Microsoft

# Database Maintenance

Module 2

# Learning Units covered in this Module

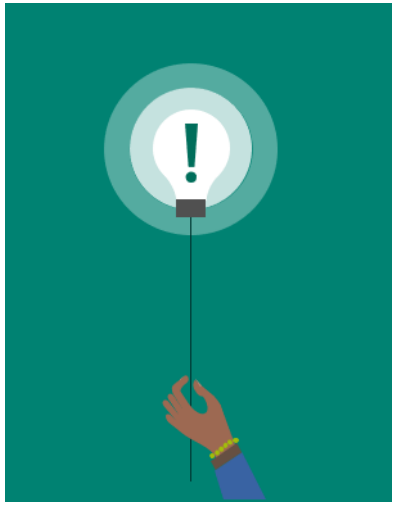
- Lesson 1: Database Configuration
- Lesson 2: Indexes and Statistics
- Lesson 3: Database Maintenance

# Lesson 1: Database Configuration

# Objectives

After completing this learning, you will be able to:

- Explain various configuration settings available at the database level.
- Describe the DATABASE SCOPED CONFIGURATION command.



# Database files and filegroups

Database files

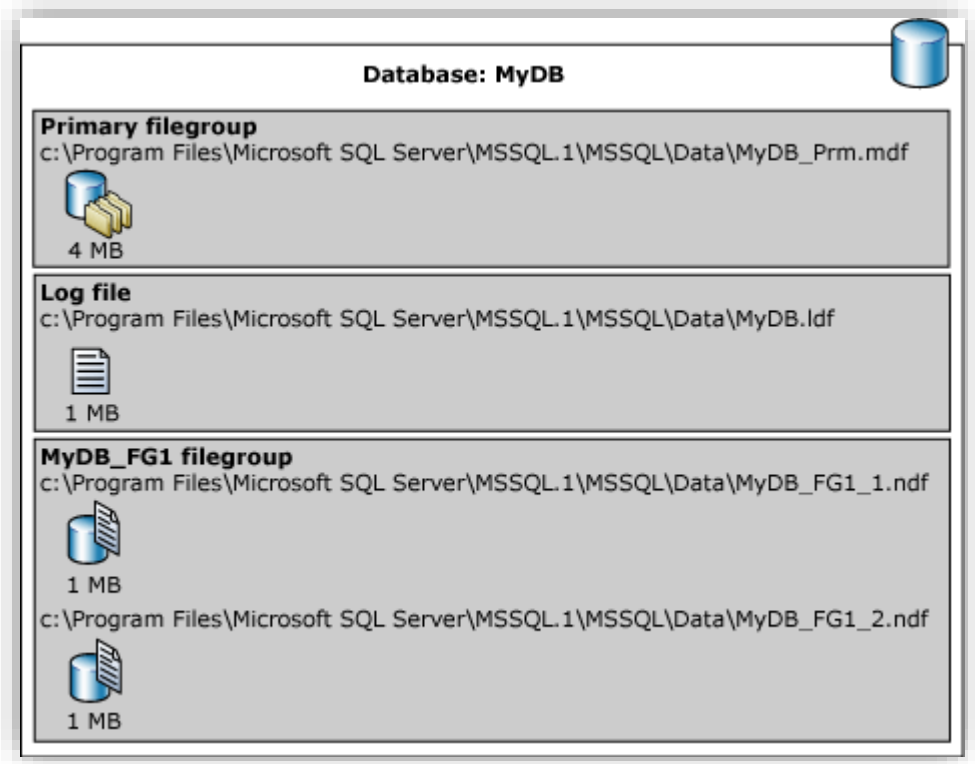
A database is composed by at least two operating system files:

## Data files

- Contain database objects and data
- First data file is called primary data file. This file has a .mdf extension
- A database may have additional data files, known as secondary data files. They use .ndf extension
- Can be grouped together in filegroups for allocation and administration purposes

## Log file

- Contain Log Records and entries are sequenced

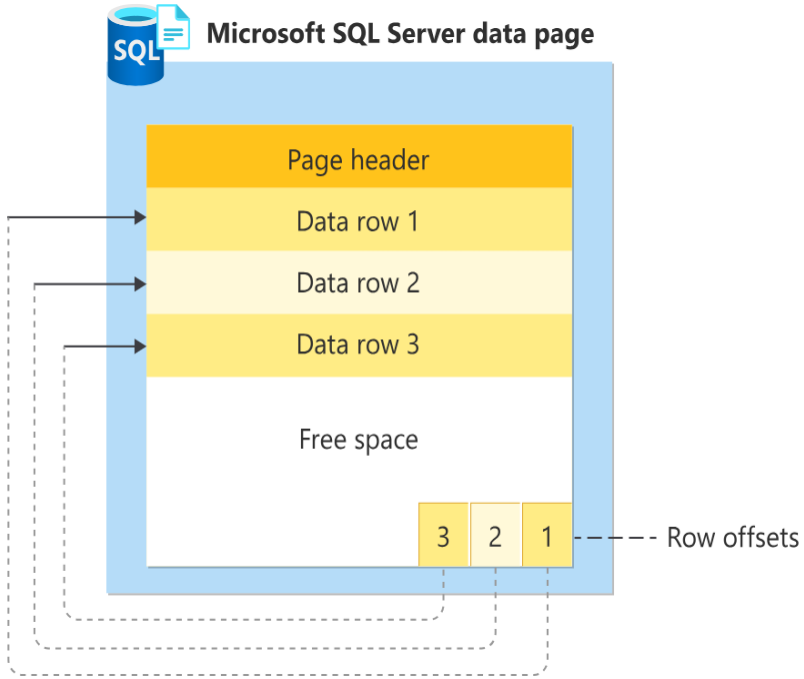


# Pages and Extents architecture

A data page is the fundamental unit of data storage in SQL Server.

- The disk space allocated to a data file (.mdf or .ndf) is logically divided into pages.
- Each page is 8 KB in size
- Pages are numbered contiguously from 0 to n.
- Disk I/O operations are performed at the page level.

Extents are a collection of eight physically contiguous pages (64KB) and are used to efficiently manage the pages.



# Database Configuration

Database Configuration Settings Categories

- General
- File/Filegroups
- Options
- Change Tracking
- Transaction Log Shipping
- Mirroring

# Database Configuration

General Database Configuration Settings

## Collation

- Governs how SQL Server sorts and compares data.

## Recovery Model

- Governs how transactions are stored in the transaction log.

## Compatibility Level

- Sets database functionality and behaviors to the SQL Server level specified.

## Containment Type

- Determines if a database is partially contained.

# Database Configuration

## Database AUTO Configuration Settings

### Auto Close

- Shuts down the database cleanly and frees resources after the last user exits

### Auto Shrink

- Allows for the periodic shrinking of database files
- If turned on, this can cause an “accordion effect”.  Generally recommended to be turned off

### Auto Create Statistics

- Determines whether a database automatically creates missing optimization statistics

### Auto Create Incremental Statistics

- Update statistics for only a specific partition

### Auto Update Statistics

- Automatic update of out-of-date statistics

### Auto Update Statistics ASYNC

- Updates statistics asynchronously

# Database Configuration

Database State Configuration Settings

## Database State

- Online, Offline, Restoring, Recovery Pending, Suspect, and Emergency

## Restricted Access

- Specifies which users may access the database (Multiple, Single, and Restricted)

## Encryption Enabled

- Determines whether a database is enabled for encryption

## Read Only

- Make the database read only

# Database Configuration

Other Database Configuration Options

## Parameterization

- Determines how parameterization of queries will be handled (SIMPLE versus FORCED)

## Page Verify

- Defines the SQL Server mechanism of verifying page consistency when the page is written to disk and when it is read again from disk

## Delayed Durability

- Accomplished using asynchronous log writes to disk

## Target Recovery Time

- Enables Indirect Checkpoint

# Database Configuration

DATABASE SCOPED CONFIGURATION

Implement some of the settings at the database level, which were earlier configurable only at the instance level. These are also customizable on Always On secondary replicas.

- CLEAR PROCEDURE_CACHE
- MAXDOP
- LEGACY_CARDINALITY_ESTIMATION
- PARAMETER_SNIFFING
- QUERY_OPTIMIZER_HOTFIXES
- IDENTITY_CACHE
- VERBOSE_TRUNCATION_WARNINGS
- LAST_QUERY_PLAN_STATS

```
SELECT * From sys.database_scoped_configurations;
```

# Demonstration

ALTER DATABASE SCOPED CONFIGURATION command

# SQL Server System Databases

## master

- Records all the system-level information for an instance of SQL Server.
- Contains metadata about other databases, logins, and configuration settings.

## model

- Serves as the template for all databases created on the SQL Server instance.
- Modifications made to the model database apply to new databases.

## msdb

- Used by SQL Server Agent for scheduling alerts and jobs.
- Stores information related to maintenance plans, and other administrative tasks.

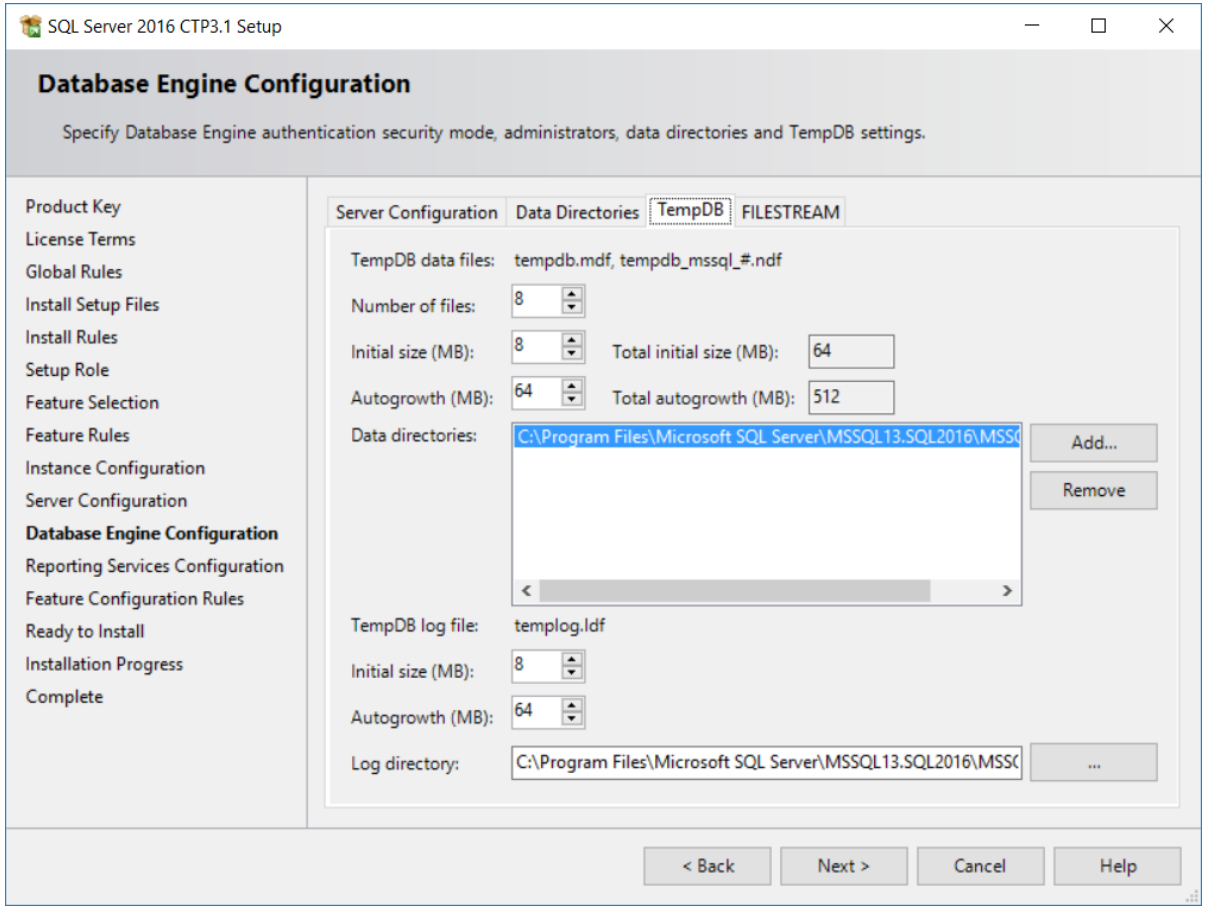
## tempdb

- Workspace for holding temporary objects or intermediate result sets.
- Used for sorting, temporary tables, and other temporary storage needs.

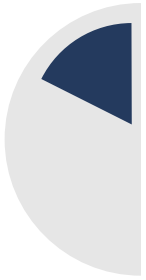
# Performance improvements in TempDB

Starting with SQL Server 2016 (13.x)

- Setup adds multiple TempDB data files during instance installation.
- By default, setup adds as many TempDB data files as the logical processor count or eight, whichever is lower.



# “Hands-free” tempdb



## Pre-SQL 2016

1 file = PFS, GAM, SGAM contention

Add multiple files

Trace flags 1117 and 1118

## SQL Server 2016

SQL 2016 setup automatically adds multiple files

Trace flags not required by SQL Server 2016 as

Autogrow and uniform default for TempDB

## SQL Server 2019

PFS concurrency

Now system table pages become hotspot

Tempdb metadata optimization ON

SGAM and GAM contention remain

## SQL Server 2022

SGAM and GAM concurrency

Latch contention gone

# Questions?

# Knowledge Check

Which version of SQL Server first introduced the ALTER DATABASE CONFIGURATION SET MAXDOP option?

What is the recommended setting for the PAGE_VERIFY database option?

What are the recommended and default settings for the AUTO_CREATE_STATISTICS and AUTO_UPDATE_STATISTICS options?

True/False: SQL Server DBAs can change the database state value from Emergency to Online in the Database Properties in SSMS.

Which database option allows updating statistics for a specific partition?

# Lesson 2: Indexes and Statistics

# What is an Index?

An index is an on-disk structure associated with a table that speeds retrieval of rows.

An index contains keys built from one or more columns in the table.

# How Data is Stored in Data Pages

Data stored in a Heap is not stored in any order and normally does not have a Primary Key.

Heap

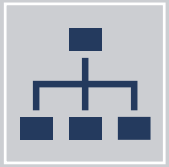
| AcctID | AcctName | Balance |
|---|---|---|
| 1 | Jack | 500.00 |
| 2 | Diane | 750.00 |
| 29 | Kelli | 1250.00 |
| 27 | Jessica | 1005.00 |
| 18 | Maddison | 745.00 |
| 22 | Bella | 445.00 |

Clustered Index data is stored in sorted order by the Clustering key. In many cases, this is the same value as the Primary Key.

Clustered Index

| AcctID | AcctName | Balance |
|---|---|---|
| 1 | Jack | 500.00 |
| 2 | Diane | 750.00 |
| 12 | Danny | 630.00 |
| 14 | Mayleigh | 204.00 |
| 15 | Molly | 790.00 |
| 18 | Maddison | 745.00 |

# Clustered Indexes



An ordered data structure that is implemented as a Balanced (B) Tree.

When a table is clustered, the leaf level of the index contains ALL data in the table. This means that the clustered index IS the table. This is also why there is only one per table.

The leaf level of the clustered index contains data pages.

# Non-Clustered Indexes

Same B-tree data structure as a clustered index.

It is an independent structure that is created from a Heap or Clustered Index.

Only contains a subset of the columns in the base table

The leaf level contains only the columns defined in the index as well as the clustered key/heap ID that points to the base table structure.

# Clustered vs Nonclustered Indexes

An index is an on-disk structure associated with a table or view that speeds retrieval of rows.

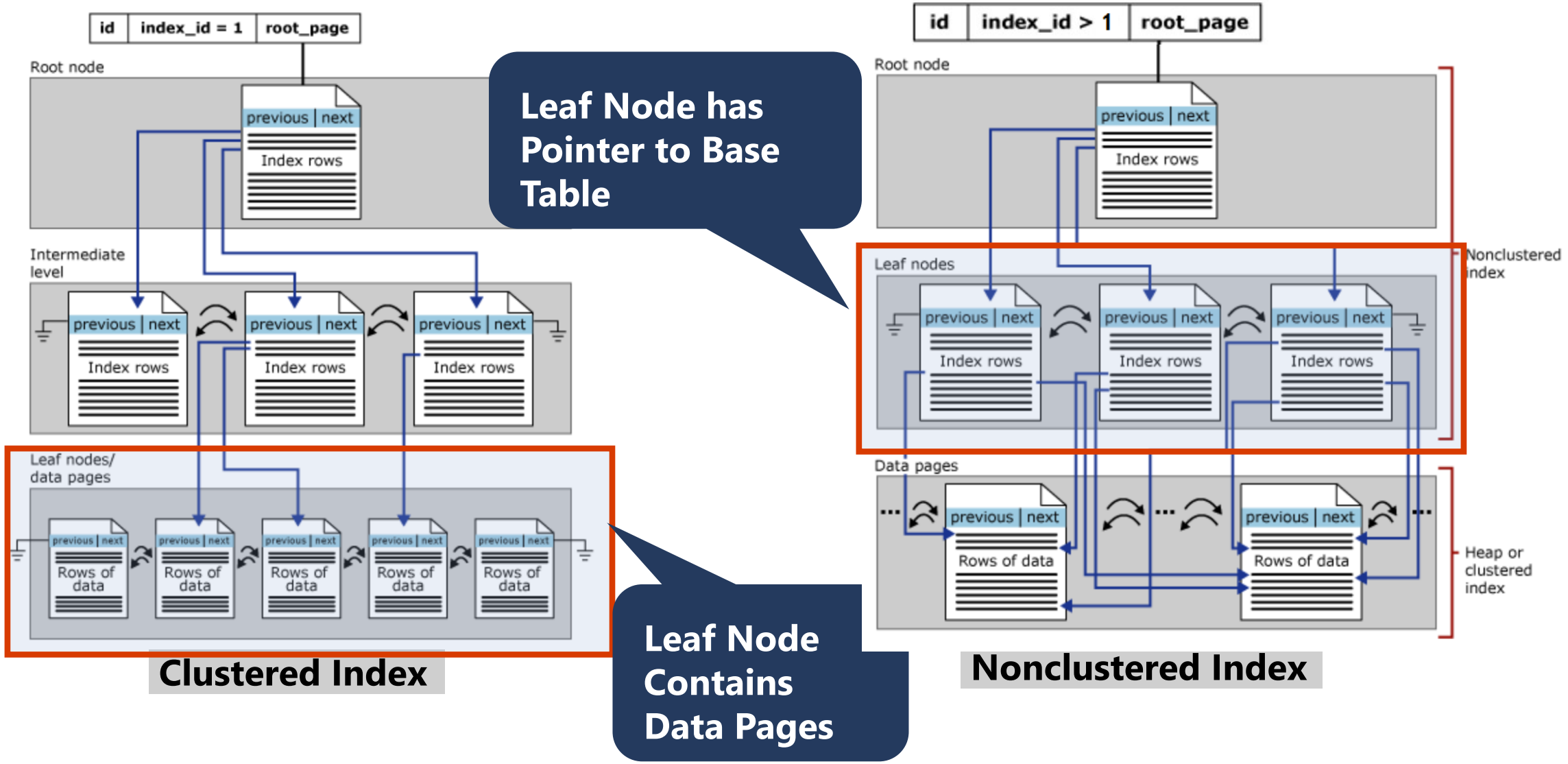
## Clustered Indexes

- Defines the order in which data is physically stored in a table.
- Table data can be sorted in only one way.
- Leaf level has data rows stored with index.
- When a table has a clustered index, the object is called clustered table.
- Cluster key is added to nonclustered index (as the pointer), keep it as narrow as possible.

## Non-clustered Indexes

- Separate structure from base table.
- Contains a pointer back to base table called:
  - Row ID (RID when base table is HEAP)
  - Key (KEY when base table is Clustered)
- "Skinny" data structure as it contains a subset of base table only.
- To by-pass index key limits (1,700 bytes) ,non-key columns can be added to leaf level.
- As Leaf level contains fewer columns than base table, the non-clustered index uses fewer pages than the corresponding base table.

# Comparing Clustered and Non-clustered indexes



# Fragmentation

A fragmented table/Index is when some of its data pages point to pages that are not in sequence.

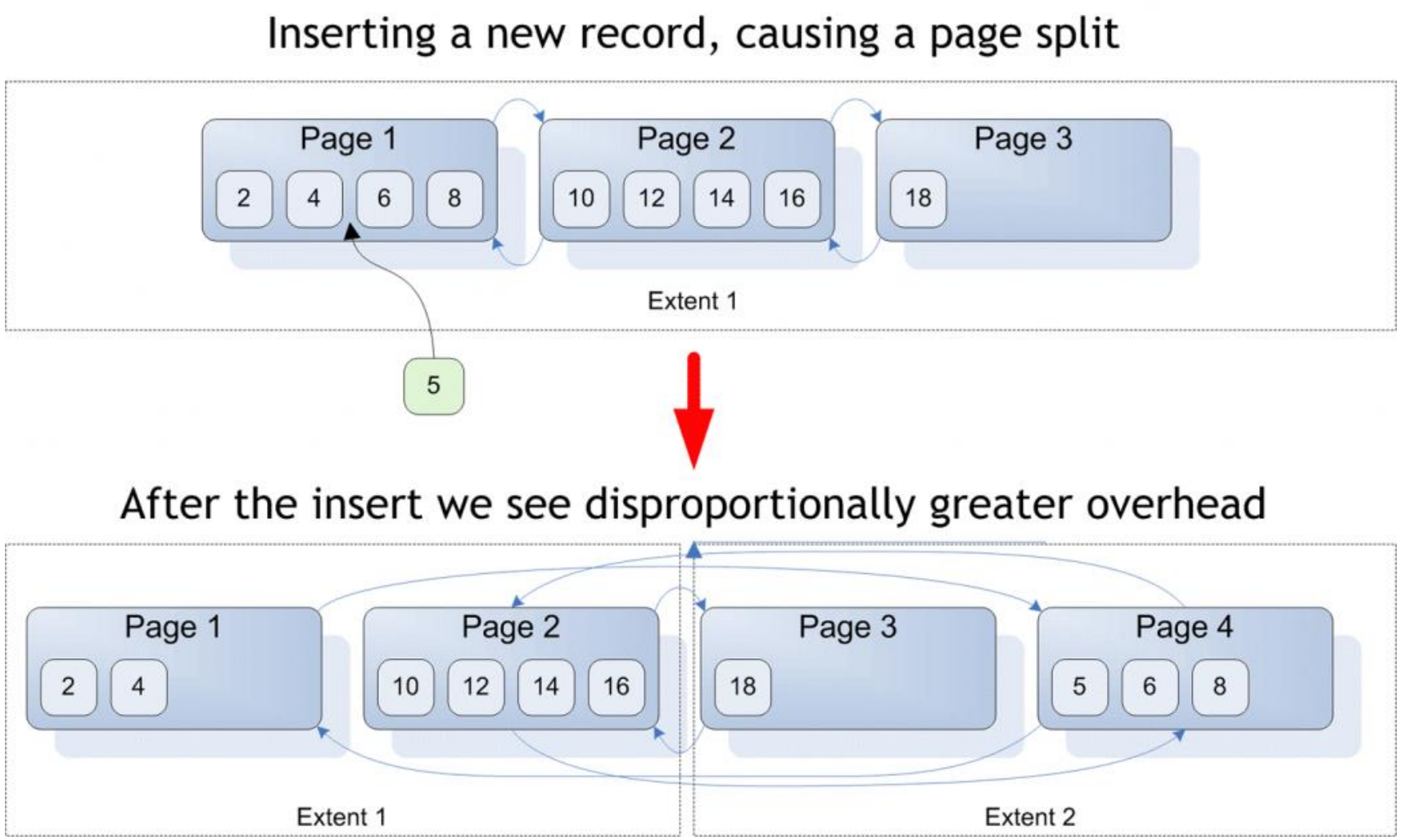
## Logical fragmentation

- Occurs when leaf level pages are not physically corresponding to the logical order of the index:
  - Pages are not in the most efficient order for scanning purposes.
- Limits the efficiency of read-ahead scans, but not seeks.

## Page density

- How full a page is when a rebuild/reorganization occurs.
- The fuller a page is, the more likely page splits occur when data is modified.
- The less full a page is, the more wasted space in the buffer pool when reading pages.

# Page Splits


Inserting a new record, causing a page split
Page 1
2
4
6
8
Page 2
10
12
14
16
Page 3
18
Extent 1
5
After the insert we see disproportionally greater overhead
Page 1
2
4
Page 2
10
12
14
16
Page 3
18
Page 4
5
6
8
Extent 1
Extent 2

# SQL Server Statistics

What are statistics?

Statistics contain statistical information about distribution of values in one or more columns of a table or index.

It is stored as binary large objects (BLOBs).

It is used by the Query Optimizer (QO) to estimate the *cardinality*, or number of rows, in the query result, and enable the creation of high-quality query plans.

Statistics are created:

- Intrinsically when indexes are created.
- Manually by using CREATE STATISTICS command.
- Automatically, to support WHERE clauses (if AUTO_CREATE_STATISTICS in ON).

# Automatically created statistics

Database options that affects automatic statistics creation and update

AUTO_CREATE_STATISTICS

AUTO_UPDATE_STATISTICS

AUTO_UPDATE_STATISTICS_ASYNC

INCREMENTAL

Use the defaults unless you NEED to do otherwise.

Do not enable auto-create statistics on SharePoint content database.

The small sample rate of AUTO_UPDATE_STATISTICS can cause some workloads to choose sub-optimal execution plans.

# Manually created Statistics

For most queries, the query optimizer generates necessary statistics for a high-quality query plan.

In a few cases, additional statistics is needed to improve query performance.

```
CREATE STATISTICS ContactPromotion1
ON Person.Person (BusinessEntityID, LastName, EmailPromotion)
```

# Available tools to review statistics

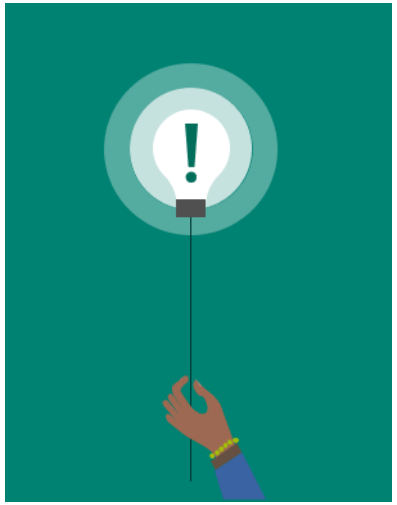
| | Metadata | Last Update Date | Sampling Rate | Row Mod Counter | Density Vector | Histogram |
|---|---|---|---|---|---|---|
| sys.stats | X | | | | | |
| STATS_DATE() | | X | | | | |
| sys.dm_db_stats_properties (object_id, stats_id) | | X | X | X | | |
| sys.dm_db_incremental_stats_properties (object_id, stats_id) | | X | X | X | | |
| sys.dm_db_stats_histogram (object_id, stats_id) | | | | | | X |
| SQL Server Mgt Studio | X | X | X | | X | X |
| DBCC SHOW_STATISTICS | | X | X | X | X | X |

# Lesson 3: Database Maintenance

# Objectives

After completing this learning, you will be able to:

- Explain various database maintenance activities such as index defragmentation, statistics maintenance, DBCC Checks.
- Explain the differences between index rebuild and index reorganize.
- Describe smart index maintenance.



# Database Maintenance

Checklist

## Data and Log File Management

## Index Maintenance

- Reorganize and Rebuilding Indexes
- Online Index Maintenance

## Statistics Maintenance

## Integrity Checks

## Smart Maintenance

- Maintenance Plans
- Custom solution

# Database Maintenance

Database File Size Management

## Monitor available space in data and log files

## Adjust auto growth settings on data and log files

- Fixed Size Growth (Recommended)
- Percentage Growth
- Enable Instant File Initialization to reduce impact of autogrowth
- Defragment database file system if lots of autogrowth events have occurred

## Monitor Virtual Log File (VLF) Fragmentation

- Monitor with the sys.dm_db_log_stats dynamic management view.

## Shrinking database and log files

- Not recommended as a regular maintenance task
- Don’t forget to defragment indexes after shrinking database files

# Database Maintenance

Index Fragmentation and Maintenance

## Types of Index Fragmentation

- Internal Fragmentation
- Logical (aka External) Fragmentation

## Monitor index fragmentation

- sys.dm_db_index_physical_stats data management function

## Avoid Fragmentation

- Page Splits and Fill Factor

## Address index fragmentation

- Index Reorganize versus Index Rebuild
- CREATE INDEX…WITH (DROP_EXISTING=ON)

## Online Index Rebuild

## Resumable Online Index Rebuild

# Database Maintenance

## Statistics Maintenance

### Statistics Maintenance is important

- For good query plans and optimal performance

### Two types of Statistics - Index Statistics and Column Statistics

### Statistics sample size is important

### Automatic Statistics Update

- 20% threshold to kick in automatic update of statistics
- Trace Flag 2371 (enabled by default)

### Index Rebuild will update statistics associated with the index

### Manual Statistics Update

- UPDATE STATISTICS command
- Sp_updatestats

### Incremental Statistics

- Creates, stores and refreshes statistics on specific partitions thus reducing maintenance time

# Database Maintenance

## Database Integrity Checks

### Why database integrity checks are important

- Databases may be come corrupt for many reasons
- Recovery relies on proactive detection and mitigation

### Commands to perform database integrity checks

- DBCC CHECKDB (most used)
- DBCC CHECKALLOC
- DBCC CHECKCATALOG
- DBCC CHECKTABLE

### Repair Options

- REPAIR_FAST
- REPAIR_REBUILD
- REPAIR_ALLOW_DATA_LOSS

# Database Maintenance

Smart Maintenance

## Maintenance Plans

- Backup database and log files
- Check database integrity
- Reorganize index
- Update statistics
- Shrink database
- Rebuild index
- Maintenance cleanup

# Demonstration

Viewing index fragmentation

# Index Maintenance

- Identifying and removing physical index fragmentation

# Statistics Update

- Observing Automatic statistics update
- Updating Statistics by executing ALTER INDEX

# Questions?

# Knowledge Check

Which new dynamic management function was introduced in SQL Server 2017 to monitor VLF fragmentation?

What is the disadvantages of shrinking database and transaction log files?

Starting which version of SQL Server, trace flag 2371 (for changing auto statistics threshold) is enabled by default?

What commands can be used to update statistics manually on a specific object or an entire database?

What enhancements were made to DBCC CHECKDB in SQL Server 2016?

Which feature can be enabled to reduce the impact of large file size autogrowth?