IT/Dev CONNECTIONS

Getting Your Database off to a Right Start

Andrew J. Kelly
Sr. Technology Subject Matter Specialist
B3 Group Inc.

Delivered by KNect365 TMT
an Informa business

Andrew J. Kelly

Andrew J. Kelly is a Sr. Technology Subject Matter Specialist with B3 Group Inc. and a Data Platform MVP with over 20 years' experience with relational databases and application development. He specializes in Performance, Scalability and Maintainability of large scale SQL Servers. He speaks routinely at various conferences around the world and has edited and written numerous articles and books throughout his career.

http://sqlblog.com/blogs/andrew_kelly/default.aspx

@GunneyK

B3 GROUP INCORPORATED

MVP Microsoft® Most Valuable Professional

Agenda

- Consistency
- Recovery Intervals
- Database File & File Groups
- Indexes
- Partitioning
- Schemas
- Configurations

- Questions & Answers

Consistency

- Consistency is the key to managing large numbers of databases
  - Object naming
  - Object placement
  - Configuration
  - Etc. etc.
- Model DB
  - This is used as a template for all new database creations
  - Utilize this for standard objects and settings
- Consider Policy Based Management

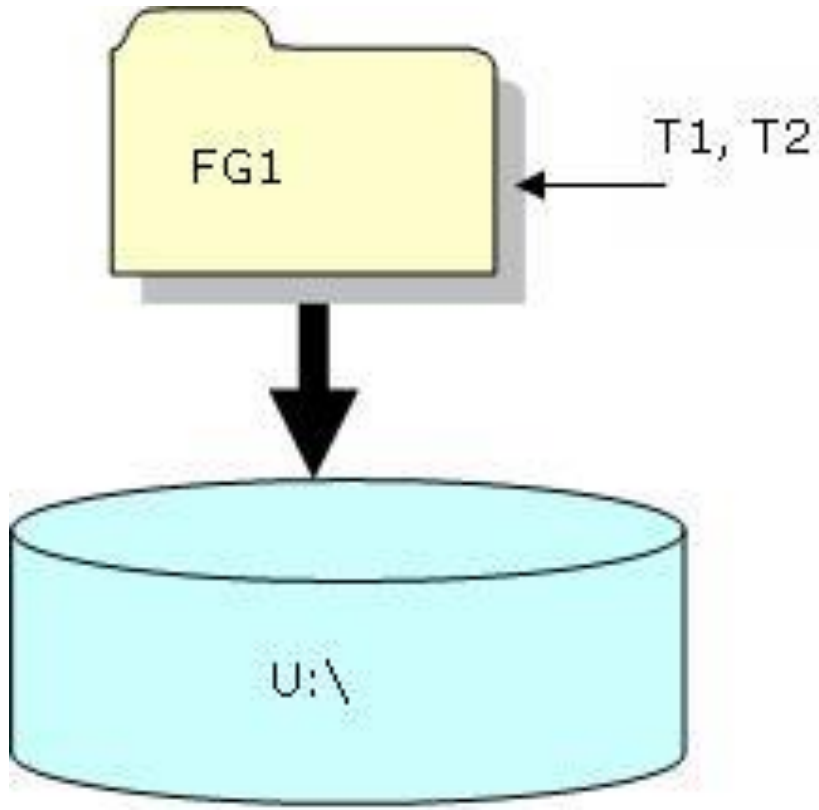
Data Types

- Choose your Data types wisely
- Here is where proper planning can really pay off
- Consider the following:
  - What are the requirements for a given column
    - Size
    - Accuracy
    - What will it hold? How will it be used?
  - Size does matter
  - Be careful of User Defined Data Types

Instant File Initialization

- The operating system and file system must support instant file initialization.
- The SQL Server startup account must possess the SE_MANAGE_VOLUME_NAME privilege. This privilege is required to successfully run SetFileValidData.
- The file must be a data file. Transaction Log files are not eligible for instant file initialization.
- Newer SQL Server installation wizard will set this for you

Files and File Groups



FileGroups

- What is a Filegroup?
  - A Logical grouping of one or more Physical files
  - For Data files only. Log files do not have a FG
  - FileGroups can not span databases
  - A single file can not span more than one FG
- The Primary FG always holds the system objects and may contain user objects
- Ensure that files within a FG are all the same size

FileGroups (cont)

- What good is a Filegroup?
  - Spreads objects across multiple drive arrays
  - Separates Objects (IE: data & indexes)
  - Finer level of backup and recovery
  - Separates the system objects from the user objects

Files & Filegroups

- How many Filegroups should you have?
  - Depends but BP is at least 2
    - Primary Filegroup for the system objects
    - Secondary Filegroup(s) for user objects

- You may want to separate objects into different Filegroups for future growth
  - Can be difficult to determine up front what should go where

- Partitioning will often drive your FG scheme

Files & Filegroups

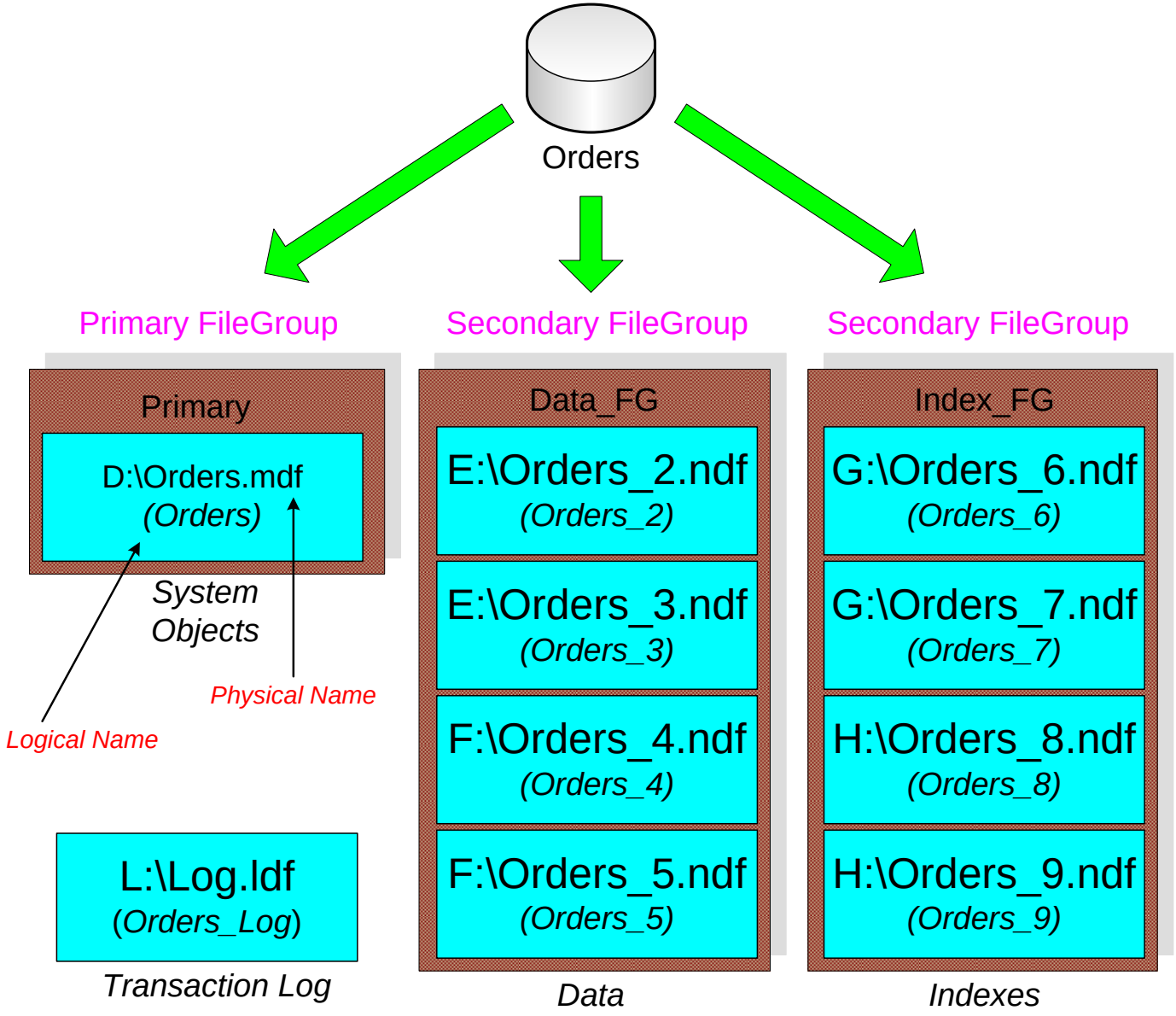
- How many Files in each Filegroup?
  - User Databases also depends
    - Average small to midsize databases get by fine with a single file
    - Larger databases should consider multiple files
    - Extreme activity may warrant more files
      - Contention for internal bitmaps (SGAM, PFS etc.)
      - Extremely heavy reads or writes
  - Tempdb usually benefits from multiple files for contention reasons

Files & Filegroups

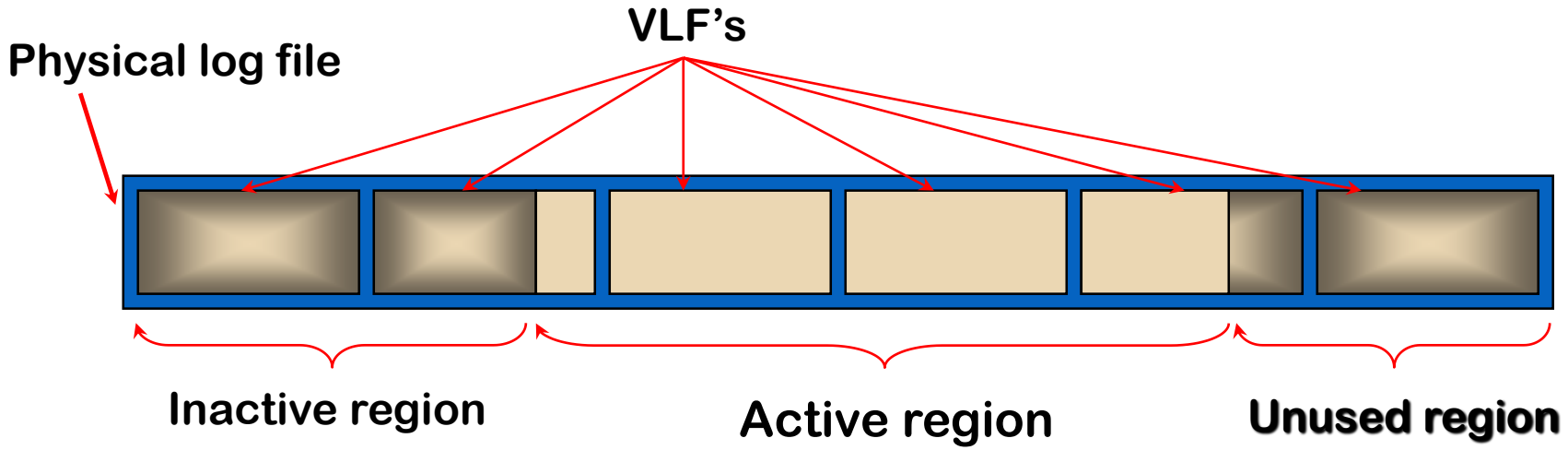
- All files in the same FG must be the same size
- Change Autogrow settings to be a fixed amount in MB or GB
- Set the Autogrow to the same size across all files in a given File Group
  - Prior to SQL2016 use trace flag T1117 to for all files in a FG to grow at once
  - After SQL 2016 use ALTER DATABASE [DB Name] MODIFY FILEGROUP [FG Name] AUTOGROW_ALL_FILES
- Don’t count on Autogrow to get you by. Be proactive instead
- Tran Log should be a single file

TempDB Contention

- Recommend use of trace flags
  - T1117 (Grow all files equally)
  - T1118 (No Shared extents)
- Default behavior for SQL2016 in Tempdb
  - For user dbs refer to the AUTOGROW_ALL_FILES db option


Orders
Primary FileGroup
Secondary FileGroup
Secondary FileGroup
Primary
D:\Orders.mdf
(Orders)
System Objects
Logical Name
Physical Name
L:\Log.ldf
(Orders_Log)
Transaction Log
Data_FG
E:\Orders_2.ndf
(Orders_2)
E:\Orders_3.ndf
(Orders_3)
F:\Orders_4.ndf
(Orders_4)
F:\Orders_5.ndf
(Orders_5)
Data
Index_FG
G:\Orders_6.ndf
(Orders_6)
G:\Orders_7.ndf
(Orders_7)
H:\Orders_8.ndf
(Orders_8)
H:\Orders_9.ndf
(Orders_9)
Indexes

Transaction Log

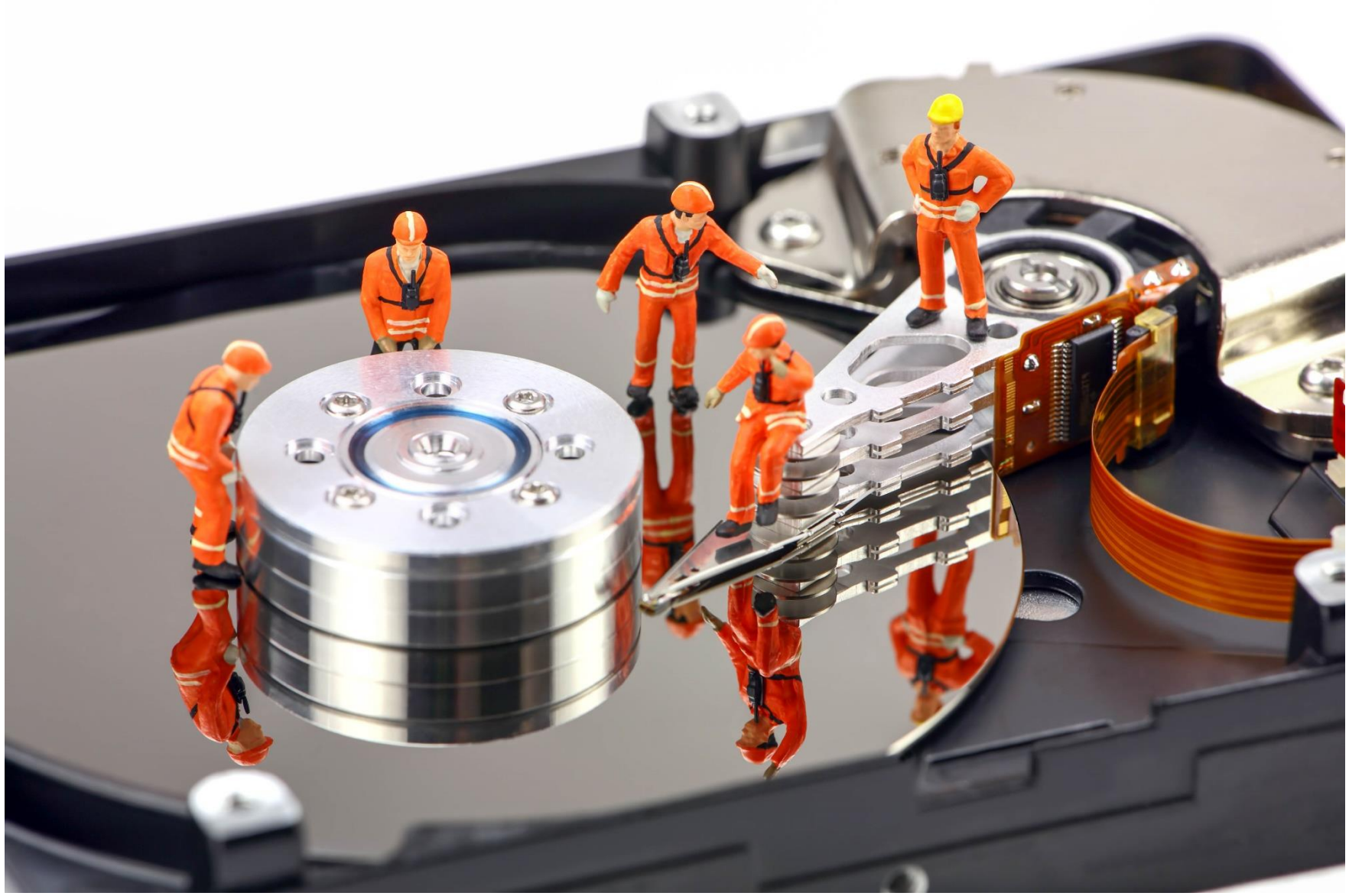


- Log files are comprised of smaller Virtual Log Files or VLF's
- DBCC LOGINFO or sys.dm_db_log_info() will show the status and size of each VLF
- Keep the # of VLF's reasonable
  - Lets talk about what reasonable is

Demo

- VLF’s

File Placement



Tables & Indexes

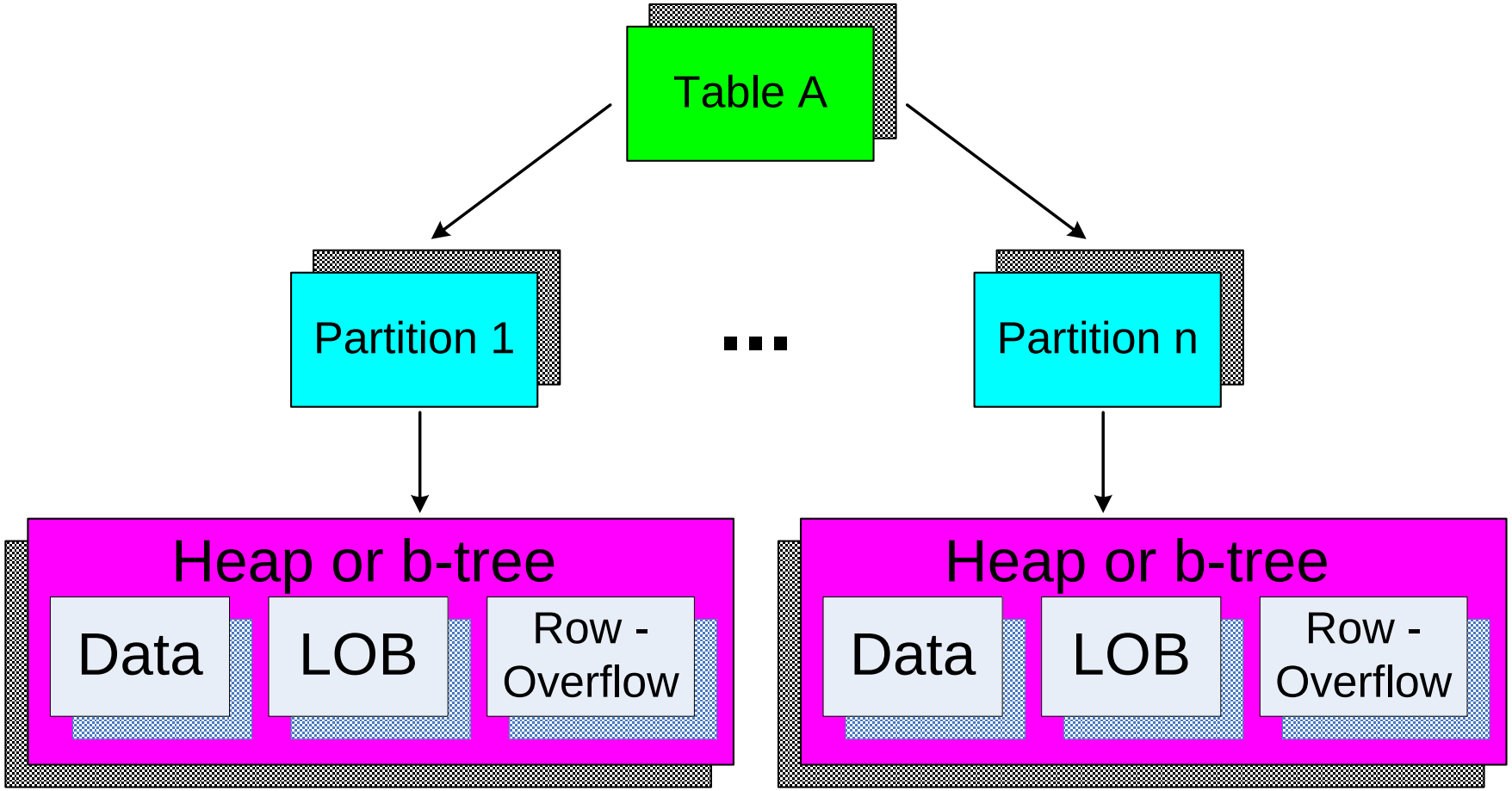


Table & Index Structures

- HEAP
  - Table with no Clustered Index
  - Data is in un-sorted order
  - Pages are not linked in a doubly-linked list
  - Uses the PFS pages when inserting a new row
  - Data placement is not configurable or controllable

Heap

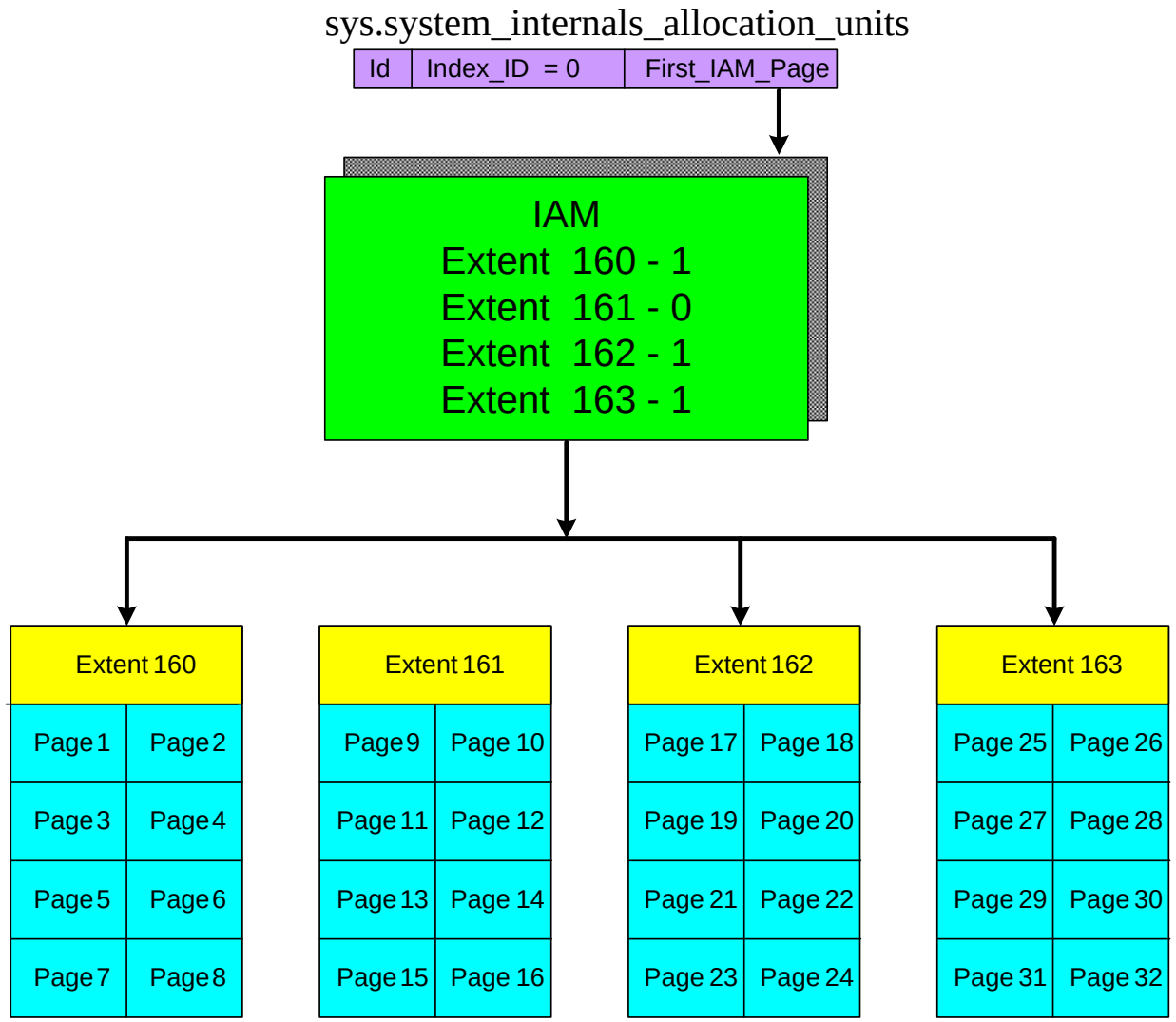

sys.system_internals_allocation_units
Id
Index_ID = 0
First_IAM_Page
IAM
Extent 160 - 1
Extent 161 - 0
Extent 162 - 1
Extent 163 - 1
Extent 160
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
Page 8
Extent 161
Page 9
Page 10
Page 11
Page 12
Page 13
Page 14
Page 15
Page 16
Extent 162
Page 17
Page 18
Page 19
Page 20
Page 21
Page 22
Page 23
Page 24
Extent 163
Page 25
Page 26
Page 27
Page 28
Page 29
Page 30
Page 31
Page 32

Table & Index Structures

- Clustered Index
  - Table with a Clustered Index
  - Data is physically sorted in the order of the index expression value
  - Only one clustered index per table
  - Implemented as a B-tree index structure
  - Pages are linked in a doubly-linked list

Clustered Index

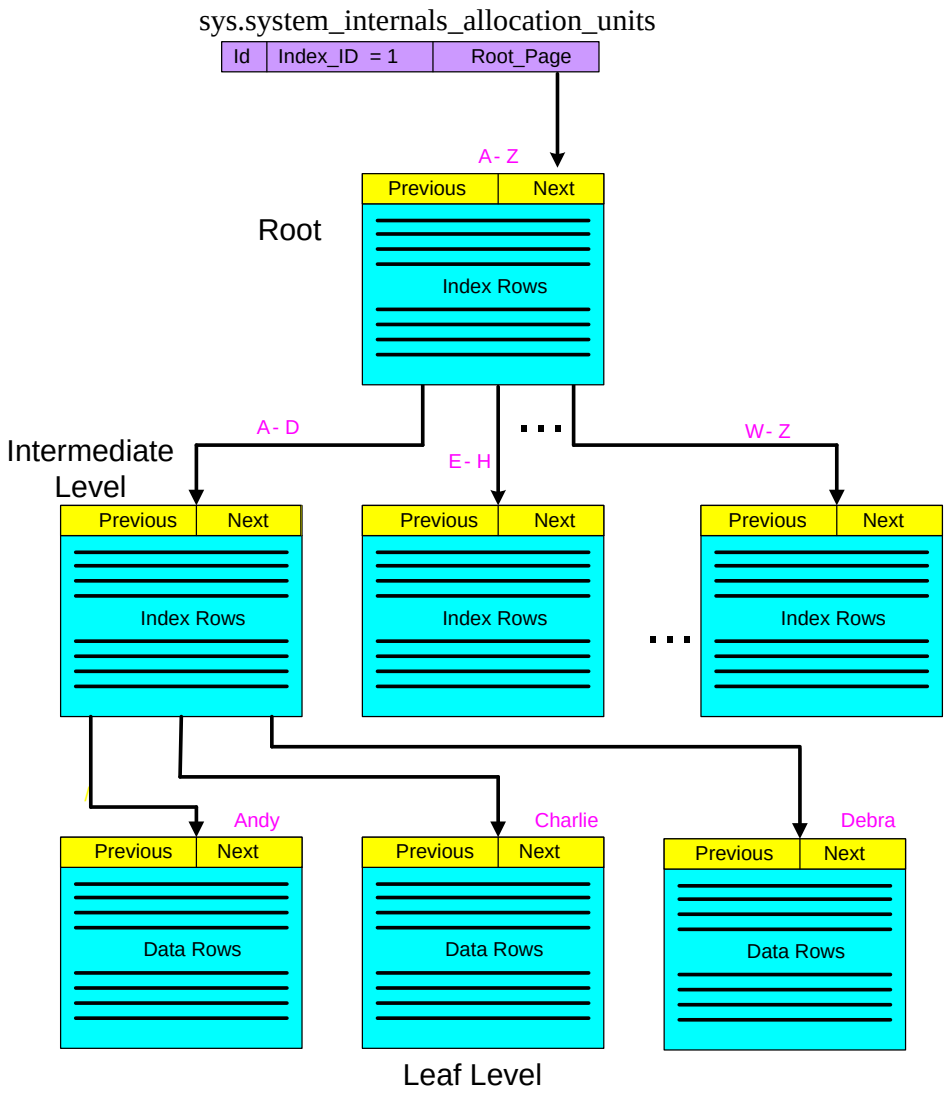
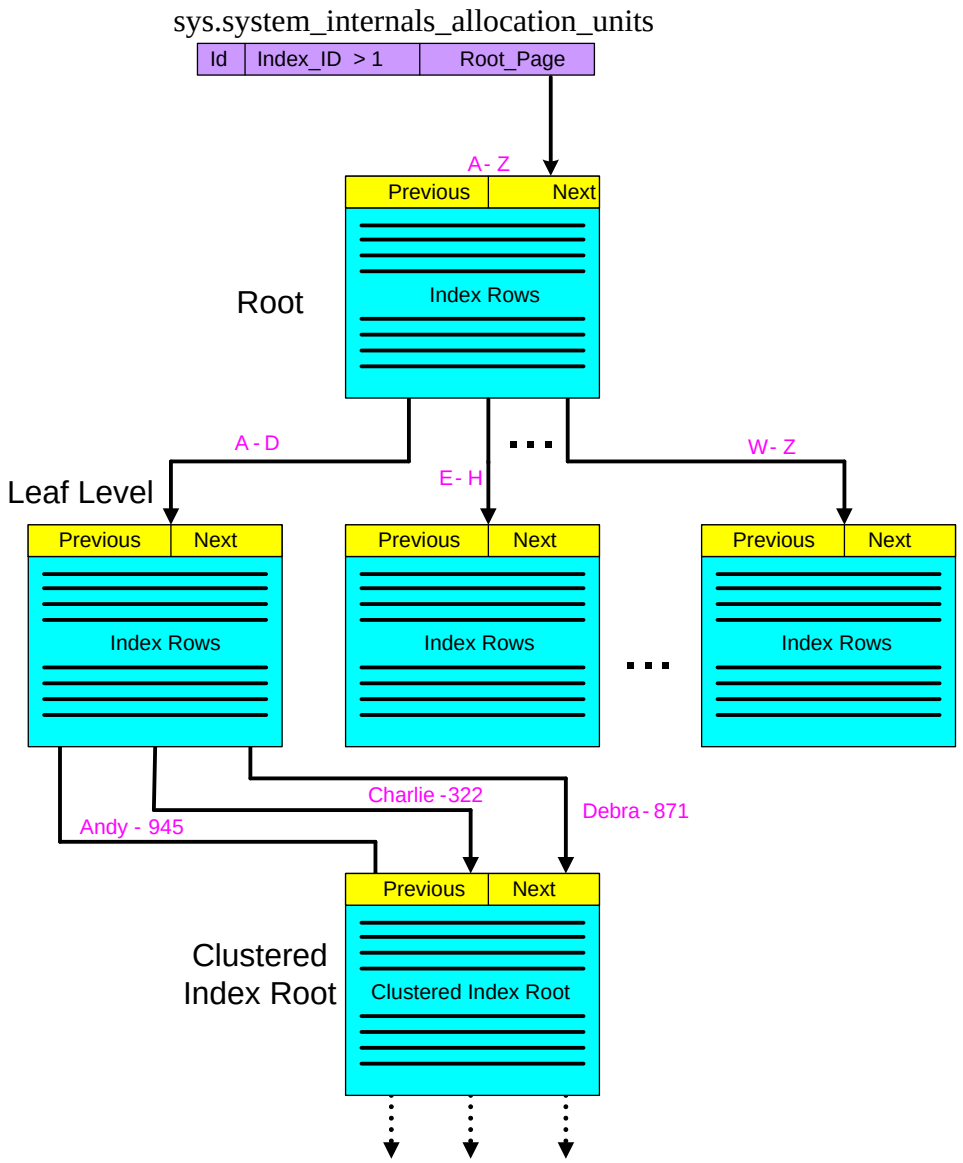


Table & Index Structures

- Non-Clustered Index
  - Maintained in a separate index page different from the data pages
  - Data is not sorted
  - Index key value has pointers to the data rows
  - Row locater can be RID (file #, page #, slot #) or unique clustered index key
  - Also implemented as a B-tree index structure

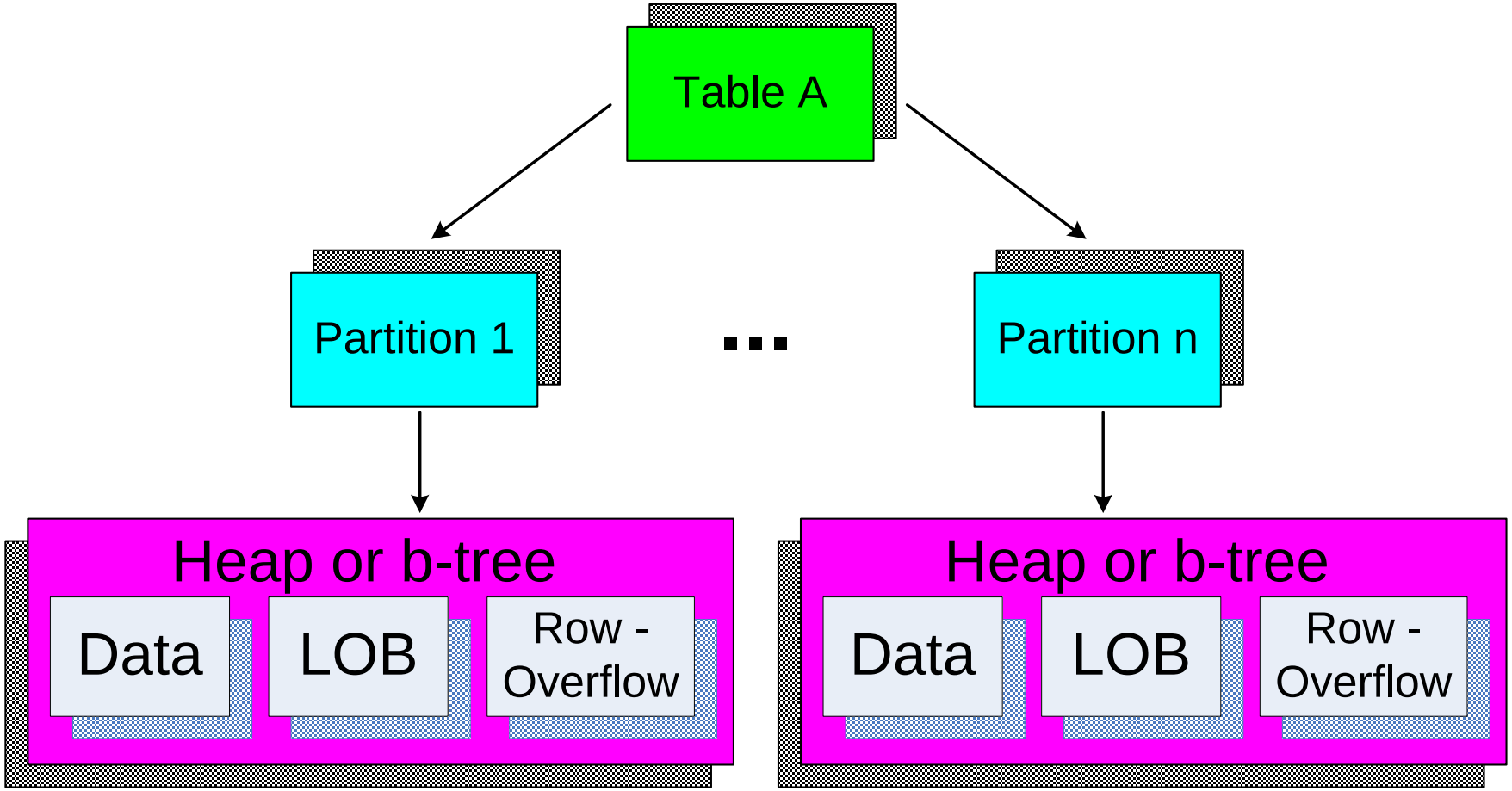
Non-Clustered Index



Heap & Index Choices

- Which is better, Heap or Clustered?
  - Most tables are recommended to have a clustered index
  - Heaps can be useful too
- What is the best choice for an index?
  - Obviously that depends
    - Clustered
      - Great for Range Scans
      - Great for singleton lookups to avoid bookmark
    - Non-clustered
      - All other indexes
      - Great for singleton or small #’s of lookups
      - Can use Included columns to make covering
- What about the Fill Factor & Pad Index?

Partitioning



Partitioning

- Is first and foremost a data management tool
  - Performance may be a side effect
- Allows you to manage data as if each partition was a separate physical object but appears to the end user as a single logical object (over simplification but you get it)
  - Allows for certain operations on aligned indexes
    - Separate index rebuilds
    - Separate statistics updates
      - Still one histogram
  - Switching partitions in and out can be a meta data operation
- This should be decided long before you start loading data
  - Will dictate what indexes you have
  - Will move data if created or changed after the fact
  - May dictate how you use File and File Groups

Schema’s

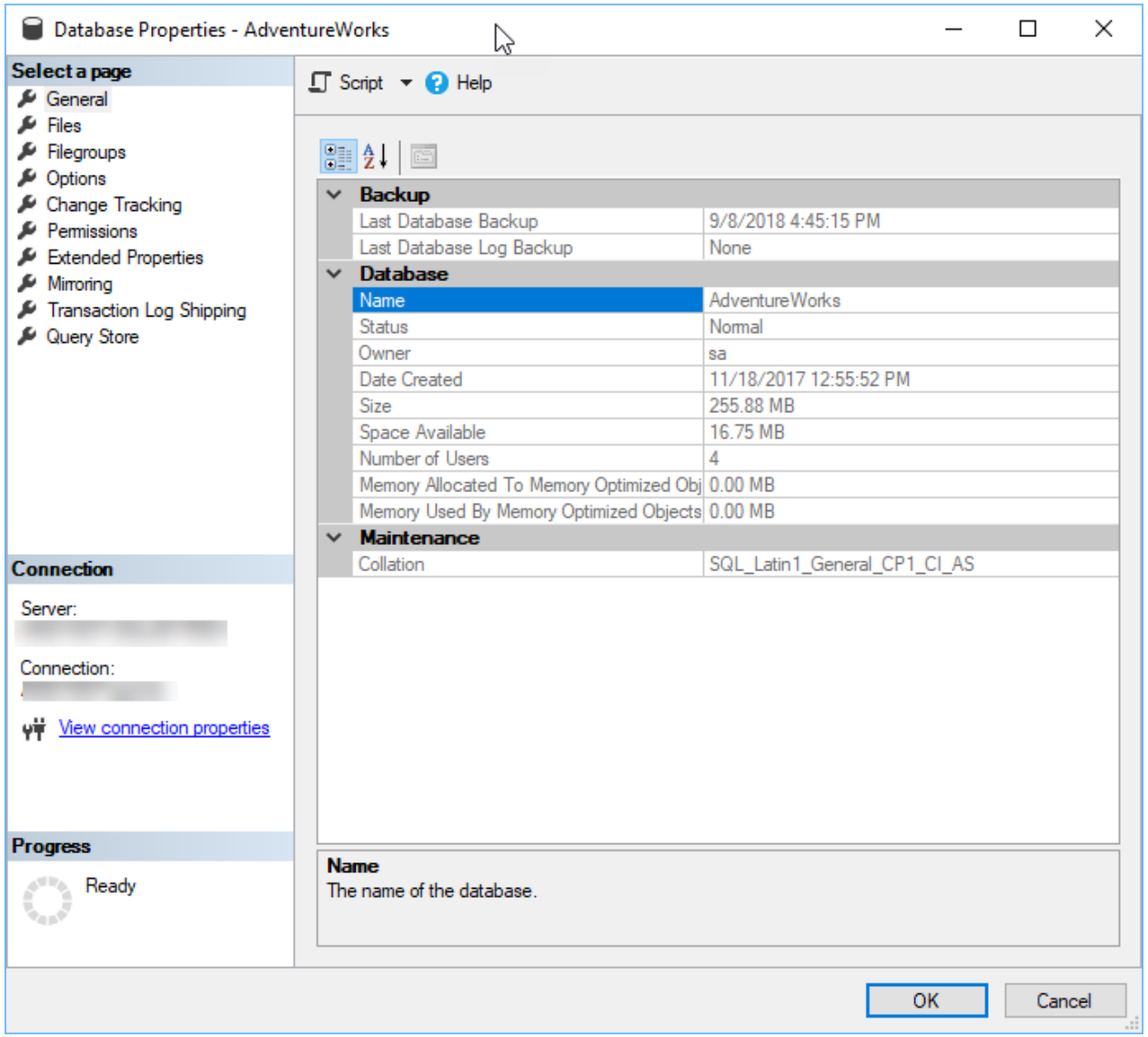
- A schema is a collection of database entities that form a single namespace
- A namespace is a set in which every element has a unique name
- Server.Database.Schema.Object
- Use the built in segregation that Schema’s allow to better manage access to various objects

Configuration

Where do you start?

- Never change anything f[illegible]s unless you have a good reason to
  - This is probably the single [illegible]BP
- Just because yo[illegible] good reason☺
- A good reason also [illegible]you understand why you need to change it and what the chan[illegible]lo
- Most SQL Servers [illegible]t fine with little or no changes to the default settings

Database Properties

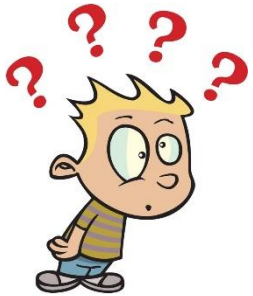


Demo

- Database Properties

Shrink Database / Logs

- When should you shrink a data or log file?



- You have to be careful or you may not get the result you expected.

Questions