Module 6

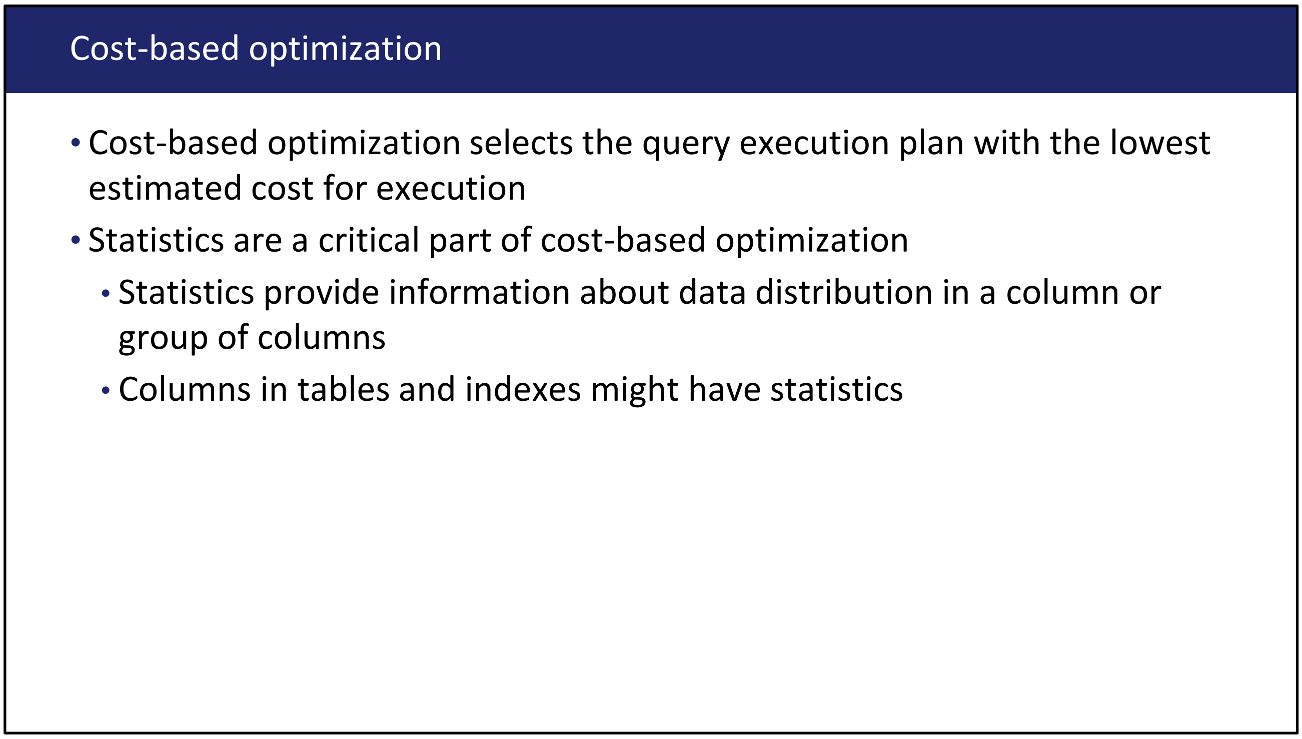
Statistics and indexes

Module Overview

- Statistics and cardinality estimation
- Indexes
- Columnstore indexes

Lesson: Statistics and cardinality estimation

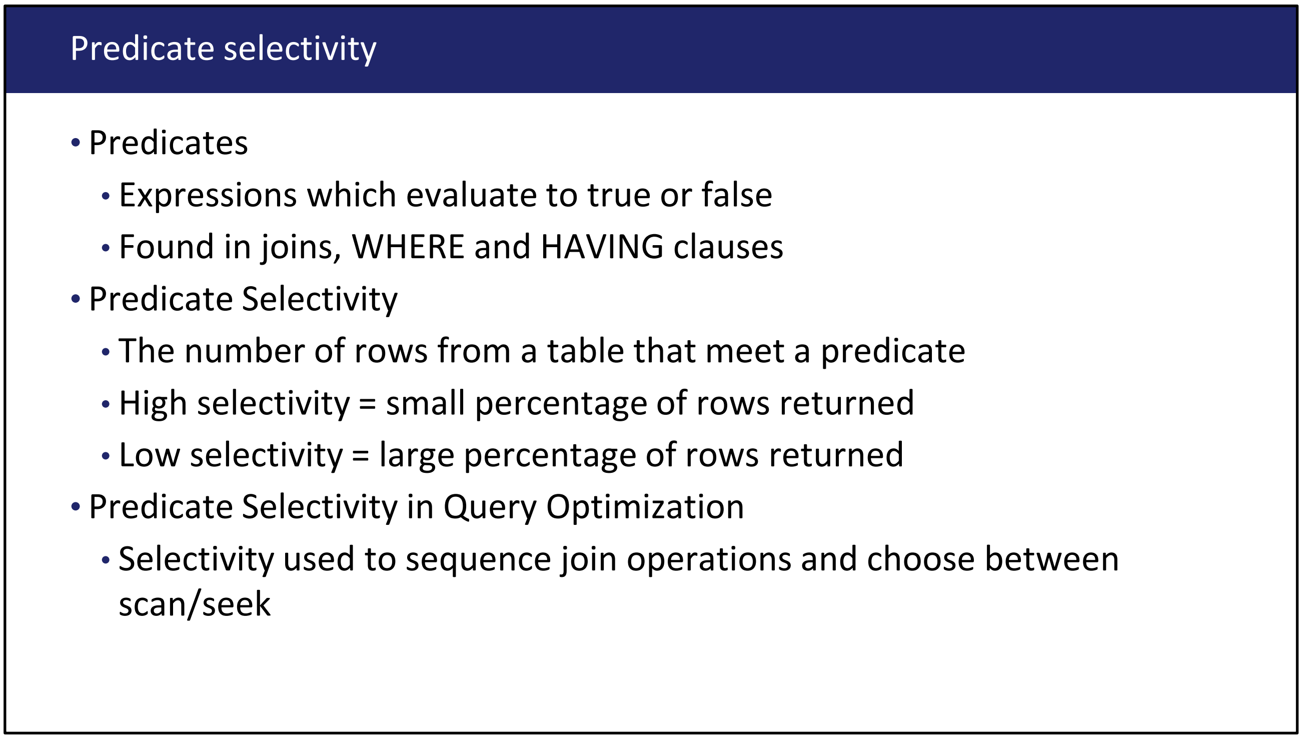
- Cost-based optimization
- Predicate selectivity
- Inspecting statistics
- Cardinality estimation
- Optimizer fixes
- Optimizer enhancements
- Controlling CE, OF and OE
- Creating statistics
- Updating statistics
- Filtered statistics



Cost-based optimization

- Cost-based optimization selects the query execution plan with the lowest estimated cost for execution
- Statistics are a critical part of cost-based optimization
  - Statistics provide information about data distribution in a column or group of columns
  - Columns in tables and indexes might have statistics

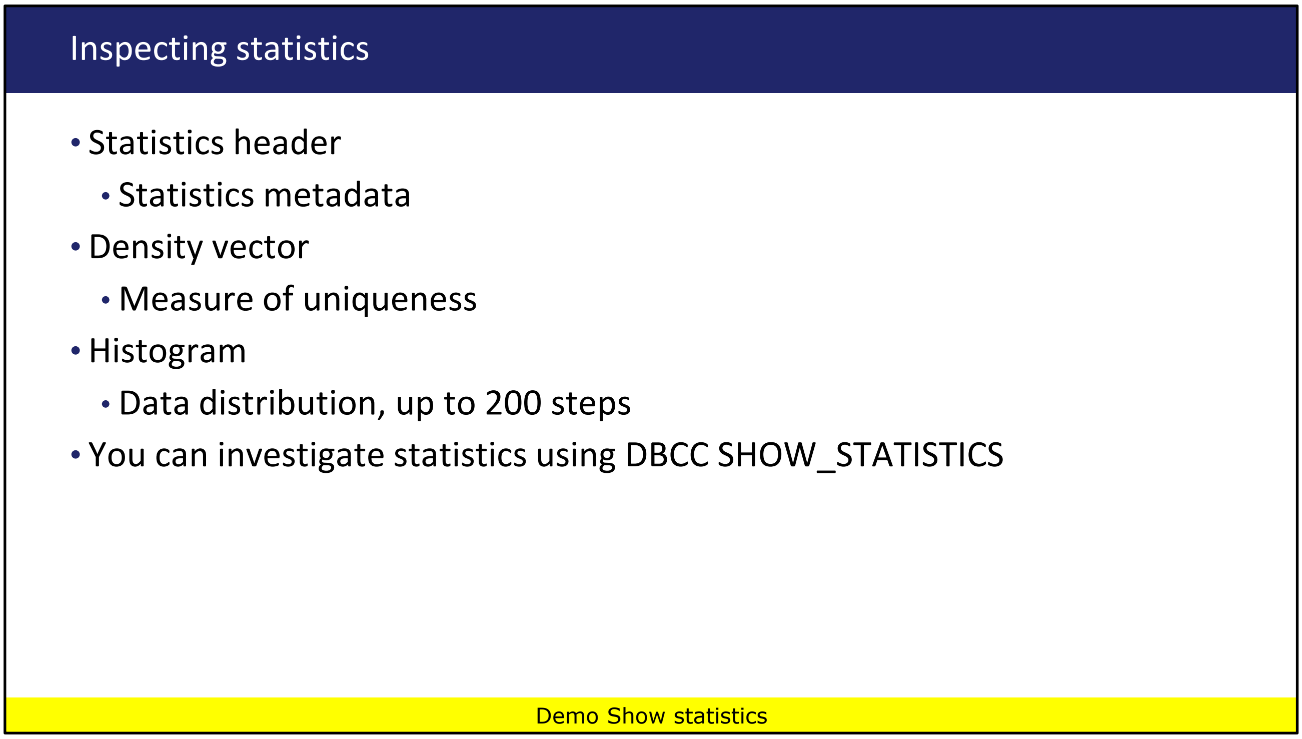
What Is a Cost-Based Optimizer?
https://www.brentozar.com/archive/2021/09/what-is-a-cost-based-optimizer/



Predicate selectivity

- Predicates
  - Expressions which evaluate to true or false
  - Found in joins, WHERE and HAVING clauses
- Predicate Selectivity
  - The number of rows from a table that meet a predicate
  - High selectivity = small percentage of rows returned
  - Low selectivity = large percentage of rows returned
- Predicate Selectivity in Query Optimization
  - Selectivity used to sequence join operations and choose between scan/seek

Query Tuning Fundamentals: Density, Predicates, Selectivity, and Cardinality
https://sqlserverdbknowledge.wordpress.com/2014/02/24/query-tuning-fundamentals-density-predicates-selectivity-and-cardinality/



Inspecting statistics

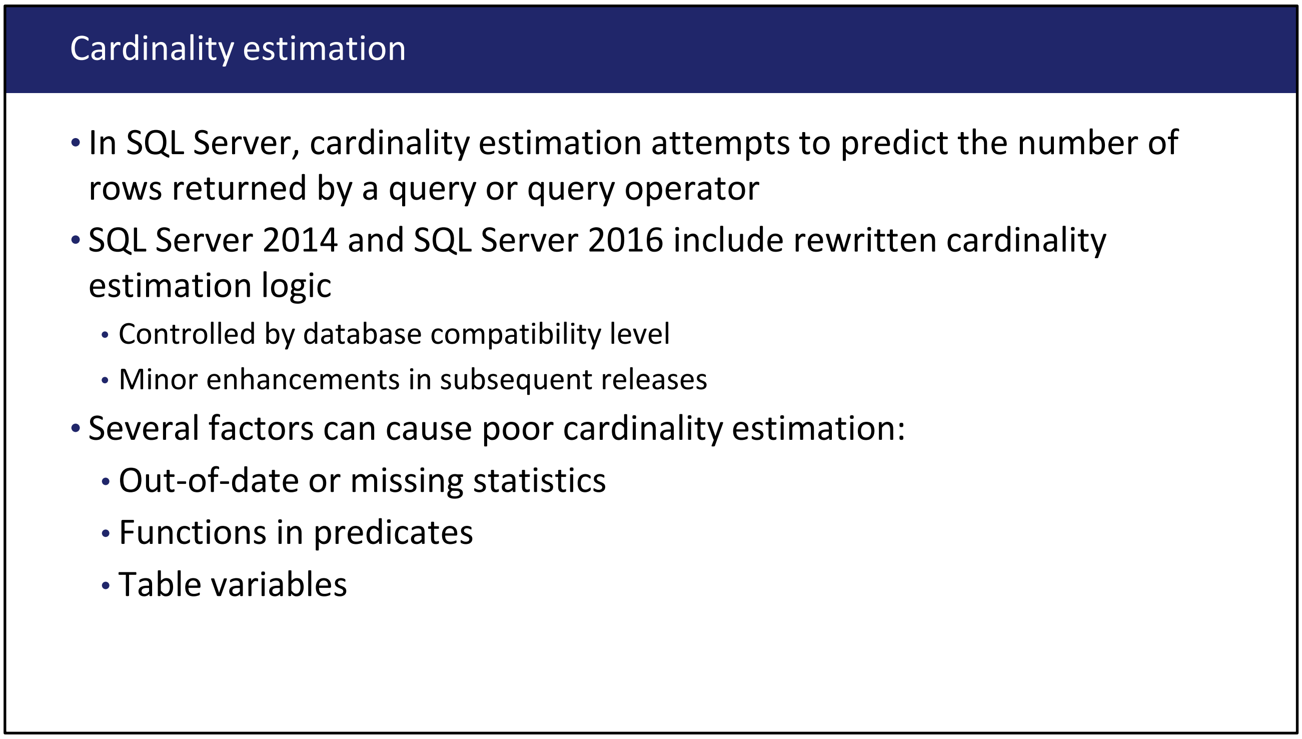
- Statistics header
  - Statistics metadata
- Density vector
  - Measure of uniqueness
- Histogram
  - Data distribution, up to 200 steps
- You can investigate statistics using DBCC SHOW_STATISTICS

Demo Show statistics

Statistics
https://docs.microsoft.com/en-us/sql/relational-databases/statistics/statistics

What are SQL Server Statistics and Where are they Stored?
https://www.sqlnethub.com/blog/what-are-sql-server-statistics-and-where-are-they-stored/

DBCC SHOW_STATISTICS (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/database-console-commands/dbcc-show-statistics-transact-sql



Cardinality estimation

- In SQL Server, cardinality estimation attempts to predict the number of rows returned by a query or query operator
- SQL Server 2014 and SQL Server 2016 include rewritten cardinality estimation logic
  - Controlled by database compatibility level
  - Minor enhancements in subsequent releases
- Several factors can cause poor cardinality estimation:
  - Out-of-date or missing statistics
  - Functions in predicates
  - Table variables

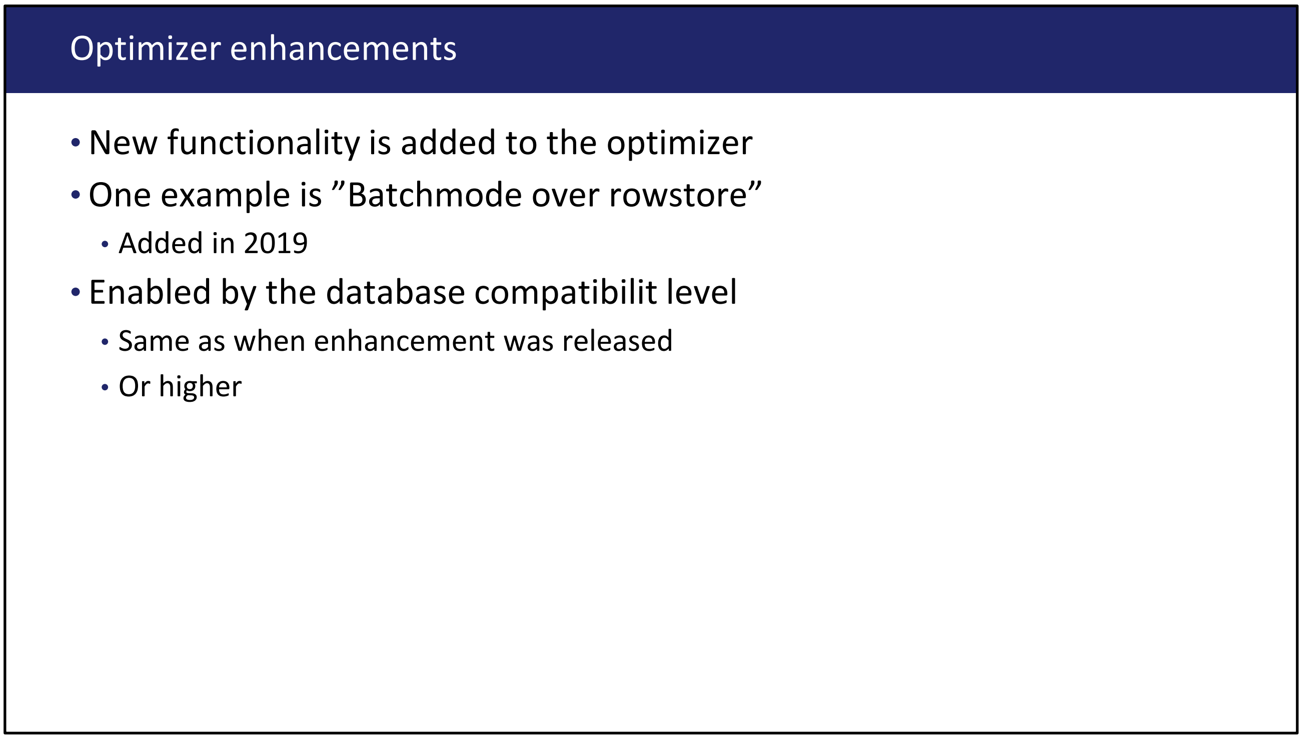
Cardinality Estimation (SQL Server)
https://docs.microsoft.com/en-us/sql/relational-databases/performance/cardinality-estimation-sql-server

Compatibility Levels and Cardinality Estimation Primer
https://sqlperformance.com/2019/01/sql-performance/compatibility-levels-and-cardinality-estimation-primer

Optimizer fixes

- A weakness in the optimizer can be fixed
- Prior to 2016 we had to opt in:
  - Using the trace flag for that particular fix
  - Or 4199 whcih enables all fixes
- As of 2016 the fixes of RTM are enabled
  - Assuming 2016 compatibility level
  - Patching SQL server will not add fixed
  - Unless you use trace flag 4199

Enabling SQL Server Optimizer Hotfixes
https://sqlrus.com/2020/07/understanding-how-to-enable-sql-server-optimizer-hotfixes/



Optimizer enhancements

- New functionality is added to the optimizer
- One example is ”Batchmode over rowstore”
  - Added in 2019
- Enabled by the database compatibilit level
  - Same as when enhancement was released
  - Or higher

Intelligent query processing in SQL databases
https://docs.microsoft.com/en-us/sql/relational-databases/performance/intelligent-query-processing

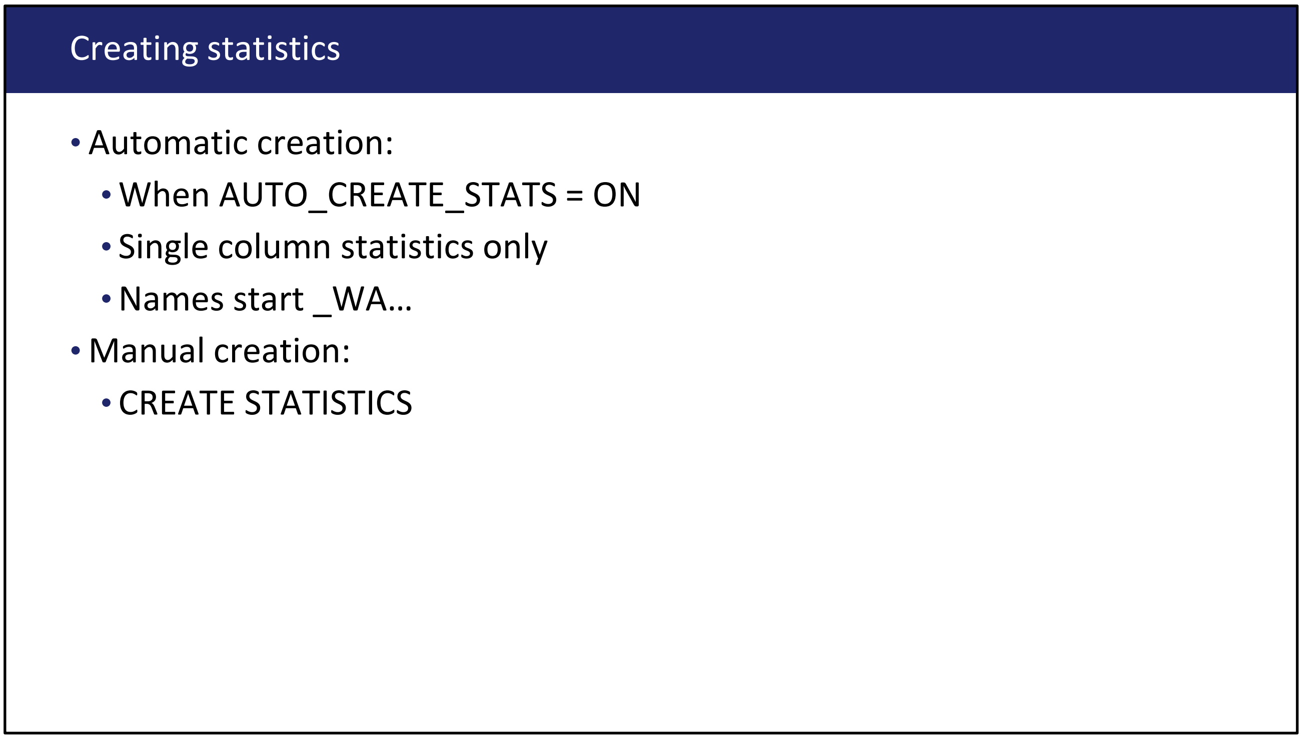
SQL SERVER 2022 Features for Performance Optimization
https://blog.sqlauthority.com/2021/11/08/sql-server-2022-features-for-performance-optimization/

Controlling CE, OF and OE

Setting	Type	CE	OF	OE
Db compat level	DB	Y	Y	Y
QUERY_OPTIMIZER_COMPATIBILITY_LEVEL_xxx	H	N	Y	Y
QUERY_OPTIMIZER_HOTFIXES	DB	N	Y	N
ENABLE_QUERY_OPTIMIZER_HOTFIXES	H	N	Y	N
FORCE_LEGACY_CARDINALTY_ESTIMATION	H	Y	N	N
LEGACY_CARDINALITY_ESTIMATION	DB	Y	N	N
9481 (force legacy CE)	TF	Y	N	N
2312 (force current CE with db compat 2012)	TF	Y	N	N

DB: Database setting
H: Hint
TF: Trace flag, can also be turned on at query level with QUERYTRACEFLAG hint

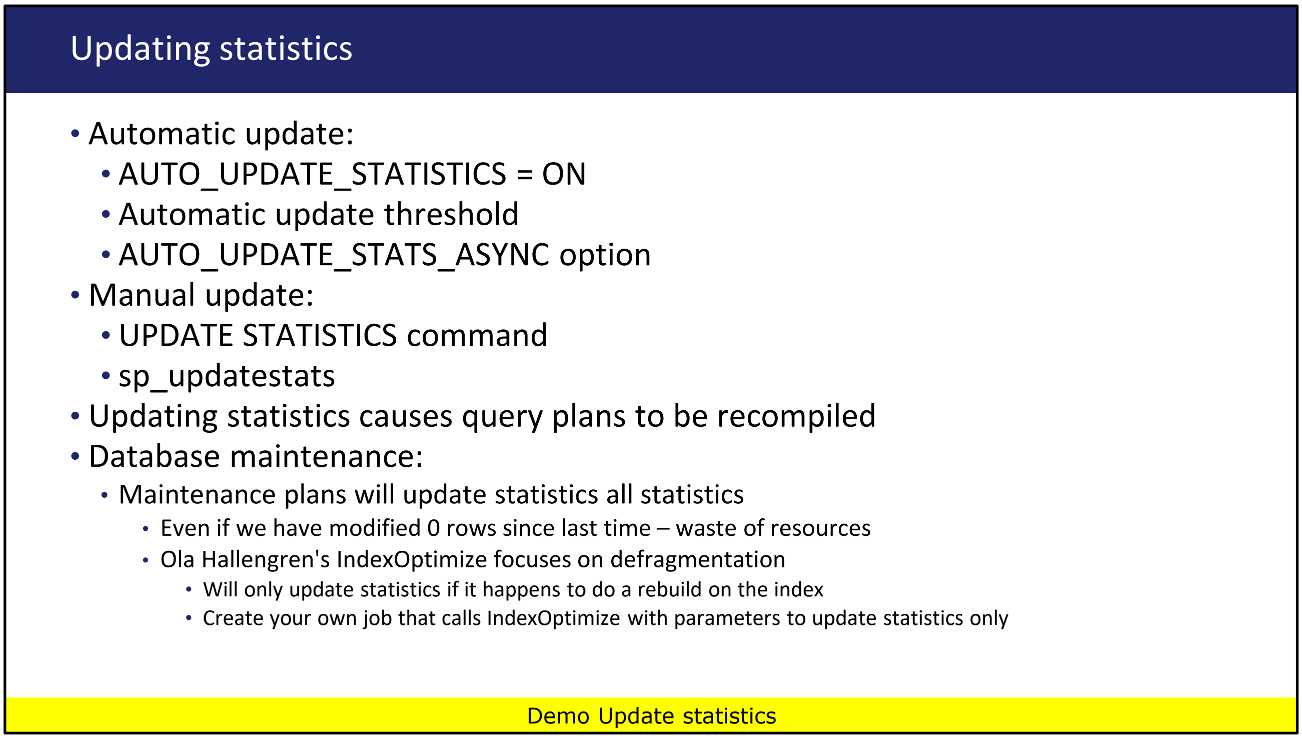
Demo CE and optimizer fixes



Creating statistics

- Automatic creation:
  - When AUTO_CREATE_STATS = ON
  - Single column statistics only
  - Names start _WA…
- Manual creation:
  - CREATE STATISTICS

Create Statistics
https://docs.microsoft.com/en-us/sql/relational-databases/statistics/create-statistics



Updating statistics

- Automatic update:
  - AUTO_UPDATE_STATISTICS = ON
  - Automatic update threshold
  - AUTO_UPDATE_STATS_ASYNC option
- Manual update:
  - UPDATE STATISTICS command
  - sp_updatestats
- Updating statistics causes query plans to be recompiled
- Database maintenance:
  - Maintenance plans will update statistics all statistics
    - Even if we have modified 0 rows since last time – waste of resources
    - Ola Hallengren's IndexOptimize focuses on defragmentation
      - Will only update statistics if it happens to do a rebuild on the index
      - Create your own job that calls IndexOptimize with parameters to update statistics only

Demo Update statistics

SQL Server Statistics and how to perform Update Statistics in SQL
https://www.sqlshack.com/sql-server-statistics-and-how-to-perform-update-statistics-in-sql/

UPDATE STATISTICS (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/statements/update-statistics-transact-sql

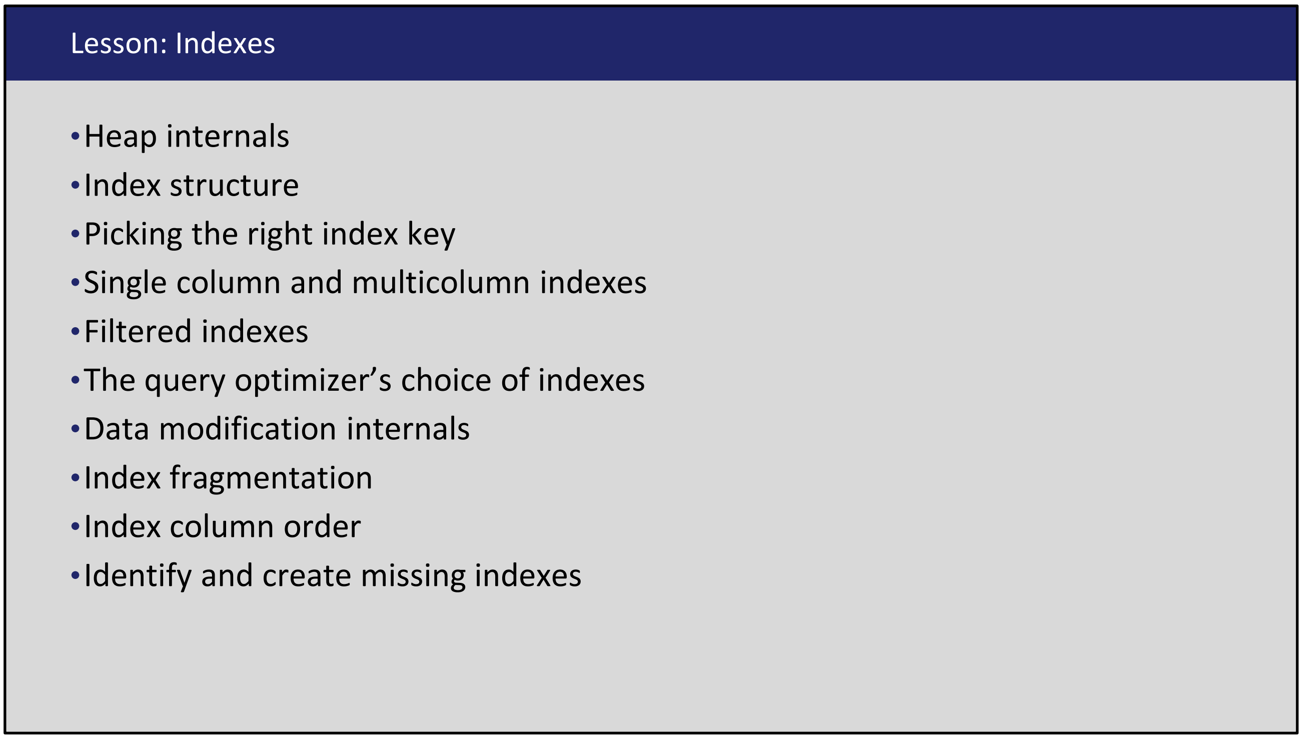
Update Statistics
https://docs.microsoft.com/en-us/sql/relational-databases/statistics/update-statistics

Filtered statistics

- CREATE STATISTICS using a WHERE clause
- Some limitations:
  - AUTO_UPDATE_STATISTICS threshold is based on all rows in table
    - Not the rows that the filter specifies
  - Limited to simple comparison logic
  - Cannot be created on all data/column types
  - Cannot be created on indexed views
- Histogram has finer resolution
  - When you combine several statistics

Filtered Statistics Follow-up
https://www.brentozar.com/archive/2016/12/filtered-statistics-follow/

CREATE STATISTICS (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/statements/create-statistics-transact-sql

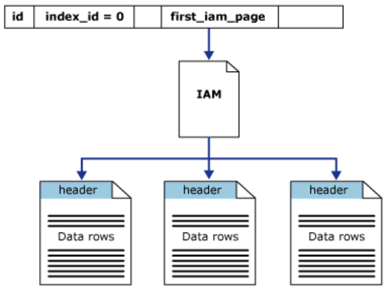


Lesson: Indexes

- Heap internals
- Index structure
- Picking the right index key
- Single column and multicolumn indexes
- Filtered indexes
- The query optimizer’s choice of indexes
- Data modification internals
- Index fragmentation
- Index column order
- Identify and create missing indexes

Heap internals

- A table without a clustered index is a heap
- IAM pages contain pointers to extents containing heap data
  - The IAM is the only link between pages in a heap; row data is unordered
- Heap rows are identified by a row identifier (RID)
  - Nonclustered indexes on a heap contain RID pointers



Heaps (Tables without Clustered Indexes)
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/heaps-tables-without-clustered-indexes

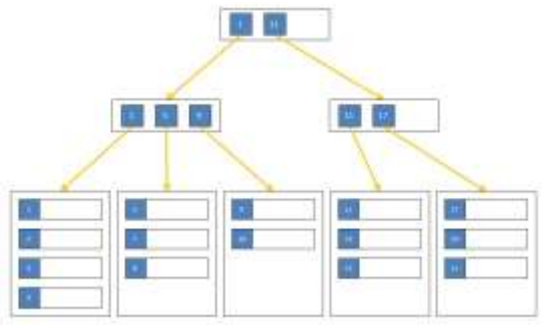
Tables Without Clustered Indexes
https://www.brentozar.com/blitz/heaps-tables-without-primary-key-clustered-index/

Heaps in SQL Server: Part 1 The Basics
https://www.red-gate.com/simple-talk/databases/sql-server/t-sql-programming-sql-server/heaps-in-sql-server-part-1-the-basics/

Clustered Index vs. Heap in SQL Server
https://www.sqlshack.com/clustered-index-vs-heap/

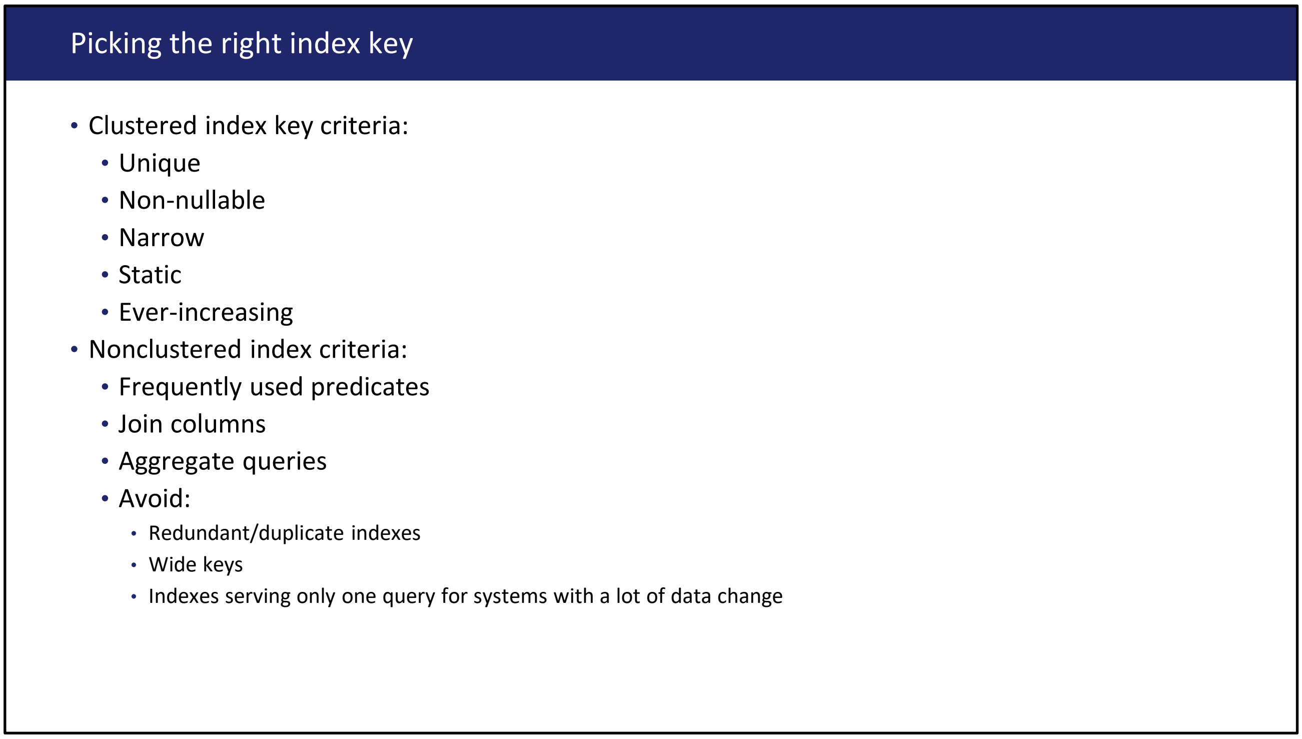
Index structure

- B-trees:
  - Self-balancing tree data structure
  - One root node; many non-leaf nodes; many leaf nodes
  - Clustered and nonclustered indexes are b-trees
- Index key:
  - Defines the order of data in an index
- Clustered index:
  - Leaf note ***is*** the data, contains all columns
  - Higher levels has index key
  - Index order = table data order
- Nonclustered index:
  - All nodes contain index pages
- There are other index types than B-Trees:
  - XML, full-text, hash, columnstore and spatial indexes



Indexes
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/indexes

Clustered and nonclustered indexes described
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/clustered-and-nonclustered-indexes-described



Picking the right index key

- Clustered index key criteria:
  - Unique
  - Non-nullable
  - Narrow
  - Static
  - Ever-increasing
- Nonclustered index criteria:
  - Frequently used predicates
  - Join columns
  - Aggregate queries
  - Avoid:
    - Redundant/duplicate indexes
    - Wide keys
    - Indexes serving only one query for systems with a lot of data change

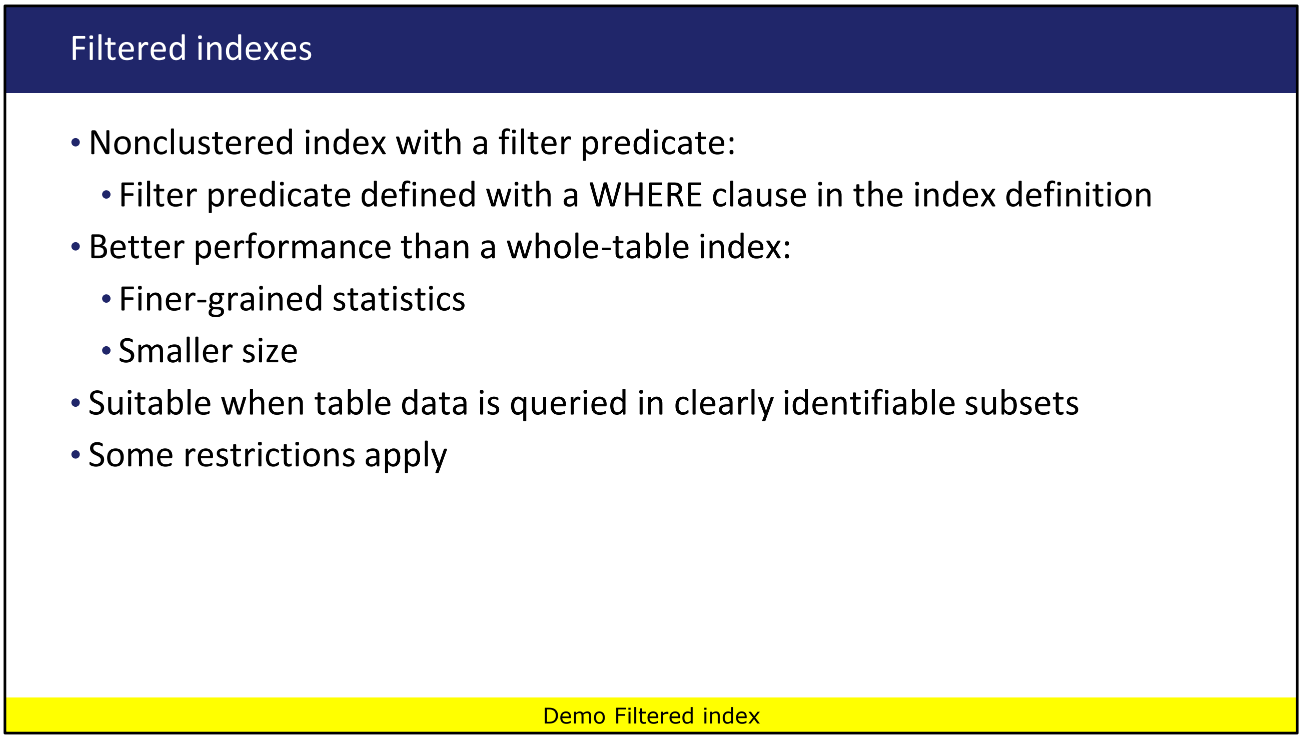
Designing effective SQL Server clustered indexes
https://www.sqlshack.com/designing-effective-sql-server-clustered-indexes/

Clustered and Non Clustered Index: 7 Top Points Explained
https://codingsight.com/clustered-and-non-clustered-index-7-top-points-explained/

Single column and multicolumn indexes

- Single column index:
  - Each predicate column has its own index
  - Less performance gain, but more reusable
- Multicolumn index:
  - All predicate columns are included in the key of the index
  - Greatest performance gain, but limited reusability

Demo Index several columns



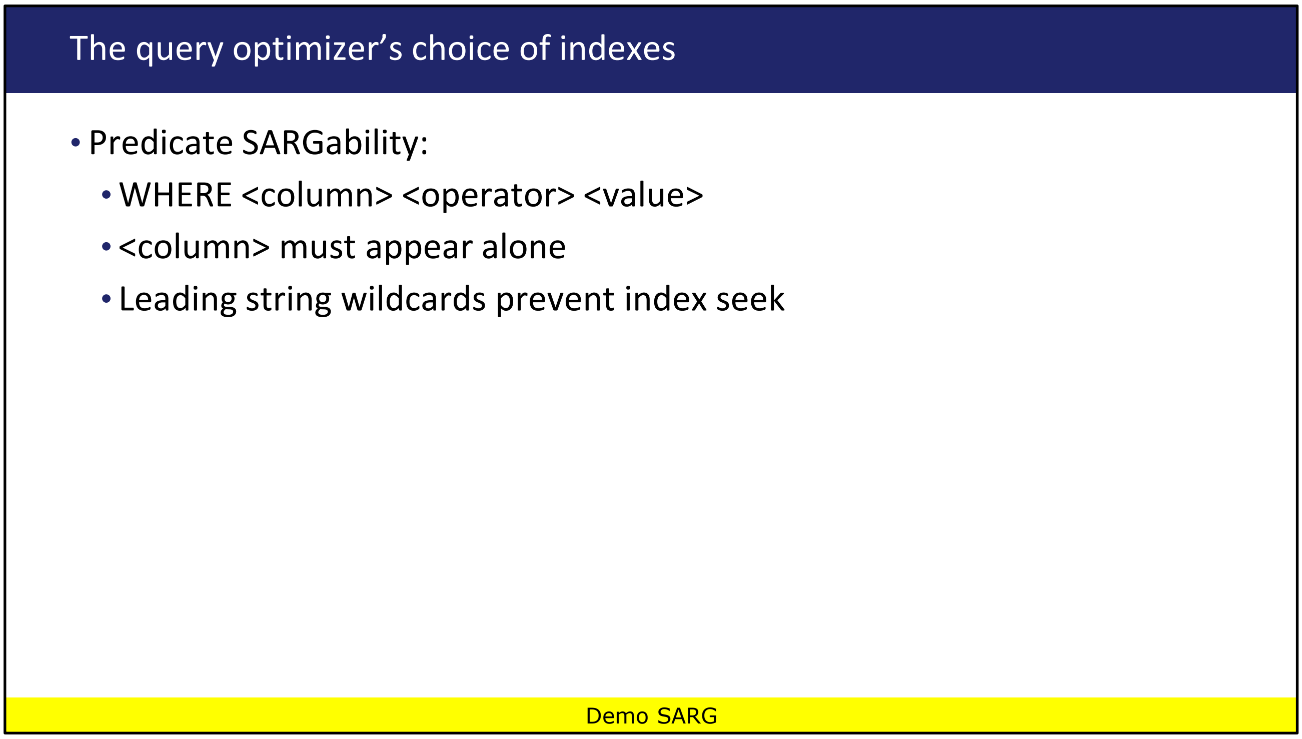
Filtered indexes

- Nonclustered index with a filter predicate:
  - Filter predicate defined with a WHERE clause in the index definition
- Better performance than a whole-table index:
  - Finer-grained statistics
  - Smaller size
- Suitable when table data is queried in clearly identifiable subsets
- Some restrictions apply

Demo Filtered index

Create filtered indexes
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/create-filtered-indexes

What You Can (and Can’t) Do With Filtered Indexes
https://www.brentozar.com/archive/2013/11/what-you-can-and-cant-do-with-filtered-indexes/



The query optimizer's choice of indexes

- Predicate SARGability:
  - WHERE <column> <operator> <value>
  - <column> must appear alone
  - Leading string wildcards prevent index seek

Demo SARG

Sargable
https://en.wikipedia.org/wiki/Sargable

The Two Ways to Fix Non-Sargable Queries
https://www.brentozar.com/archive/2019/12/the-two-ways-to-fix-non-sargable-queries/

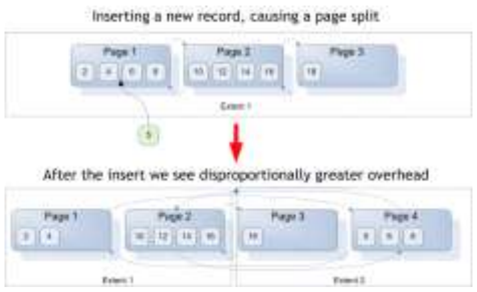
How to use sargable expressions in T-SQL queries; performance advantages and examples
https://www.sqlshack.com/how-to-use-sargable-expressions-in-t-sql-queries-performance-advantages-and-examples/

Data modification internals

- Delete:
  - Key reference removed from leaf-level page
  - A page with no references is removed from the b-tree
- Insert:
  - New key reference added to either:
    - Existing free space on a page
    - New page
- Update:
  - A delete followed by an insert

Index fragmentation

- External fragmentation
  - Linked list jumps back and forth in the database file
  - Less relevant unless you have single magnetic disks
- Internal fragmentation
  - Pages are less than 100% full
  - Watch out for
    - Rebuilding and re-applying a lower fill-factor
    - Rebuilding and ***not*** re-applying a lower fill-factor


Inserting a new record, causing a page split
Page 1
Page 2
Page 3
Extent 1
After the insert we see disproportionally greater overhead
Page 1
Page 2
Page 3
Page 4
Extent 1
Extent 2

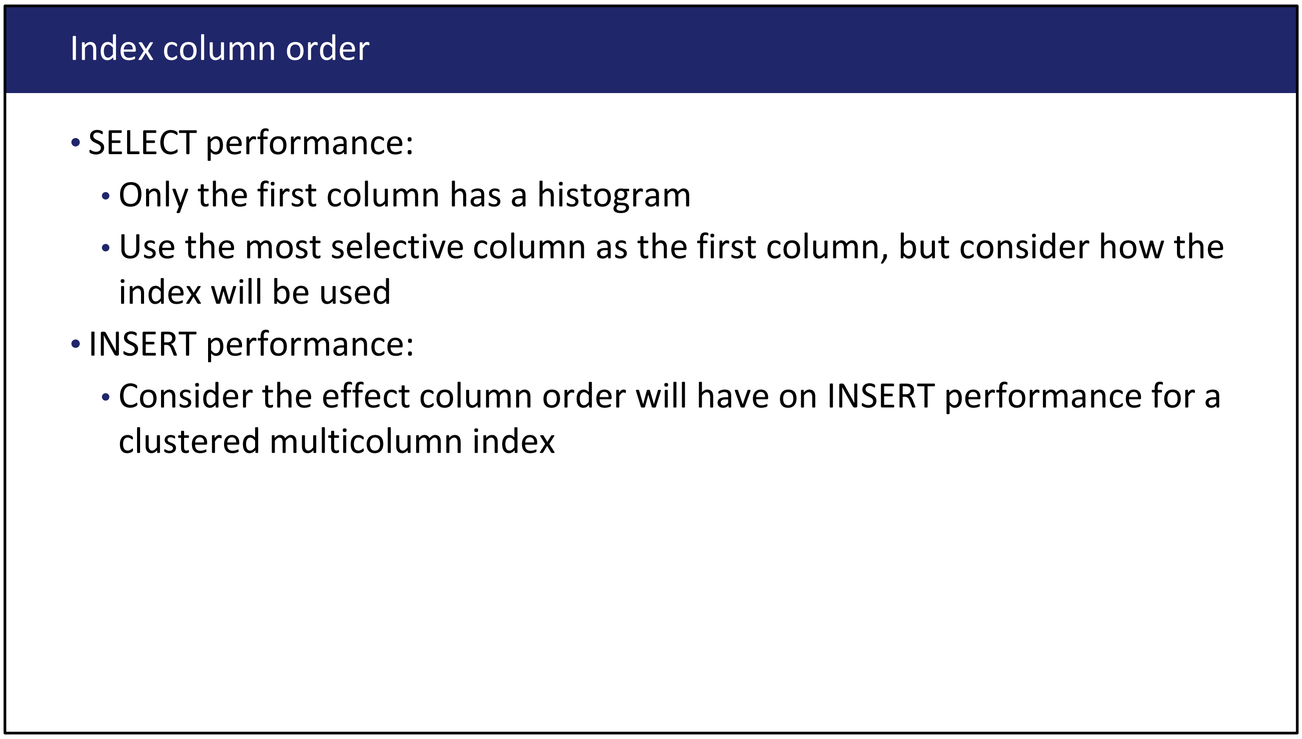
Demo Fragmentation

Optimize index maintenance to improve query performance and reduce resource consumption
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/reorganize-and-rebuild-indexes

Index fragmentation revisited
https://sqlblog.karaszi.com/index-fragmentation-revisited/

Stop Worrying About SQL Server Fragmentation
https://www.brentozar.com/archive/2012/08/sql-server-index-fragmentation/

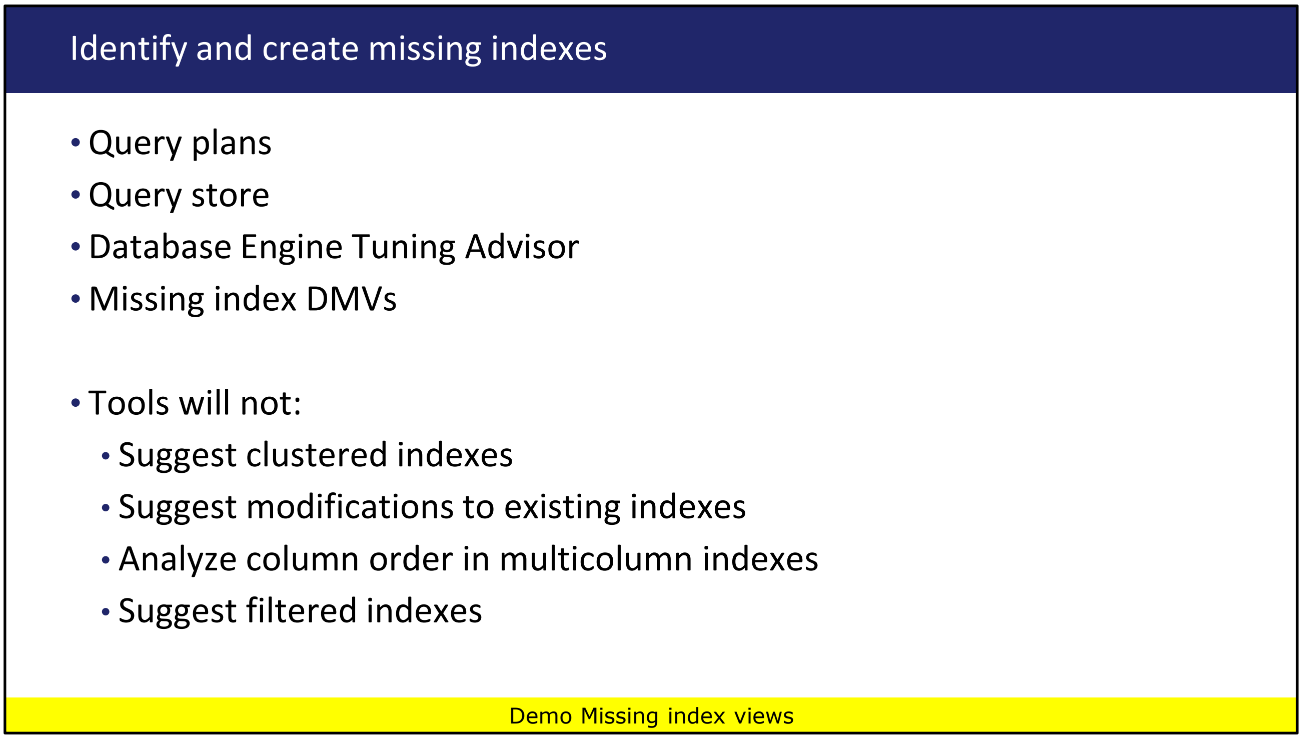
Rebuild all fragmented heaps
https://karaszi.com/rebuild-all-fragmented-heaps



Index column order

- SELECT performance:
  - Only the first column has a histogram
  - Use the most selective column as the first column, but consider how the index will be used
- INSERT performance:
  - Consider the effect column order will have on INSERT performance for a clustered multicolumn index

How to Think Like the Engine: Index Column Order Matters a LOT.
https://www.brentozar.com/archive/2019/11/how-to-think-like-the-engine-index-column-order-matters-a-lot/



Identify and create missing indexes

- Query plans
- Query store
- Database Engine Tuning Advisor
- Missing index DMVs

- Tools will not:
  - Suggest clustered indexes
  - Suggest modifications to existing indexes
  - Analyze column order in multicolumn indexes
  - Suggest filtered indexes

Demo Missing index views

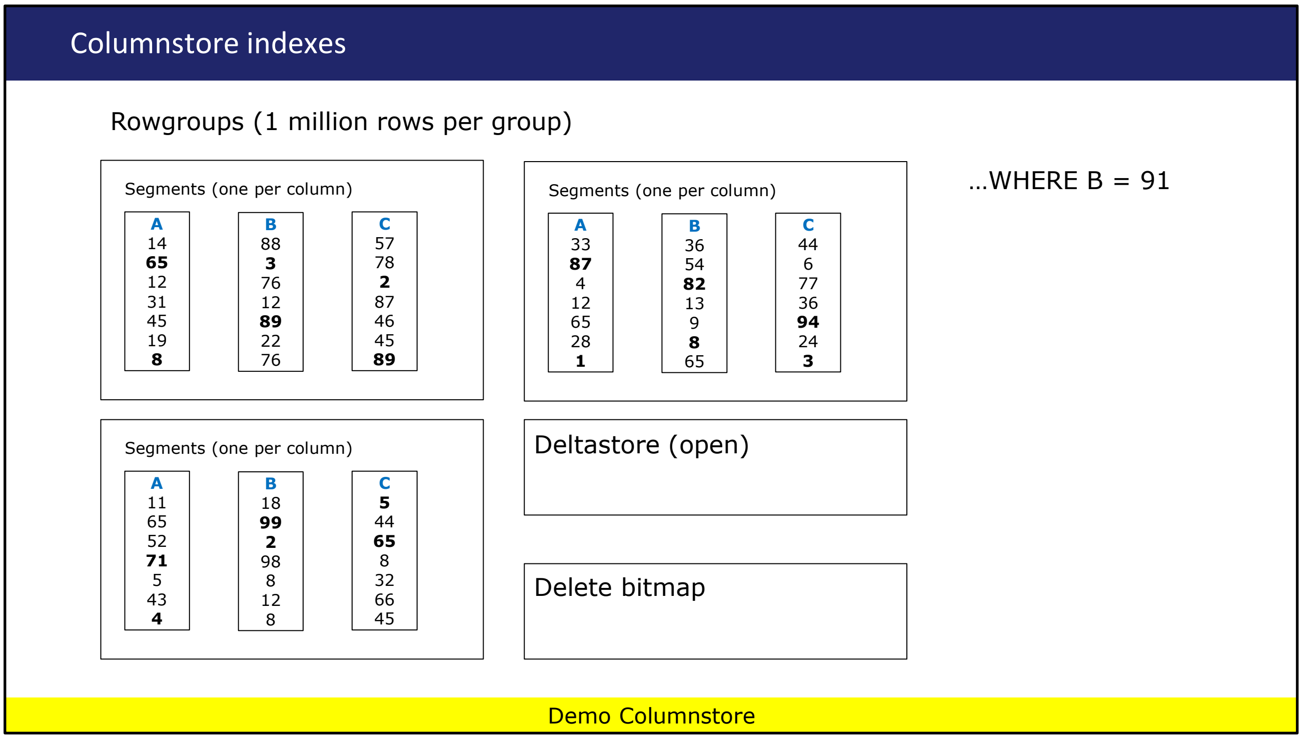
Index Related Dynamic Management Views and Functions (Transact-SQL)
https://docs.microsoft.com/en-us/sql/relational-databases/system-dynamic-management-views/index-related-dynamic-management-views-and-functions-transact-sql

Diagnosis: Index-a-phobia
https://www.brentozar.com/blitzindex/sp_blitzindex-missing-indexes/

Glenn Berry’s SQL Server Diagnostic Queries
https://glennsqlperformance.com/resources/

Lesson: Columnstore indexes

- Columnstore indexes
- Columnstore index features
- Columnstore index recommendations
- Maintaining Columnstore indexes



Columnstore indexes

Rowgroups (1 million rows per group)

Segments (one per column)

A	B	C
14	88	57
65	**3**	78
12	76	**2**
31	12	87
45	**89**	46
19	22	45
8	76	**89**

Segments (one per column)

A	B	C
33	36	44
87	54	6
4	**82**	77
12	13	36
65	9	**94**
28	**8**	24
1	65	**3**

Segments (one per column)

A	B	C
11	18	**5**
65	**99**	44
52	**2**	**65**
71	98	8
5	8	32
43	12	66
4	8	45

Deltastore (open)

Delete bitmap

…WHERE B = 91

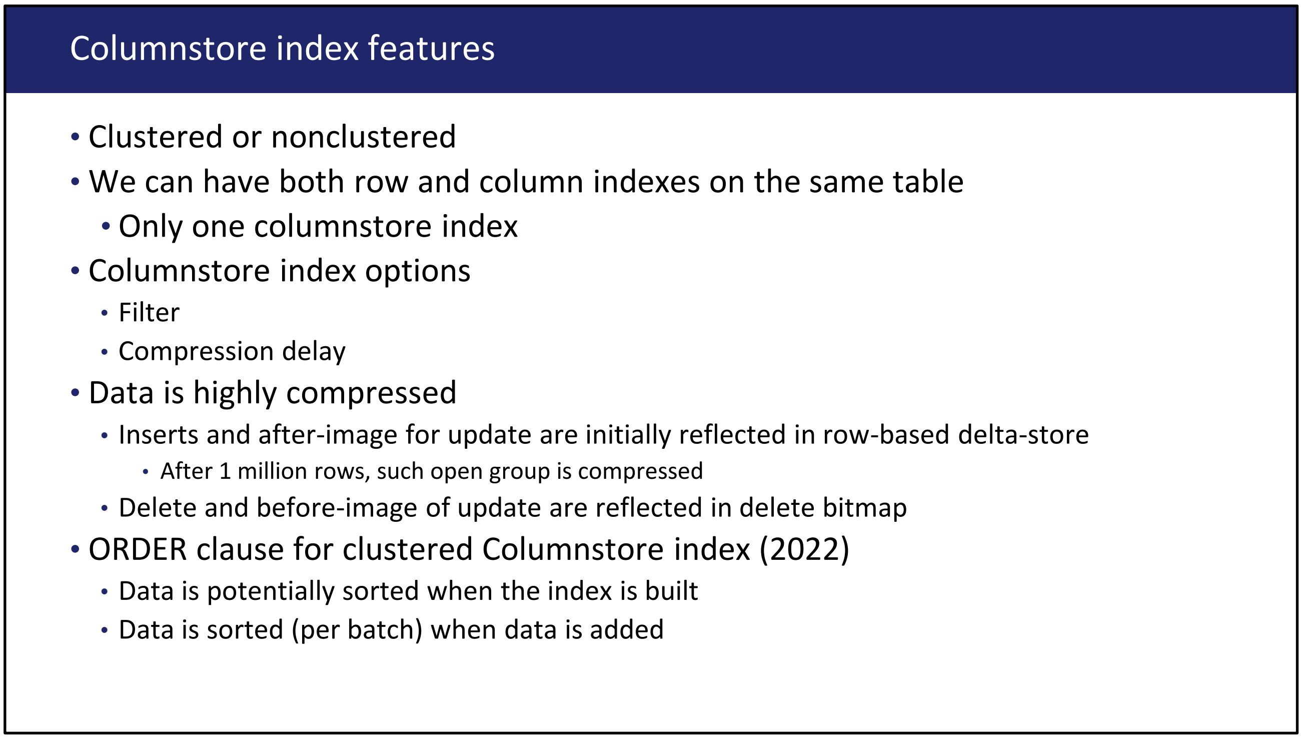
Demo Columnstore

Columnstore indexes: Overview
https://learn.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-overview

What are Columnstore Indexes?
https://www.red-gate.com/simple-talk/databases/sql-server/t-sql-programming-sql-server/what-are-columnstore-indexes/

Columnstore Index Scan
https://sqlserverfast.com/epr/columnstore-index-scan/

Niko Neugebauer, Columnstore
https://www.nikoport.com/columnstore/



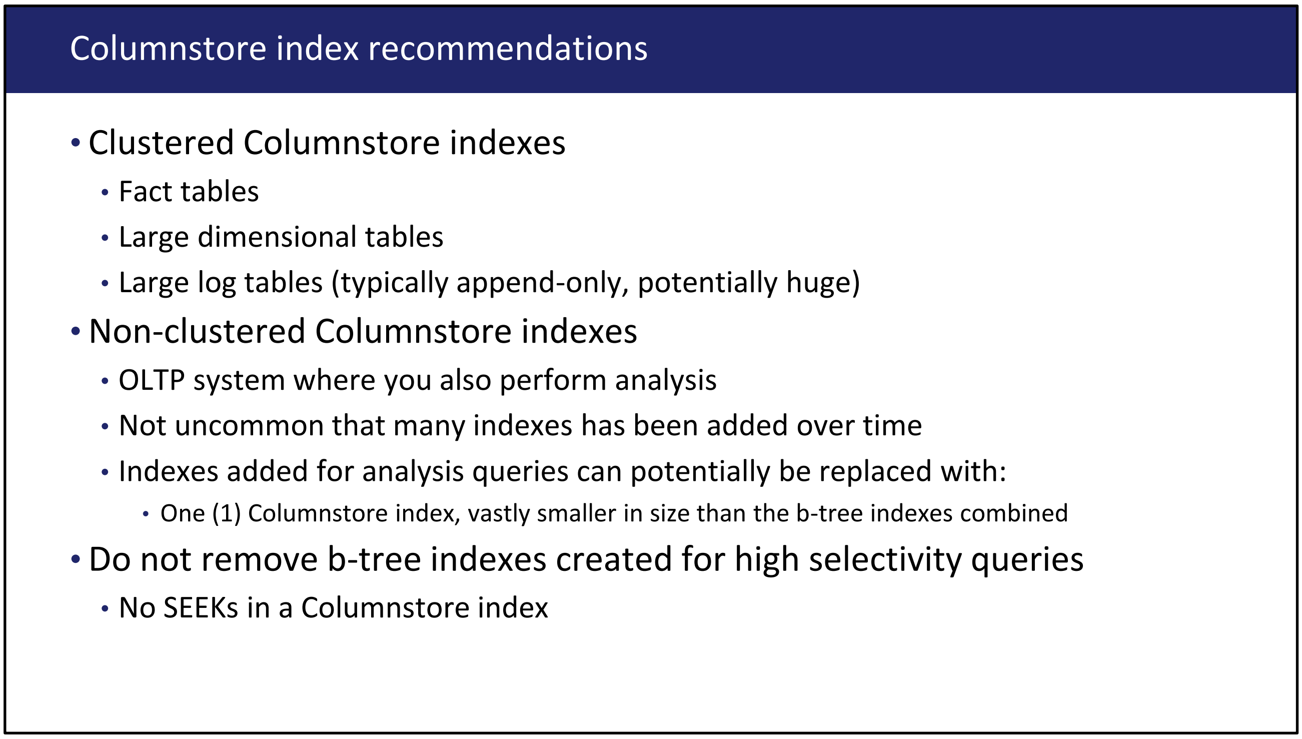
Columnstore index features

- Clustered or nonclustered
- We can have both row and column indexes on the same table
  - Only one columnstore index
- Columnstore index options
  - Filter
  - Compression delay
- Data is highly compressed
  - Inserts and after-image for update are initially reflected in row-based delta-store
    - After 1 million rows, such open group is compressed
  - Delete and before-image of update are reflected in delete bitmap
- ORDER clause for clustered Columnstore index (2022)
  - Data is potentially sorted when the index is built
  - Data is sorted (per batch) when data is added

CREATE COLUMNSTORE INDEX (Transact-SQL)
https://docs.microsoft.com/en-us/sql/t-sql/statements/create-columnstore-index-transact-sql

Columnstore indexes - what's new
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-what-s-new

Stairway to Columnstore Indexes
https://www.sqlservercentral.com/stairways/stairway-to-columnstore-indexes

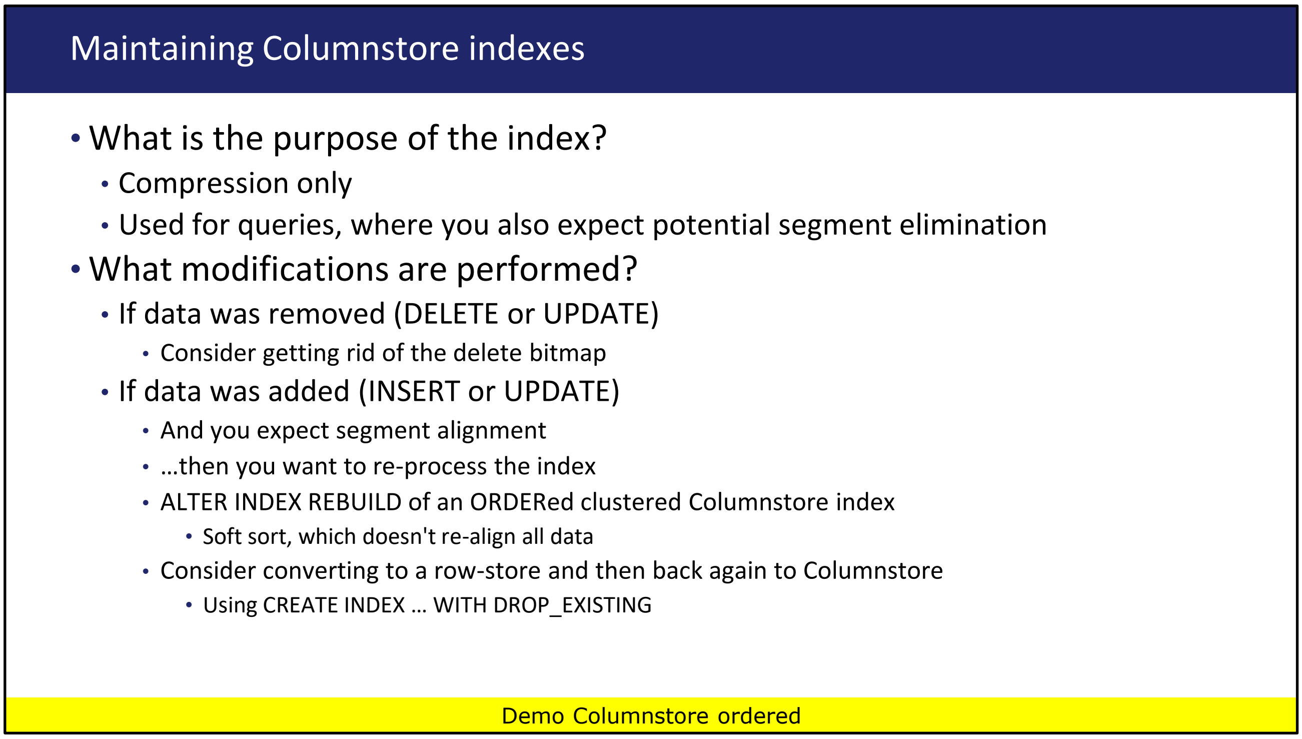


Columnstore index recommendations

- Clustered Columnstore indexes
  - Fact tables
  - Large dimensional tables
  - Large log tables (typically append-only, potentially huge)
- Non-clustered Columnstore indexes
  - OLTP system where you also perform analysis
  - Not uncommon that many indexes has been added over time
  - Indexes added for analysis queries can potentially be replaced with:
    - One (1) Columnstore index, vastly smaller in size than the b-tree indexes combined
- Do not remove b-tree indexes created for high selectivity queries
  - No SEEKs in a Columnstore index

Columnstore indexes - what's new
https://docs.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-what-s-new

Niko Neugebauer, Columnstore
https://www.nikoport.com/columnstore/



Maintaining Columnstore indexes

- What is the purpose of the index?
  - Compression only
  - Used for queries, where you also expect potential segment elimination
- What modifications are performed?
  - If data was removed (DELETE or UPDATE)
    - Consider getting rid of the delete bitmap
  - If data was added (INSERT or UPDATE)
    - And you expect segment alignment
    - …then you want to re-process the index
    - ALTER INDEX REBUILD of an ORDERed clustered Columnstore index
      - Soft sort, which doesn't re-align all data
    - Consider converting to a row-store and then back again to Columnstore
      - Using CREATE INDEX … WITH DROP_EXISTING

Demo Columnstore ordered

ORDER for columnstore index
https://sqlblog.karaszi.com/order-for-columnstore-index/

Performance tuning with ordered clustered columnstore index
https://learn.microsoft.com/en-us/azure/synapse-analytics/sql-data-warehouse/performance-tuning-ordered-cci

Columnstore indexes - Data loading guidance
https://learn.microsoft.com/en-us/sql/relational-databases/indexes/columnstore-indexes-data-loading-guidance

Lab 6: Statistics and indexes

- Ex 1: Improve a query, the easy way
- Ex 2: Improve a query, the traditional way
- Ex 3: Improve a query, by query re-write

Estimated Time: 30 minutes