SQL Server Indexes and Storage Deep Dive

Tomáš Jecha

Microsoft Most Valuable Professional (MVP), MCSD

.NET team lead, software architect, lector

tomas@jecha.net


How data are queried?
Good Database Design
How data are stored?

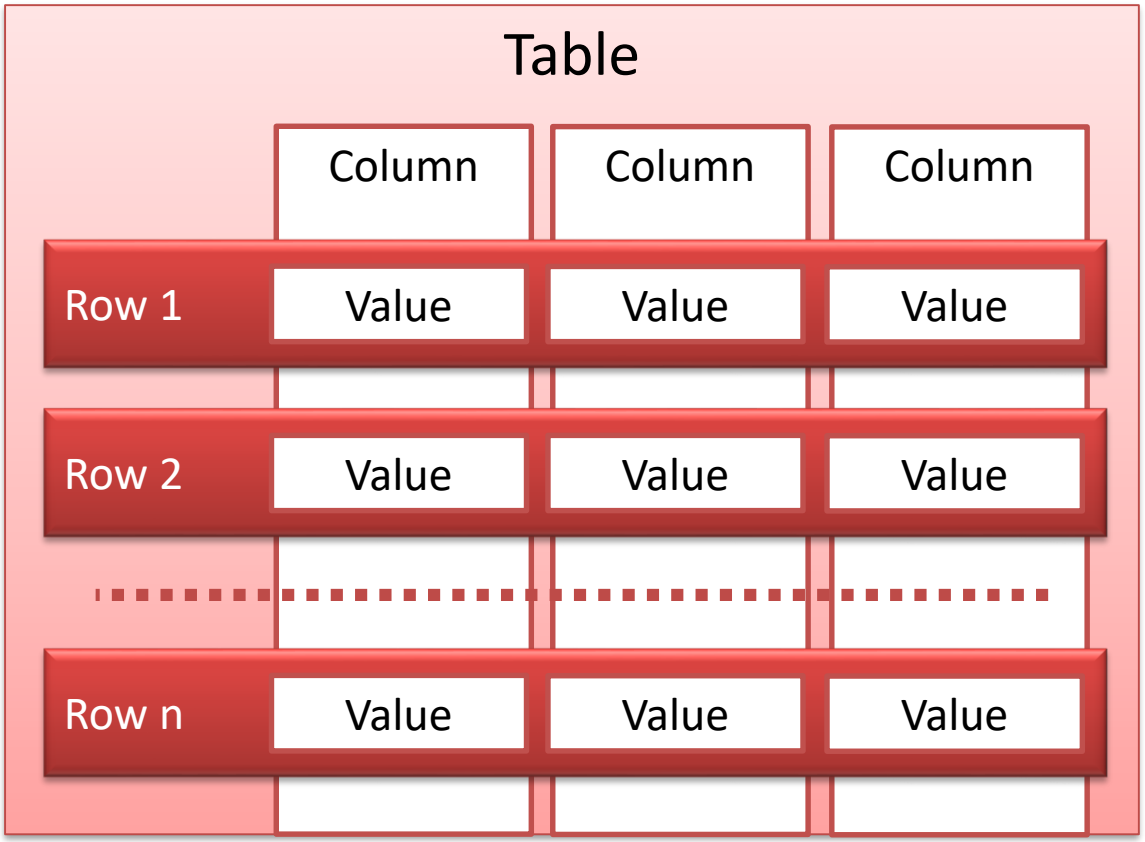
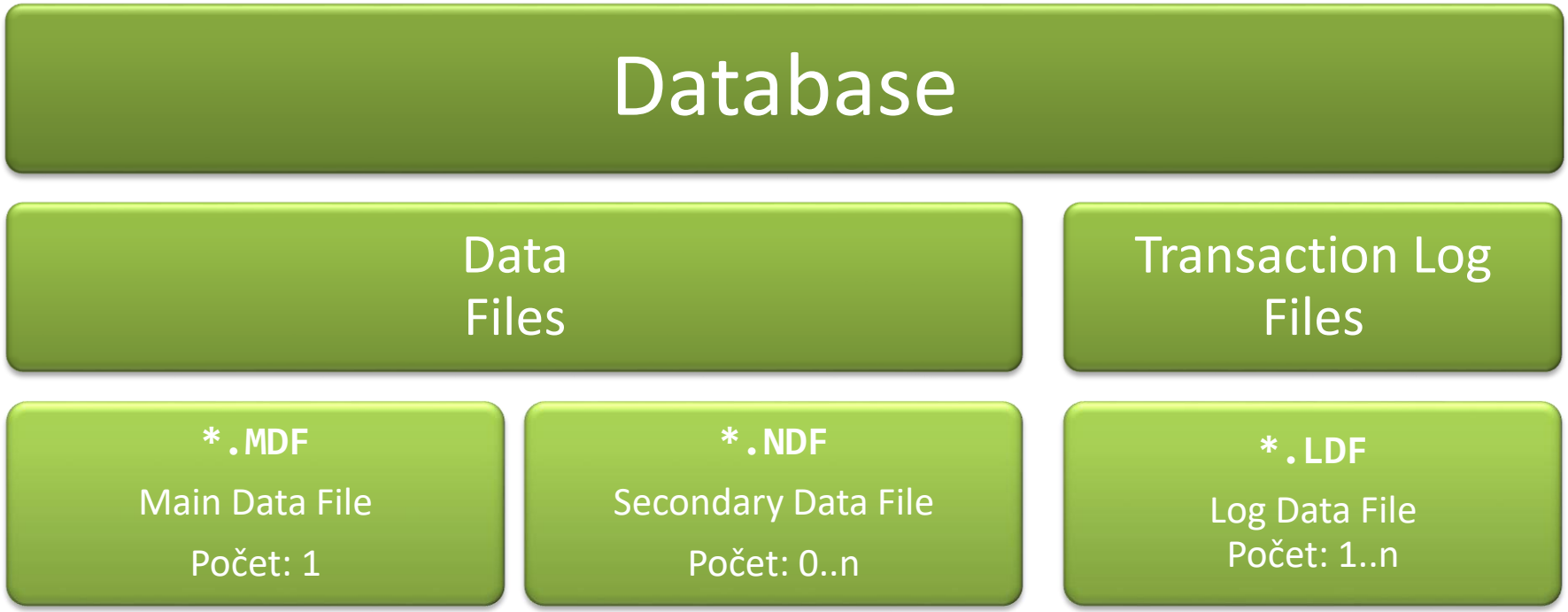


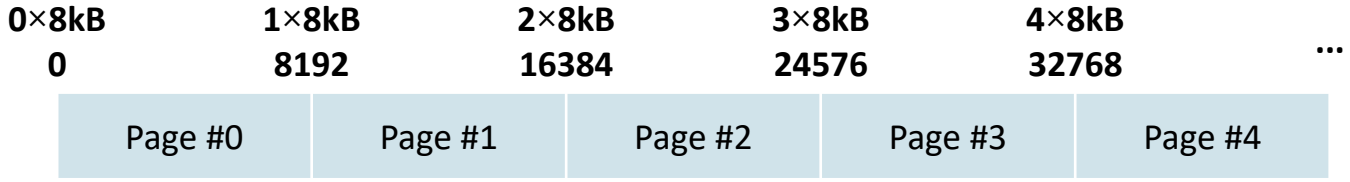
Table
Column
Column
Column
Row 1
Value
Value
Value
Row 2
Value
Value
Value
Row n
Value
Value
Value

Database Files



Structure of data file - Pages

- Splitted to 8kB pages
  (1MB = 128 pages)



- Page can be allocated or free
- Reference format: `{1:0} {1:356} {3:30}`
- Different types of pages:

  Data Page, Index Page, File Header Page…

Structure of data file - Extends

- Group of 8 pages (64kB)
  - `Extend_Index = Page_Index % 8`

Extend #0								Extend #1							
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Extend #2								Extend #3							
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Extend #4								...							
32	33	34	35	36	37	38	39								

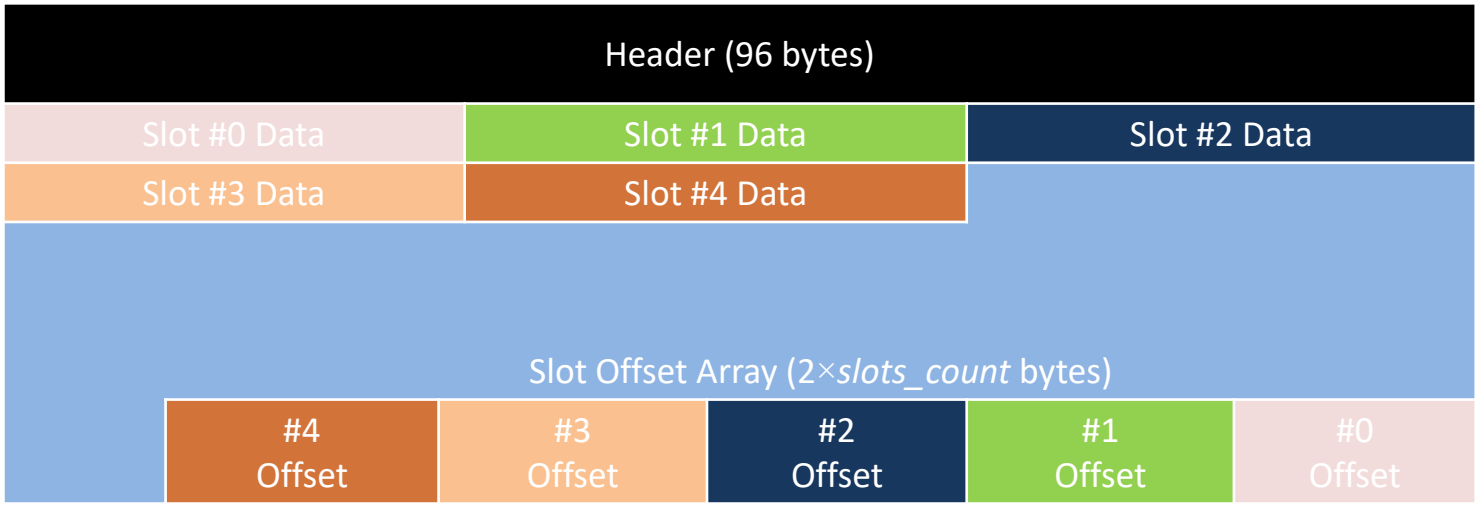
- Block reading and writing is faster
- Allocation of larger blocks saves space
- Uniform/mixed extend – uniform extend is owned by single allocation unit

Most common page types

- System or allocation pages
  - Header
  - Allocation units scope: IAM
  - Global data file scope: PFS, GAM, SGAM
- Data
  - Index and data
  - Row overflow, LOB

Common page structure

- Header – 96 bytes
- Data – 8096 bytes
  - Slot Offset Array – end of data section



DBCC PAGE

demo

Global Allocation Map (GAM)
Shared Global Allocation Map (SGAM)

- Main type of allocation
- 2 pages (GAM then SGAM) each 4GB
  - 64000 extends
  - Bitmap (1bit = 1 extend)
- GAM – least one page from extend used?
- SGAM – extend mixed and have at least one unallocated page?

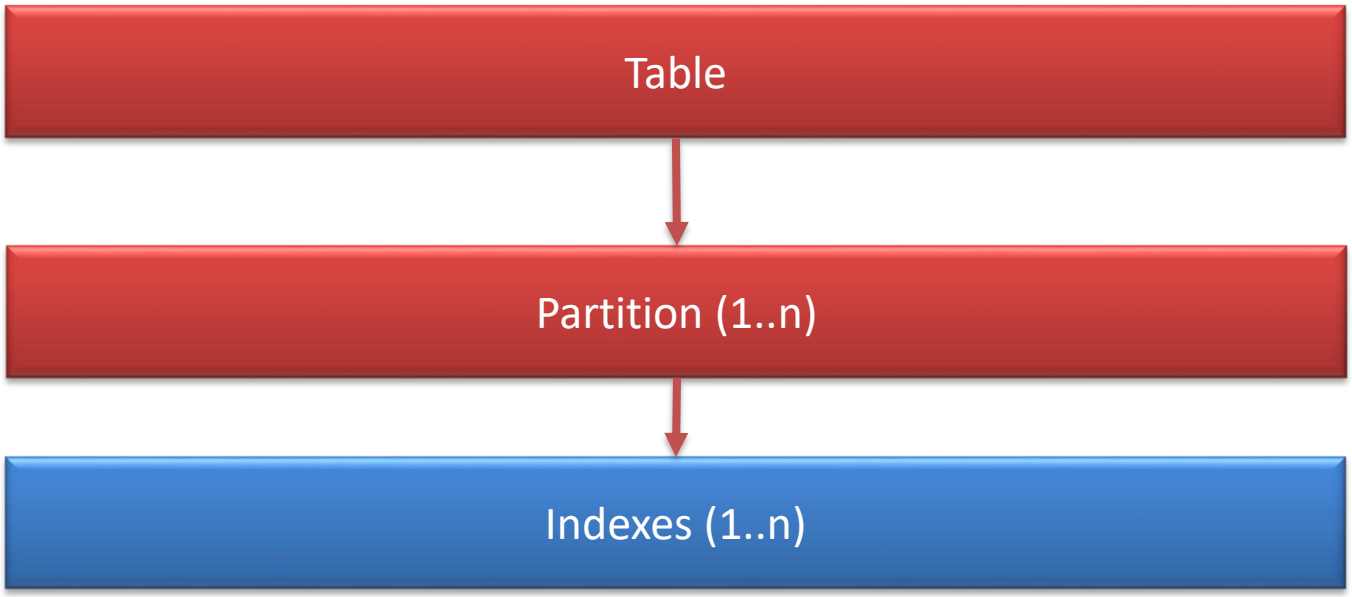
Index Allocation Map (IAM)

- List of pages & extends owned by allocation unit
- One per allocation unit per 4GB interval
- Stores:
  - Reference to first 8 pages
  - And map of 511232 pages in current 4GB interval (1bit = 1extend)

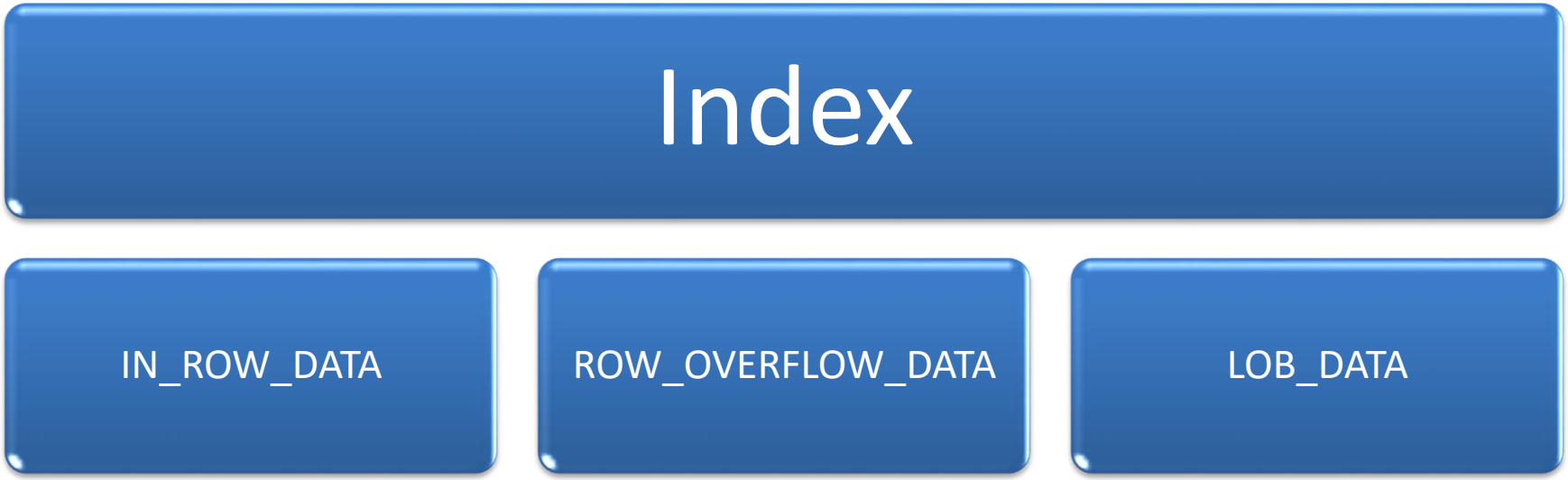
Allocation Maps

demo

Table Structure



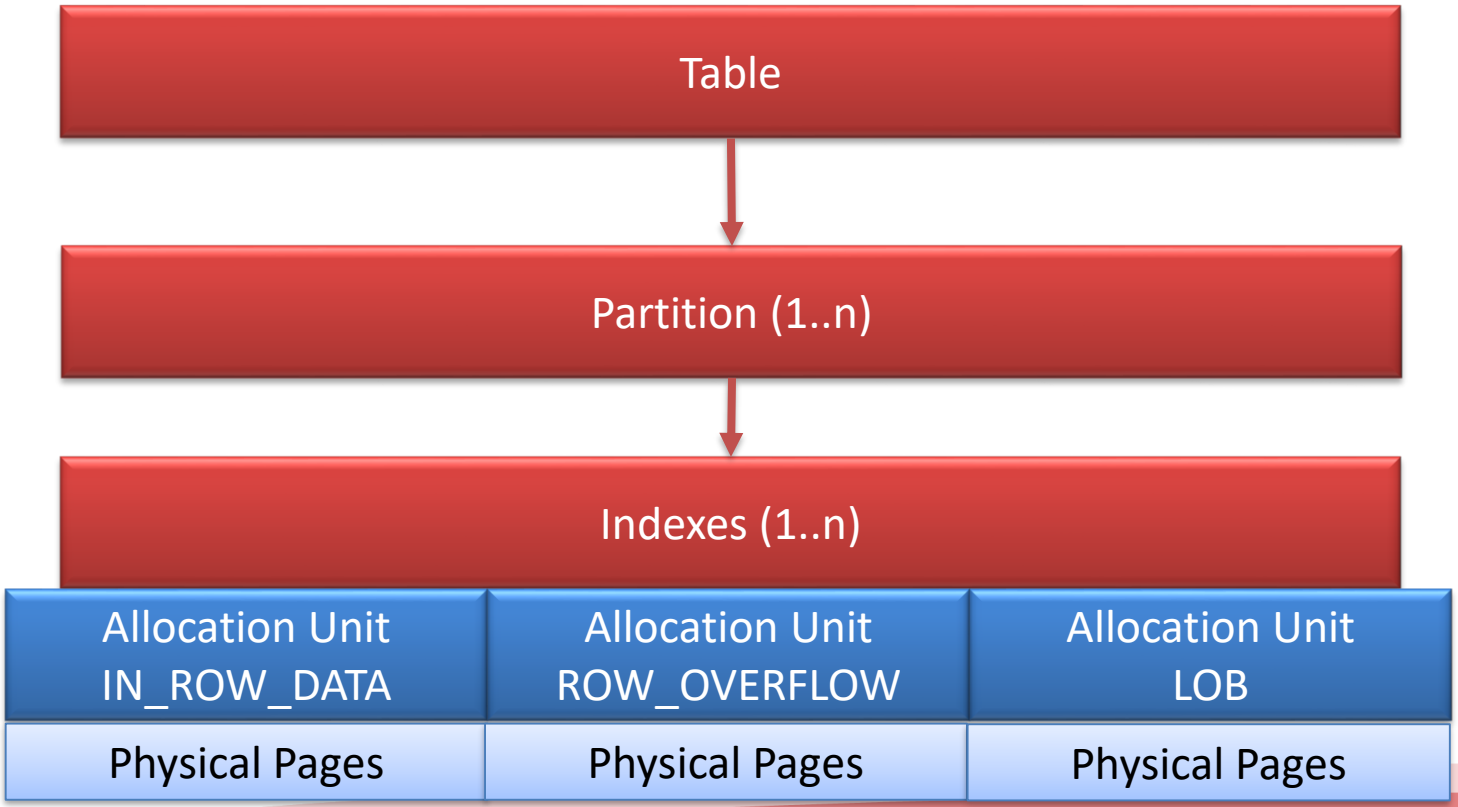
Allocation Units of Index



Index Allocation Units

demo

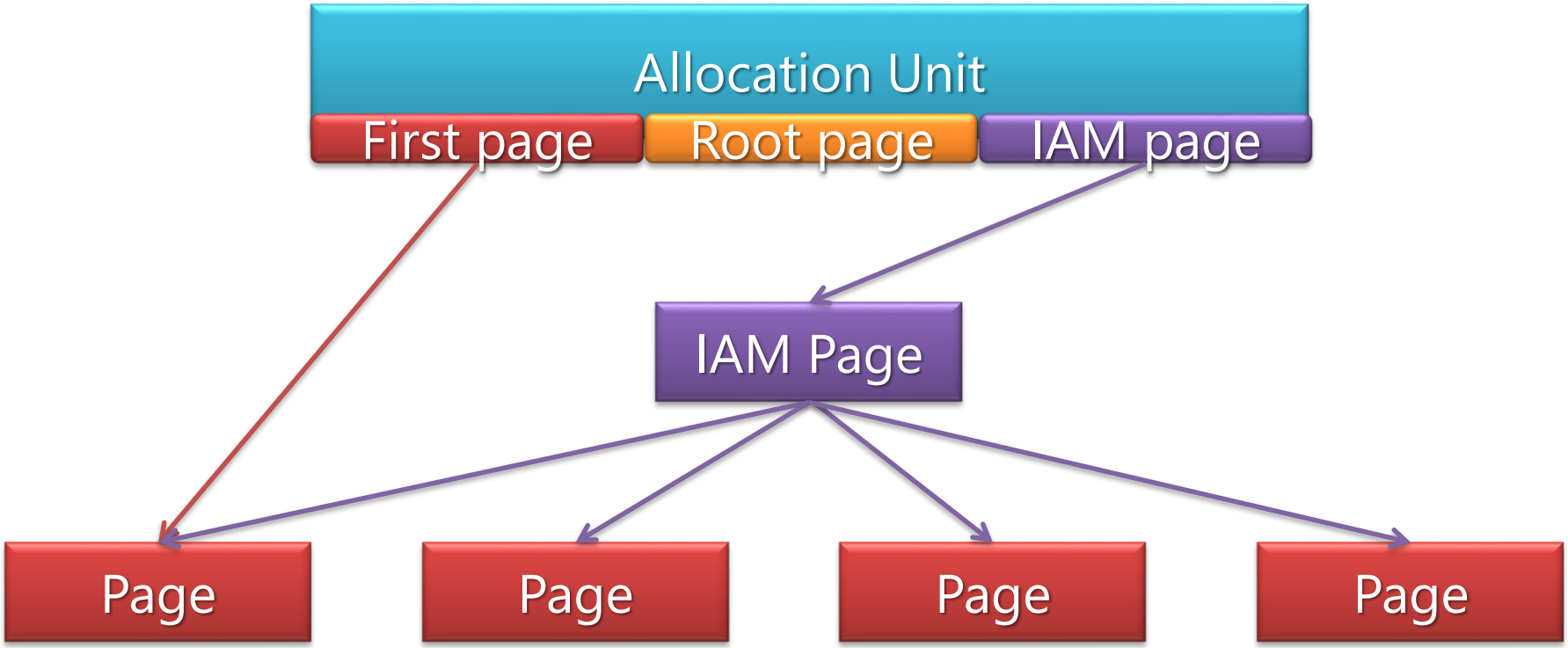
Table Structure



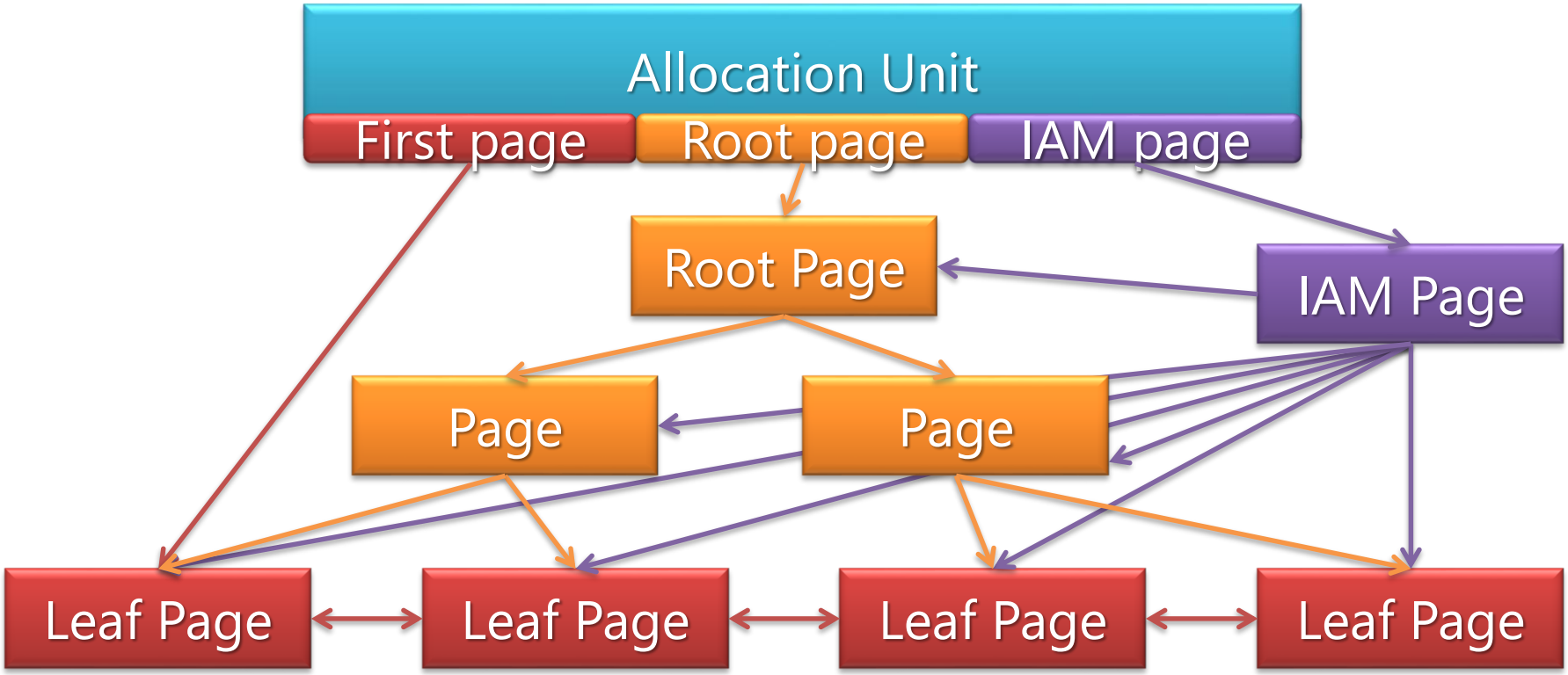
Data Pages Structures

- Heap – unsorted structure
  - No read-ahead and other optimizations
  - RID references – expensive reorganization, larger non-clustered indexes
- B-Tree – sorted structure
  - Most common usage
- Columnstore – column-based storage
  - For storing and querying large raw data tables

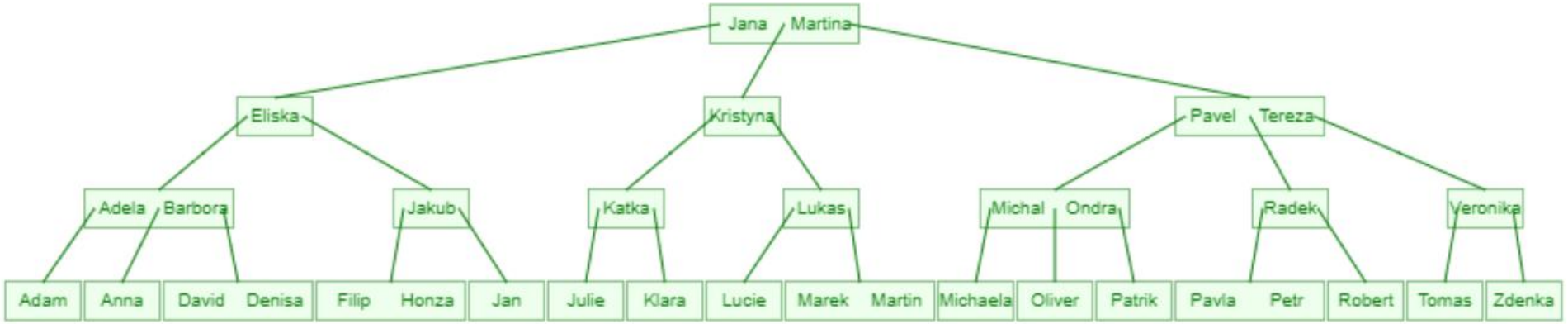
Index – heap



Index – b-tree



Index – b-tree index pages example

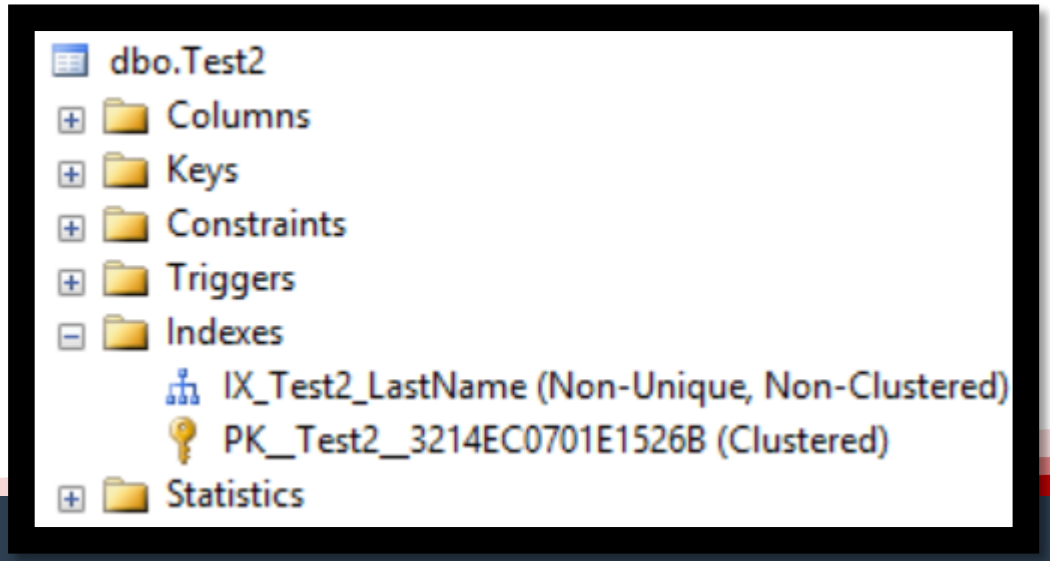


Tables

- Contains metadata:
  - Logical columns
  - Indexes / foreign keys / defaults / constraints / triggers…
- Physically stored as:
  - 1 main index – heap or b-tree (clustered index)
    - Contains all columns and rows
  - 0..n non-clustered indexes
    - Only selected columns (indexes column, included columns, reference to main index)
    - Only selected rows (filtered index)

Index Samples

```
create table [Temp.TestDatabase].dbo.Test2 (
	Id int not null primary key identity(1,1),
	FirstName nvarchar(500) not null,
	LastName nvarchar(500) not null,
	Email varchar(500) not null,
	IpAddress varchar(100) not null,
	index IX_Test2_LastName nonclustered (LastName ASC)
)
```



Index Samples

Id	FirstName	LastName	Email	IpAddress
1	Dody	Records	drecords0@marketwatch.com	195.68.247.229
2	Honey	Smale	hsmale1@slashdot.org	249.24.46.48
3	Guido	Gullis	ggullis2@1und1.de	98.243.35.227
4	Pammi	Hutson	phutson3@zimbio.com	26.97.217.184
5	Shanie	Pengelly	spengelly4@europa.eu	146.52.37.99
6	Vite	Deluze	vdeluze5@issuu.com	125.9.152.232
7	Dyana	Kowal	dkowal6@theglobeandmail.com	79.213.195.241
8	Viki	Golley	vgolley7@constantcontact.com	97.80.222.96

Clustered Index
Id ASC

LastName	Id
Deluze	6
Golley	8
Gullis	3
Hutson	4
Kowal	7
Pengelly	5
Records	1
Smale	2

Non-clustered Index
LastName ASC

What is included in non-clustered index?

- Value(s) of index (how it is sorted)
- Identification of row in main index
  - If main is heap – RID (row ID) = file, page, slot
  - If main is clustered – value of clustered + index (if not unique)
- Included columns
- If filtered index – only rows by predicate

Sample Index

demo

Unique index / Primary Key

- In most cases primary key = clustered index
- Regular b-tree structure
- SQL Server enforces uniqueness
- Difference PK vs unique
  - PK can't be NULL
  - PK can be on table only one
  - PK can't be filtered index

Primary key / Unique key

demo

Data operations

Buffer Management

- Pages reads from disk are cached
  - Physical vs logical reads
  - Having database in memory helps performance
- There can be delay between changing page and writing it to disk
  - Frequently changed pages don't to be written to disk with each operation
  - Write-ahead to log

Statistics And Execution Plan

- Why reading statistics and execution plan?
  - Understand how SQL Server processes queries
  - Verifying optimization
- Common read statistics types:
  - Time
  - Logical read
  - Physical read & Read-ahead read (only b-tree)
- Execution plan
  - What physical operations and indexes are used

Statistics / Execution plan

demo

Common Read Types

- Index scan
- Table scan (heap)
- Index seek
- Key Lookup
- RID Lookup (heap)

Read Types

demo

Rules for execution plan compilation

- Build up most resource efficient plan
  - As little page reads as possible
  - Fastest operations possible
- Choosing indexes:
  - Prefer read range over scan
  - Prefer small indexes over large

Index statistics

- Histogram of index values
  - Provides estimation of value distribution in specific range
- Default – enabled
  - Statistics are created and updated automatically if needed by engine

Statistics

demo

Covering Index

= index that covers all required columns and rows for given operation and key-lookup to main clustered index is then not required

- Can be created to optimize specific query / set of queries

Filtered index

- Only non-clustered index
- Index is created only for relevant rows
- Tuning of queries for specific areas of table
- Used if query filter is subset of index filter
- Filtered index advantages over full table indexes:
  - Reduced storage cost
  - Reduces maintenance costs
  - Improved query performance (better statistics)

Filtered Index

demo

Merge Strategies

- Merge Join – merging sorted streams
- Nested Loops – for each rows from input stream inner operation is invoked
- Hash Match – builds up hashtable in memory from first stream and then matches it with second stream
- Nested Loops with Table Spool – similiar as Hash Match but without hash table

Merge Strategies

demo

Aggregation Strategies

- Top – limits count of rows
- Sort – sorts input in temp database
- Distinct Sort – sorts and remove duplicates
- Hash Match Aggregate – creates hashset for each aggregate
- Stream Aggregate – aggregates rows of data sorted by aggregate

Aggregation Strategies

demo

When clustered index is not PK?

- PK always needs index:
  - Clustered by default
  - or non clustered
- Typically we don't do range scan over PK
- Use clustered key where range scan is common. Examples:
  - Date of log item – if most log queries are date range based
  - Thread Id of thread posts table – if loading thread together with posts
  - Country of customer address – if most reports are filters by country

Clustered key columns

demo

Tomáš Jecha

tomas@jecha.net