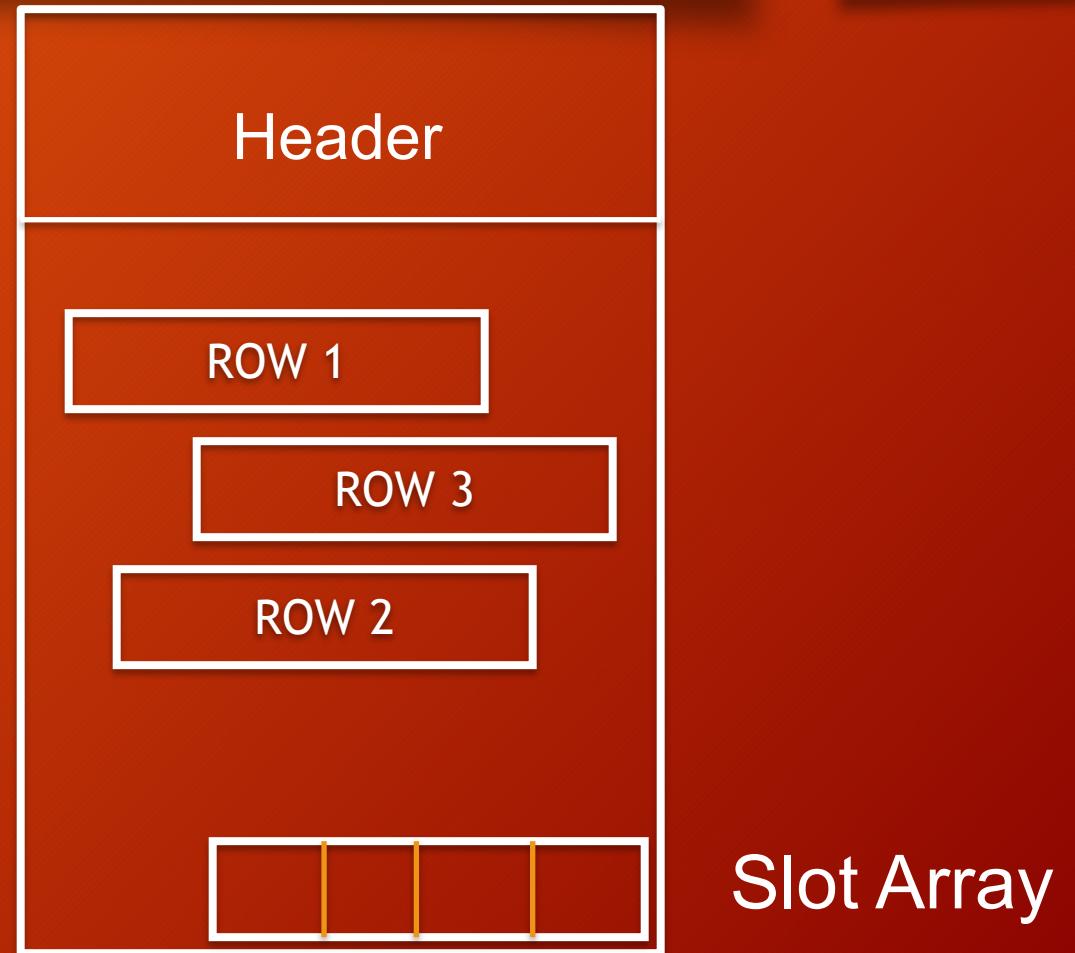
For the:	Bytes
Tag	2
NULL Offset Bitmap	2
Fixed Data Length	6
# of columns	2
Null Columns (N/8)	1
Variable Column Count	2
Variable Column Offset	2
Variable Length Columns	50
Total:	67 bytes

Overall
Structure

Records live in
Pages

Pages

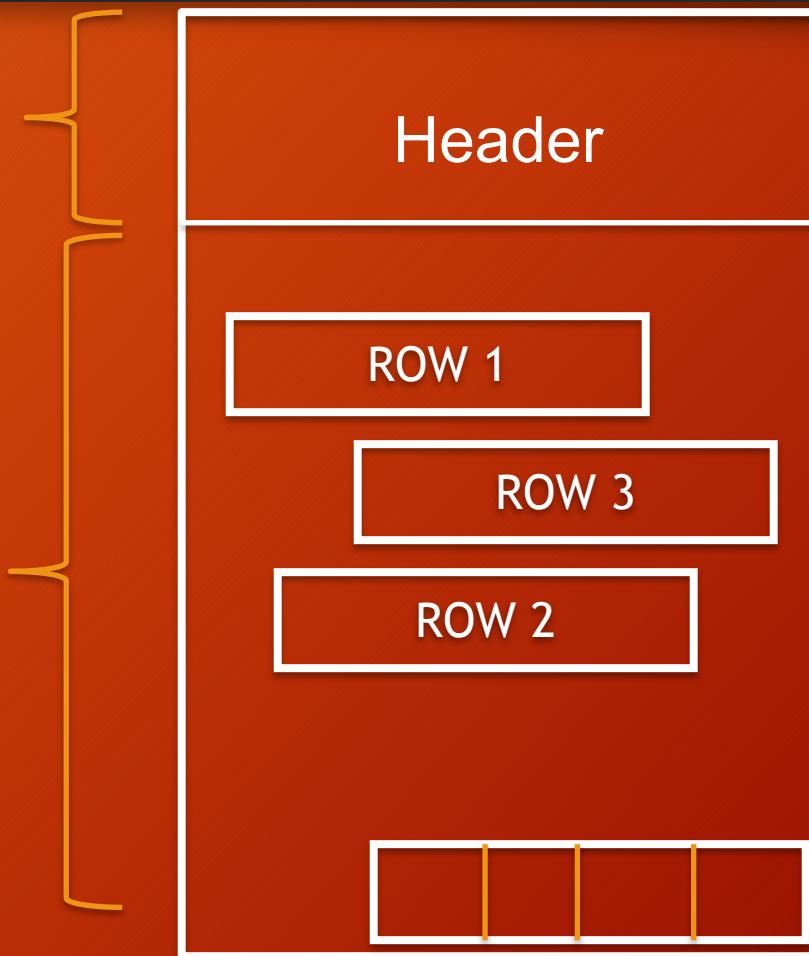
- All pages are 8192 bytes (8k) in size.
- Every page has a Header.
- Pages store the rows.
- Every page has a slot array.



Pages

96 Bytes for Header

8096 Bytes for Rows



Pages - Header

- Things to Note:
 - Page Type
 - Object ID
 - Ghost Count
 - Record
 - Slot Count

```
PAGE: (1:696)

BUFFER:

BUF @0x000000000BFB02C0

bpage = 0x000000000B2B2000          bhash = 0x0000000000000000          bpageno = (1:696)
bdbid = 6                           bpreferences = 0                   bcputicks = 376
bsampleCount = 1                   bUse1 = 4701                      bstat = 0xc00009
blog = 0x321159                   bnnext = 0x0000000000000000

PAGE HEADER:

Page @0x000000000B2B2000

m_pageId = (1:696)                  m_headerVersion = 1              m_type = 10
m_typeFlagBits = 0x0                 m_level = 0                     m_flagBits = 0x200
m_objId (AllocUnitId.idObj) = 147   m_indexId (AllocUnitId.idInd) = 256
Metadata: AllocUnitId = 72057594047561728
Metadata: PartitionId = 72057594045333504
Metadata: ObjectId = 1509580416      m_prevPage = (0:0)             Metadata: IndexId = 1
pmminlen = 90                        m_slotCnt = 2                   m_nextPage = (0:0)
m_freeData = 8182                    m_reservedCnt = 0               m_freeCnt = 6
m_xactReserved = 0                  m_xdesId = (0:0)              m_lsn = (41:3993:669)
m_tornBits = -627050546            m_ghostRecCnt = 0
```

PAGE: (1:696)

BUFFER:

BUF @0x000000000BFB02C0

bpage = 0x000000000B2B2000
bdbid = 6
bsampleCount = 1
blog = 0x32159

bhash = 0x0000000000000000
breferences = 0
bUse1 = 4701
bnext = 0x0000000000000000

bpageno = (1:696)
bcputicks = 376
bstat = 0xc00009

PAGE HEADER:

Page @0x000000000B2B2000

m_pageId = (1:696)
m_typeFlagBits = 0x0
m_objId (AllocUnitId.idObj) = 147
Metadata: AllocUnitId = 72057594047561728
Metadata: PartitionId = 72057594045333504
Metadata: ObjectId = 1509580416
pminlen = 90
m_freeData = 8182
m_xactReserved = 0
m_tornBits = -627050546

m_headerVersion = 1
m_level = 0
m_indexId (AllocUnitId.idInd) = 256
Metadata: IndexId = 1
m_prevPage = (0:0)
m_slotCnt = 2
m_reservedCnt = 0
m_xdesId = (0:0)

m_flagBits = 0x200
Metadata: IndexId = 1
m_nextPage = (0:0)
m_freeCnt = 6
m_lsn = (41:3993:669)
m_ghostRecCnt = 0

Pages

Single page size limit
for data is 8060 bytes

- No exceptions
- Can store single record or multiple records
- Off row storage

Slot Array

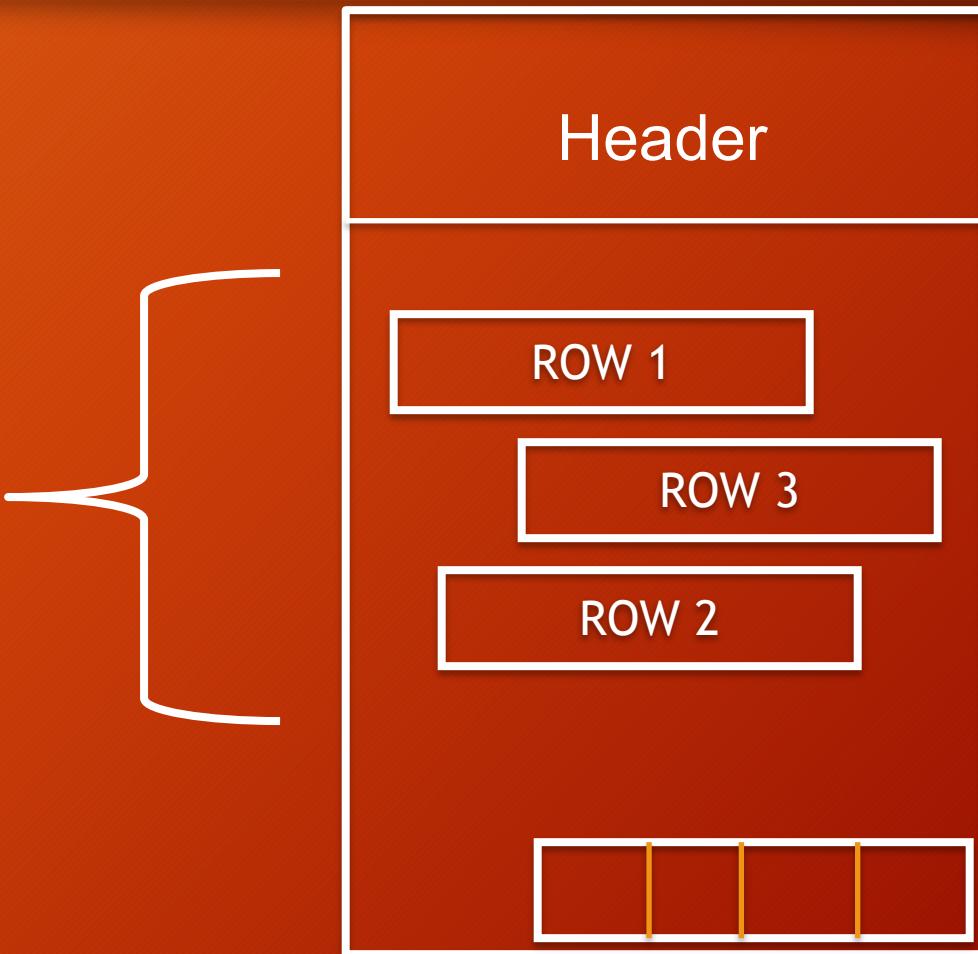
- Stores the offsets to each row on the page
- 2 bytes per row

The rows do not have to be stored in order physically on the page

- The slot array offsets will stored in sorted order

Pages

Not in order!!



Pages

Records are stored
on the same type
of pages

- IE: Data records are stored on Data pages
- IE: Index records are stored on Index pages

Boot page

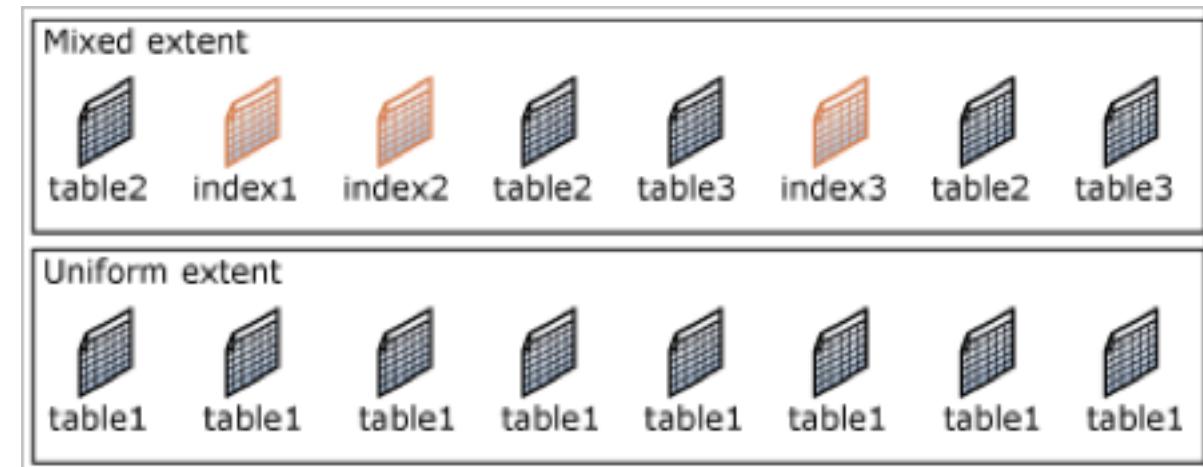
- Page (1:9) (file:page#)
- Store metadata about the database
 - Very critical page.
 - If corrupt, restore is the only option

Overall Structure

Groups of 8 pages
is an Extent

Extents

- Groups of 8 contiguous pages
 - Easier page allocation for SQL Server
- Mixed Extents
 - Single pages that are allocated to different objects
- Dedicated Extents
 - 8 pages that are allocated to the same object



- DBCC IND
- DBCC PAGE
- Sys.fn_PhysLocFormatter
- Sys.dm_db_database_page_allocations

Tools

- DBCC IND
- DBCC PAGE
- Sys.fn_PhysLocFormatter
- Sys.dm_db_database_page_allocations

Not Supported by Microsoft. There are a TON of folks online willing to help with issues and/or questions.

Tools

- Undocumented
 - Sort of
- Unsupported
 - Sort of

DBCC IND

```
DBCC IND(  
    <DatabaseName>  
    ,<'tablename'>  
    , <Index_ID>  
)
```

For example:

```
DBCC IND(AdventureWorks,' Person.Person',1)
```

DBCC PAGE

```
DBCC PAGE(  
    <database_name>  
    , <fileid>  
    , <pagenumber>  
    , <detail_level>  
)
```



- 0 = Header
- 1 = Header/hex dump
for rows
- 2 = Header/page dump
- 3 = Header/detail row
info

For example:

```
DBCC PAGE(AdventureWorks, 1, 696, 3)
```

DBCC PAGE

```
DBCC PAGE(  
    <database_name>  
    , <fileid>  
    , <pagenumber>  
    , <detail_level>  
)
```

Note: Must
execute DBCC
TRACEON(3604)
BEFORE DBCC
PAGE!

For example:

```
DBCC PAGE(AdventureWorks, 1, 696, 3)
```

SYS.DM_DB_DATABASE_PAGE_ALLOCATIONS

```
SELECT * FROM  
SYS.DM_DB_DATABASE_PAGE_ALLOCATIONS (  
    <database_id>  
    , <table_id>  
    , <index_id>  
    , <partition_id>  
    , <mode>)
```

For example:

```
SELECT * FROM  
SYS.DM_DB_DATABASE_PAGE_ALLOCATIONS(db_id(), object_id('dbo.table1'), NULL,  
NULL, 'DETAILED')
```

Demo

***STOP: 0x000000D1 (0x00000000, 0xF73120AE, 0xC0000008, 0xC0000000)

A problem has been detected and Windows has been shut down to prevent damage to your computer

DRIVER_IRQL_NOT_LESS_OR_EQUAL

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

Check to make sure any new hardware or software is properly installed. If this is a new installation, ask your hardware or software manufacturer for any windows updates you might need.

If problems continue, disable or remove any newly installed hardware or software. Disable BIOS memory options such as caching or shadowing. If you need to use Safe Mode to remove or disable components, restart your computer, press F8 to select Advanced Startup Options, and then select Safe Mode.

**** ABCD.SYS - Address F73120AE base at C0000000, DateStamp 36B072A3

Kernel1 Debugger Using: COM2 (Port 0x2F8, Baud Rate 19200)

Beginning dump of physical memory

Physical memory dump complete. Contact your system administrator or technical support group.

JUST
KIDDING!



How many will fit on a page?

```
CREATE TABLE Sharknado(  
CustomerID INT  
, CustomerName  
    VARCHAR(50)  
, CustomerState CHAR(2)  
);
```

For the:	Bytes
Tag	2
NULL Offset Bitmap	2
Fixed Data Length	4+2 = 6
# of columns	2
Null Columns (N/8)	1
Variable Column Count	2
Variable Column Offset	2
Variable Length Columns	50
Total:	67 bytes

Records

```
CREATE TABLE Sharknado(  
    CustomerID INT  
    , CustomerName VARCHAR(50)  
    , CustomerState CHAR(2)  
);
```

8060 / 69 bytes = 116 rows per page

Bring it all together

Row Density

How much data
do we have in
the row?

Page Density

How many rows
do we have on
the page?

- Understanding the internals is critical for everything.
- Smart table design will lead to:
 - More rows on a page
 - Less Pages to store the data
 - Less Work for SQL Server to read the pages
 - Faster response times

Summary

- MCM Readiness Video
 - <http://technet.microsoft.com/en-US/sqlserver/gg313756.aspx>
- DBCC IND/PAGE
 - Paul Randal -
<http://www.sqlskills.com/blogs/paul/category/dbcc/>
 - http://blogs.msdn.com/b/sqlserverstorageengine/archive/2006/12/13/more-undocumented-fun_3a00_-dbcc-ind_2c00_-dbcc-page_2c00_-and-off_2d00_row-columns.aspx
- Data Types
 - <http://msdn.microsoft.com/en-us/library/ms187752.aspx>
- SGAM, GAM, IAM, PFS, DIFFMAP
 - <http://blogs.msdn.com/b/sqlserverstorageengine/archive/2006/07/08/under-the-covers-gam-sgam-and-pfs-pages.aspx>
- Anatomy of a Record (Paul Randal)
 - <http://www.sqlskills.com/blogs/paul/inside-the-storage-engine-anatomy-of-a-record/>

Resources

Thank a Volunteer

Take a moment
on your way out
to thank one of
the many
volunteers that
put this
conference
together.



THANK YOU!!!!



john@dcac.com



@SqlRUs



<http://linkedin.com/in/sqlrus>



<http://www.sqlrus.com>