

# Why Your Data Type Choices Matter

**Andy Yun**  
Senior Solutions Engineer



# Andy Yun

## *Senior Solutions Engineer*

- SQL Server DBA & DB Developer
- Chicago Suburban User Group Chapter Leader
- Chicago SQL Association – Director-at-Large
- Working with SQL Server since 2001
- Speaking since Early 2014
- Microsoft MVP (2017-2018)



sqlbek@gmail.com | ayun@sentryone.com  
<https://blogs.sentryone.com/andyyun/>  
@SQLBek

# Data Types: Who Chose What?

- FirstName NCHAR(255)
- Address  
NVARCHAR(MAX)
- OrderVolume BIGINT
- ZipCode INT
- DateCreated CHAR(10)
- DailyRevenue FLOAT(53)

Why not just use NVARCHAR(255), NVARCHAR(MAX), BIGINT, & DECIMAL(28,14) everywhere?

# Presentation Goals

**Answer:** How does SQL Server really store data records?

An  
of

*This may be a **300 level** session,  
but you're going to get a  
helping of **400 & 500 level** content too!*

ences

**Takeaway:** Assess your own systems & implement some best practices

# Internals: How Data is Stored

# Employee Table

```
CREATE TABLE dbo.Employee_Large (  
    EmployeeID          UNIQUEIDENTIFIER PRIMARY KEY CLUSTERED,  
    FirstName           NCHAR(255),  
    LastName            NCHAR(255),  
    Gender              INT,  
    SocialSecurityNumber NCHAR(50),  
    Title               NVARCHAR(255),  
    ManagerID           UNIQUEIDENTIFIER,  
    DateHired           DATETIME,  
    Salary              FLOAT,  
    EmailAddress         NVARCHAR(255),  
    HomeOfficeLocationCode NVARCHAR(4000),  
    VacationDaysRemaining INT,  
    RecordCreatedDate   DATETIME,  
    RecordCreatedBy     UNIQUEIDENTIFIER,  
    RecordLastModifiedDate DATETIME,  
    RecordLastModifiedBy UNIQUEIDENTIFIER  
)
```

# Fixed & Variable Length Data Types

Fixed Width Data Types	
TINYINT	SMALLINT
INT	BIGINT
DATETIME	SMALLDATETIME
CHAR	NCHAR
DECIMAL	FLOAT
UNIQUEIDENTIFIER	

Variable Width Data Types	
VARCHAR	NVARCHAR

# Fixed & Variable Length Data Types

Fixed Width Data Types & Storage Requirements			
TINYINT	1 byte	SMALLINT	2 bytes
INT	4 bytes	BIGINT	8 bytes
DATETIME	8 bytes	SMALLDATETIME	4 bytes
CHAR	n bytes	NCHAR	(n * 2) bytes
DECIMAL	5-17 bytes	FLOAT	4-8 bytes
UNIQUEIDENTIFIER	16 bytes		

Variable Width Data Types & Storage Requirements			
VARCHAR	n + 2 bytes	NVARCHAR	(n * 2) + 2 bytes



# FIXVAR Format

Tag A	Tag B	Fsize	Fdata	Ncol	Nullbits	VarCount	VarOffset	VarData
1 byte	1 byte	2 bytes	Fsize - 4	2 bytes	Ceiling (Ncol / 8)	2 bytes	2 x VarCount	n bytes

Tag A & B	Status Bits A & B
Fsize	Total Fixed-Length Size
Fdata	Actual Fixed-Length Data
Ncol	Total Number of Columns
Nullbits	NULL Bitmap (1 bit per column in table)
VarCount	Total Number of Variable Length Columns
VarOffset	Variable Column Offset Array
VarData	Actual Variable-Length Data

# Fixed Width Columns

Employee ID	UNIQUEIDENTIFIER
Gender	INT
Social Security Number	NCHAR(50)
Manager ID	UNIQUEIDENTIFIER
Date Hired	DATETIME

F-Data Size	Fixed Data				
144 bytes	Employee ID 16 bytes	Gender 4 bytes	SSN 100 bytes	Manager ID 16 bytes	Date Hired 8 bytes



# Variable Width Columns

Title	NVARCHAR(255)
Email Address	NVARCHAR(255)

VarCount	VarOffset *		Variable Width Data	
2	0x1C	0x48	Database Administrator	james@countingbytes.com
2 bytes	2 bytes	2 bytes	44 bytes	46 bytes

VarCount	VarOffset *		Variable Width Data	
2	0x1C	0x2E	Developer	isabella@countingbytes.com
2 bytes	2 bytes	2 bytes	18 bytes	52 bytes

\* = Hex Address Values

# Employee Large

```
CREATE TABLE dbo.Employee_Large (  
    EmployeeID          UNIQUEIDENTIFIER PRIMARY KEY CLUSTERED,  
    FirstName           NCHAR(255),  
    LastName            NCHAR(255),  
    Gender              INT,  
    SocialSecurityNumber NCHAR(50),  
    Title               NVARCHAR(255),  
    ManagerID           UNIQUEIDENTIFIER,  
    DateHired           DATETIME,  
    Salary              FLOAT,  
    EmailAddress         NVARCHAR(255),  
    HomeOfficeLocationCode NVARCHAR(4000),  
    VacationDaysRemaining INT,  
    RecordCreatedDate    DATETIME,  
    RecordCreatedBy      UNIQUEIDENTIFIER,  
    RecordLastModifiedDate DATETIME,  
    RecordLastModifiedBy UNIQUEIDENTIFIER  
)
```

Min. Size = 1304 bytes

# Right Sizing Data Types

Data Type	Storage	Value Range
TINYINT	1 Byte	0 to 255
SMALLINT	2 Bytes	-32,768 to 32,767
INT	4 Bytes	-2,147,483,648 to 2,147,483,647
BIGINT	8 Bytes	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
DATE	3 Bytes	0001-01-01 to 9999-12-31
SMALLDATETIME	4 Bytes	1900-01-01 00:00:00 to 2079-06-06 23:59:59
DATETIME	8 Bytes	1753-01-01 00:00:00.000 to 9999-12-31 23:59:59.997

# Right Sizing: Examples

SocialSecurityNumber

NCHAR(50)

- 1. Date?
- 2. Number?
- 3. String?



- 1. CHAR
- 2. NCHAR
- 3. VARCHAR
- 4. NVARCHAR



CHAR(9)

VacationDaysRemaining

INT

- 1. Date?
- 2. Number?
- 3. String?



- 1. TINYINT
- 2. SMALLINT
- 3. DECIMAL



DECIMAL(4, 2)

# Employee Narrow

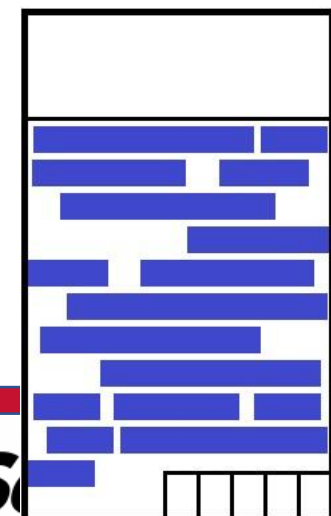
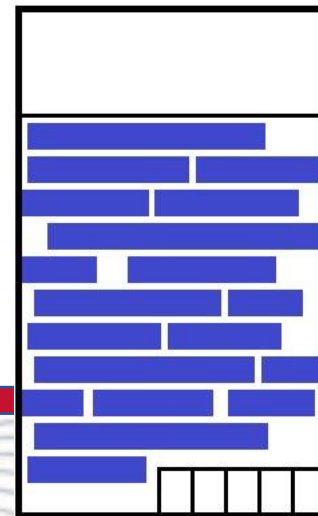
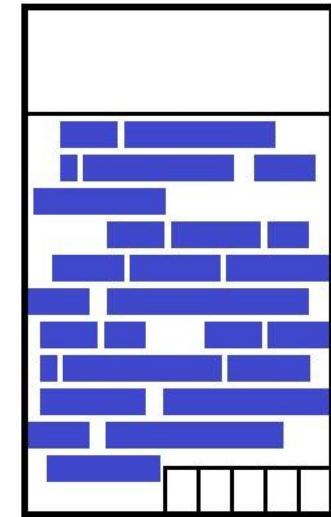
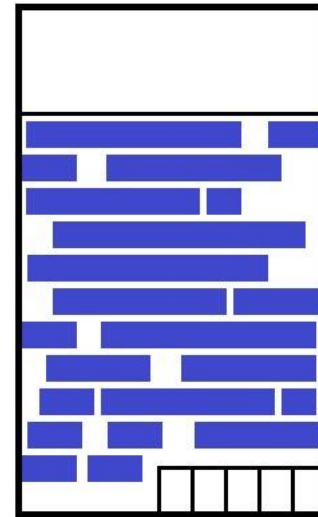
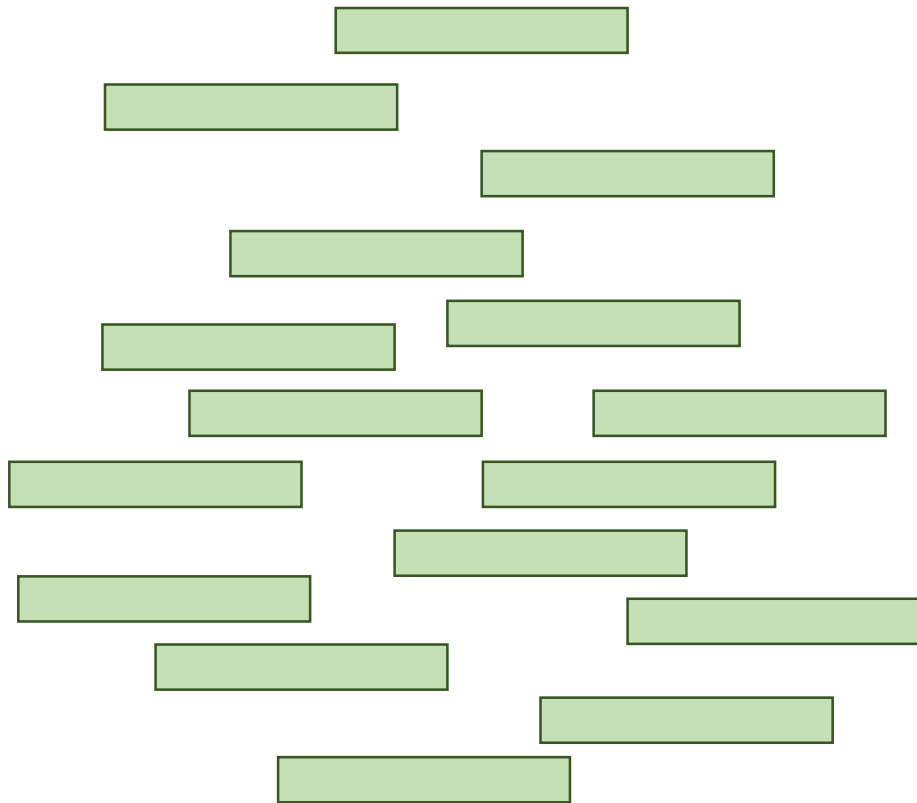
```
CREATE TABLE dbo.Employee_Narrow (  
    EmployeeID          INT IDENTITY(1, 1) PRIMARY KEY CLUSTERED,  
    FirstName           VARCHAR(100),  
    LastName            VARCHAR(100),  
    Gender              CHAR(1),  
    SocialSecurityNumber CHAR(9),  
    Title               VARCHAR(40),  
    ManagerID           INT,  
    DateHired           DATE,  
    Salary              DECIMAL(10, 2),  
    EmailAddress         VARCHAR(100),  
    HomeOfficeLocationCode TINYINT,  
    VacationDaysRemaining DECIMAL(4, 2),  
    RecordCreatedDate    SMALLDATETIME,  
    RecordCreatedBy      INT,  
    RecordLastModifiedDate SMALLDATETIME,  
    RecordLastModifiedBy INT  
)
```

Min. Size = 104 bytes

Employee\_Large:

Min. Size = 1304 bytes

# How are records stored on disk?





# Data Page: Overview

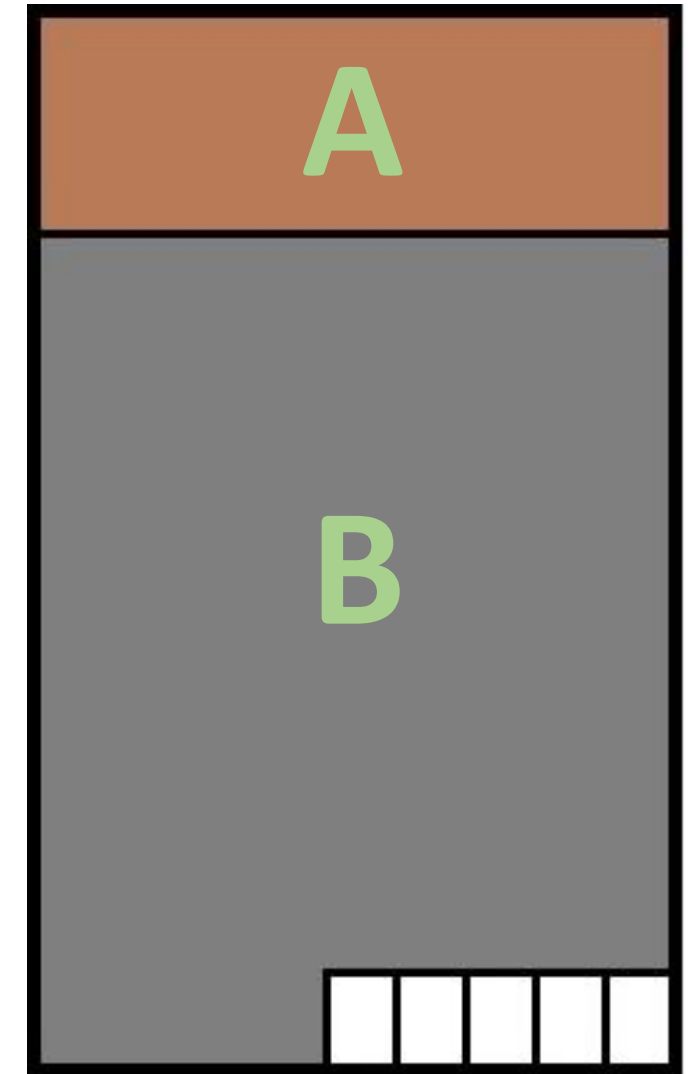
3 Types:

- In-Row Data
- Row-Overflow Data
- LOB Data

All Pages have a Fixed Size of 8KB or 8,192 bytes

Comprised of 3 components:

- A. Page Header: 96 bytes
- B. Data Records
- C. Record or Slot Array: 2 bytes per record



# The 8KB Limit

- Functionally limited in what can fit in a data page
- Because I/O executed via data PAGES, not data RECORDS, in best interest to maximize # of records per data page
- Typical data is stored “In-Row”
- VARCHAR(8000) & NVARCHAR(4000) use Row-Overflow Data Pages
- VARCHAR(MAX) & NVARCHAR(MAX) use LOB Data Pages

# MAX Types & LOB Columns

Col_One	VARCHAR(8000)
Col_Two	VARCHAR(MAX)

	Col_One	Col_Two
Value	Short String	Second Short String
Data Size	12 bytes	19 bytes

```
UPDATE tbl  
SET
```

```
    Col_One = 'Long ... string',  
    Col_Two = 'Long ... string';
```

Row Overflow  
Data Page

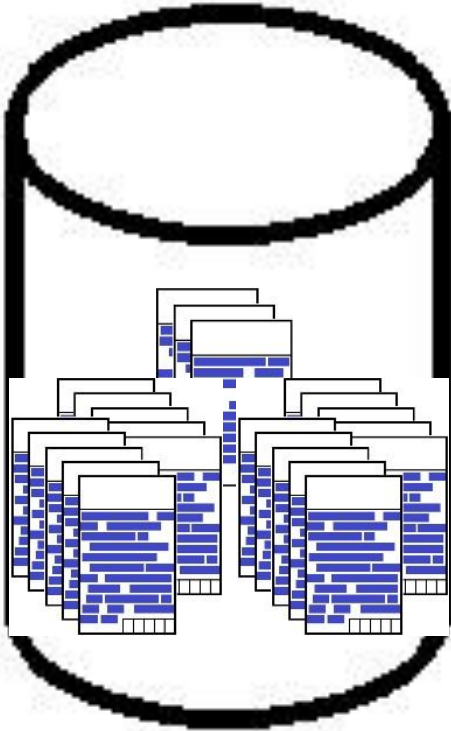
LOB  
Data Page

# How Many Per Page?

Employee Large		Records Per Page	Wasted Bytes
Min Record Size	1304 Bytes	6	272 bytes

Employee Narrow		Records Per Page	Wasted Bytes
Min Record Size	104 Bytes	77	88 bytes

# Why do we care?



```
SELECT
    FirstName, LastName
FROM Employee_Narrow
WHERE LastName LIKE 'S'
```

```
SELECT
    FirstName, LastName
FROM Employee_Large
WHERE LastName LIKE 'S'
```

# What's the Difference?

	Employee Large		Employee Small	
Number of Records	Pages	Size	Pages	Size
1	1	8 KB	1	8 KB
30,000	5,000	40 MB	390	3.12 MB
1,500,000	250,000	2,000 MB	19,481	155.48 MB
25,000,000	4,166,667	33,333 MB	324,676	2,597 MB

# Demo: Exploring Records & Pages

# Transaction Log Overhead



# Transaction Log Overhead

Does record size impact how much

*Be mindful of ALL I/O!*

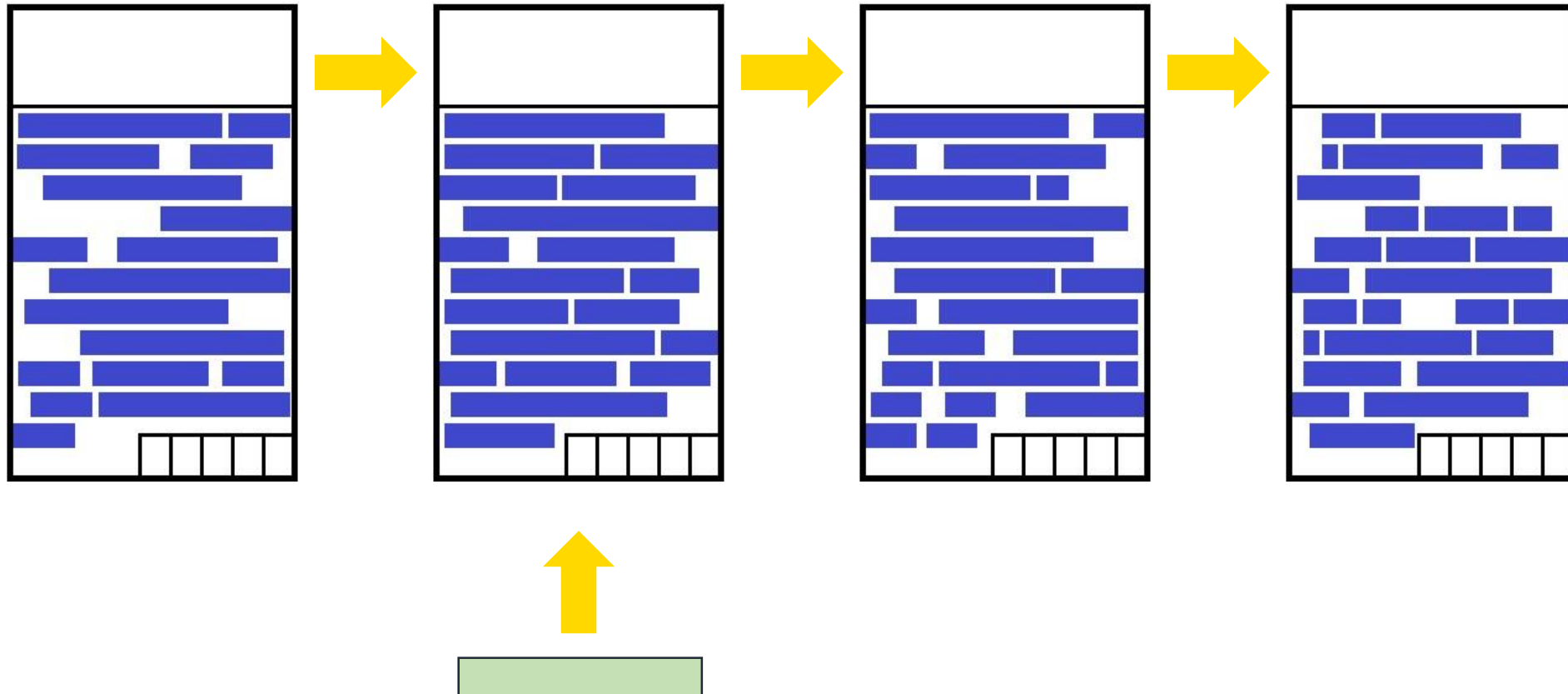
*More I/O = more work  
for SQL Server*

- INSERT, UPDATE, DELETE, MERGE, and other DML operations are writing to Transaction Log
- Amount of I/O is proportional to amount of work done by SQL Server
- T-Log volume impacts Backup/Restore, Replication, Log Shipping, Mirroring & Availability Groups

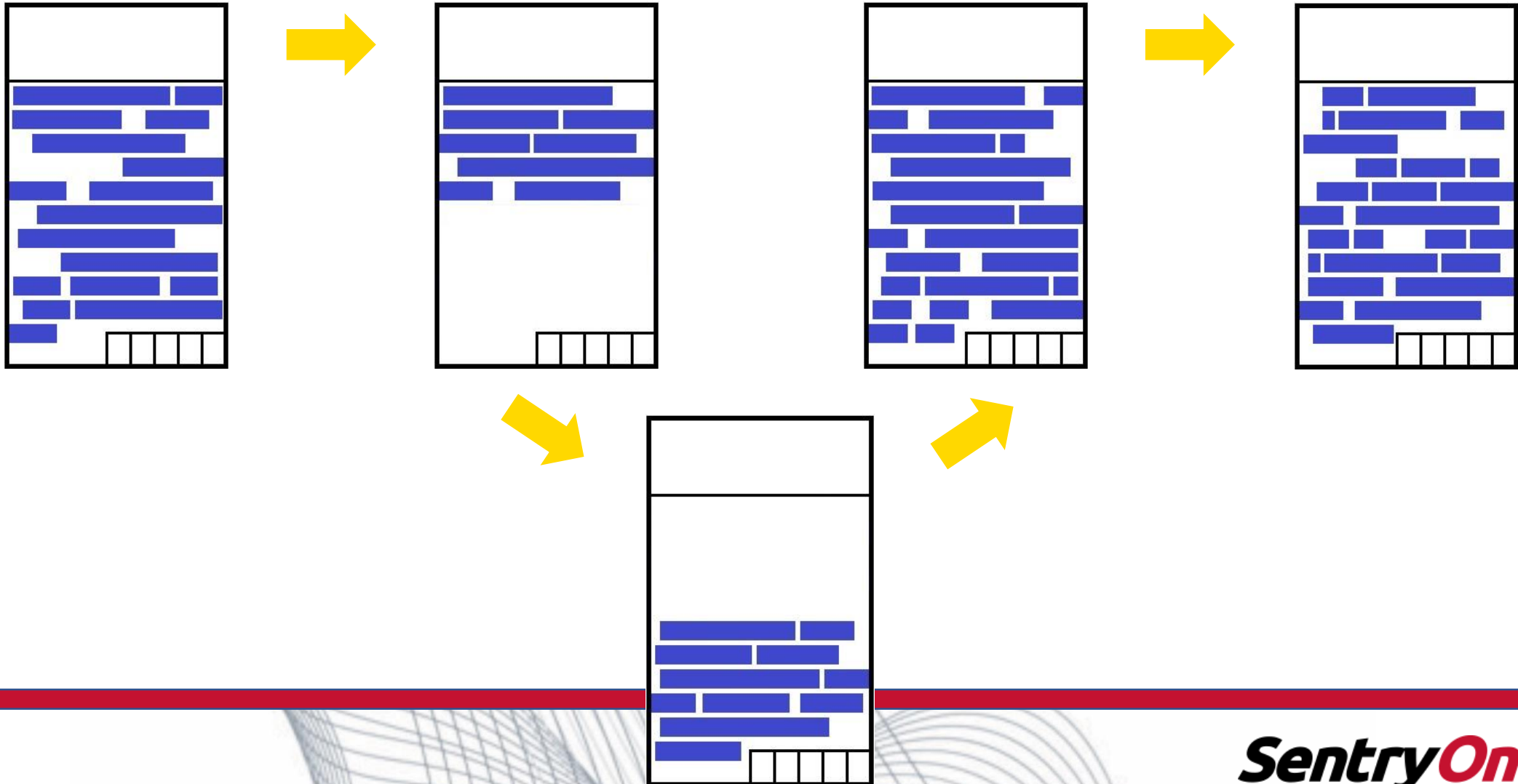
# Demo: Transaction Log Overhead

# Page Splitting

# Can I Squeeze In?



# We Need a Bigger Boat



# Page Splitting: Heaps & UPDATES

- Page Splits only occur on Clustered Tables & only on INSERT operations.
- UPDATE a Variable Length Column?
- If Record cannot be updated in-place or elsewhere on the same page, the UPDATE becomes a DELETE & INSERT!

# Demo: Page Splitting

# Thoughts on GUID Clustering Keys

General guidelines for clustering key:

Narrow	Static
Unique	Ever-Increasing

What about using NEWSEQUENTIALID()

Segregate Clustering & Primary Key = Surrogate & Business Keys  
IDENTITY Clustering Key & GUID Non-Clustered Primary Key

*Clustering Key Series By: Kimberly Tripp Randal*

<http://www.sqlskills.com/blogs/kimberly/category/clustering-key/>



# Implicit Conversions

# Alike or Not Alike?

@VariableOne	@VariableTwo
5	5.000
12345.67	'12,345.67'
2014-03-23	2014-03-23 00:00:00
'This is a String'	'This is a String'

	@VariableOne	@VariableTwo	
INT	5	5.000	DECIMAL(6, 3)
DECIMAL(6, 2)	12345.67	'12,345.67'	VARCHAR(9)
DATE	2014-03-23	2014-03-23 00:00:00	SMALLDATETIME
VARCHAR(25)	'This is a String'	'This is a String'	NVARCHAR(25)

# Converting w. Data Type Precedence

Value Conversions follow  
preset Precedence rules

Smaller Data Type always up-  
converted to larger Data Type

## Condensed Precedence Chart

- 
1. DATETIME
  2. SMALLDATETIME
  3. DATE
  4. DECIMAL
  5. BIGINT
  6. INT
  7. SMALLINT
  8. NVARCHAR
  9. NCHAR
  10. VARCHAR
  11. CHAR

# Demo: Implicit Conversions

# Homework

## Finding Implicit Column Conversions in the Plan Cache

By Jonathan Kehayias

<http://www.sqlskills.com/blogs/jonathan/finding-implicit-column-conversions-in-the-plan-cache/>

## Mining Plans : Not Just For the Plan Cache

By Erin Stellato

<https://sqlperformance.com/2018/08/sql-plan/mining-plans-not-just-plan-cache>

# To UNICODE Or Not To UNICODE?

## Negative Trade-offs?

### USE VARCHAR?

Data could be lost  
May need to refactor

### USE NVARCHAR?

Storage & I/O Requirement x2



**Andy Yun**  
**@SQLBek**

2016-07-21 14:52

OH "ALWAYS use nvarchar/nchar unless you are 100% CERTAIN that the field will NEVER require any non-western European characters"

NOOOO!!!!

# Why Not NVARCHAR(500)?

- Execution Plan Costs, whose variables includes **Estimated Row Size**
- What is the estimated row size of a variable width datatype?
  - One Half: VARCHAR/NVARCHAR( $n / 2$ )
- Costs get bloated & memory grants are oversized

# Parting Thoughts

1. Right-size your data types
2. Remember the 8KB data page size limit
3. Match your data types
4. Be mindful of Internals related performance impacts



# Thank you!

<https://github.com/SQLBek>

<http://www.speakingmentors.com/>

Andy Yun | @SQLBek  
ayun@sentryone.com | SQLBek@gmail.com

<http://blogs.sentryone.com/andyyun/>  
<http://sqlbek.wordpress.com>

Want To Learn More About SentryOne?  
Book a demo with me!

<http://www.sentryone.com/BookAndy/>