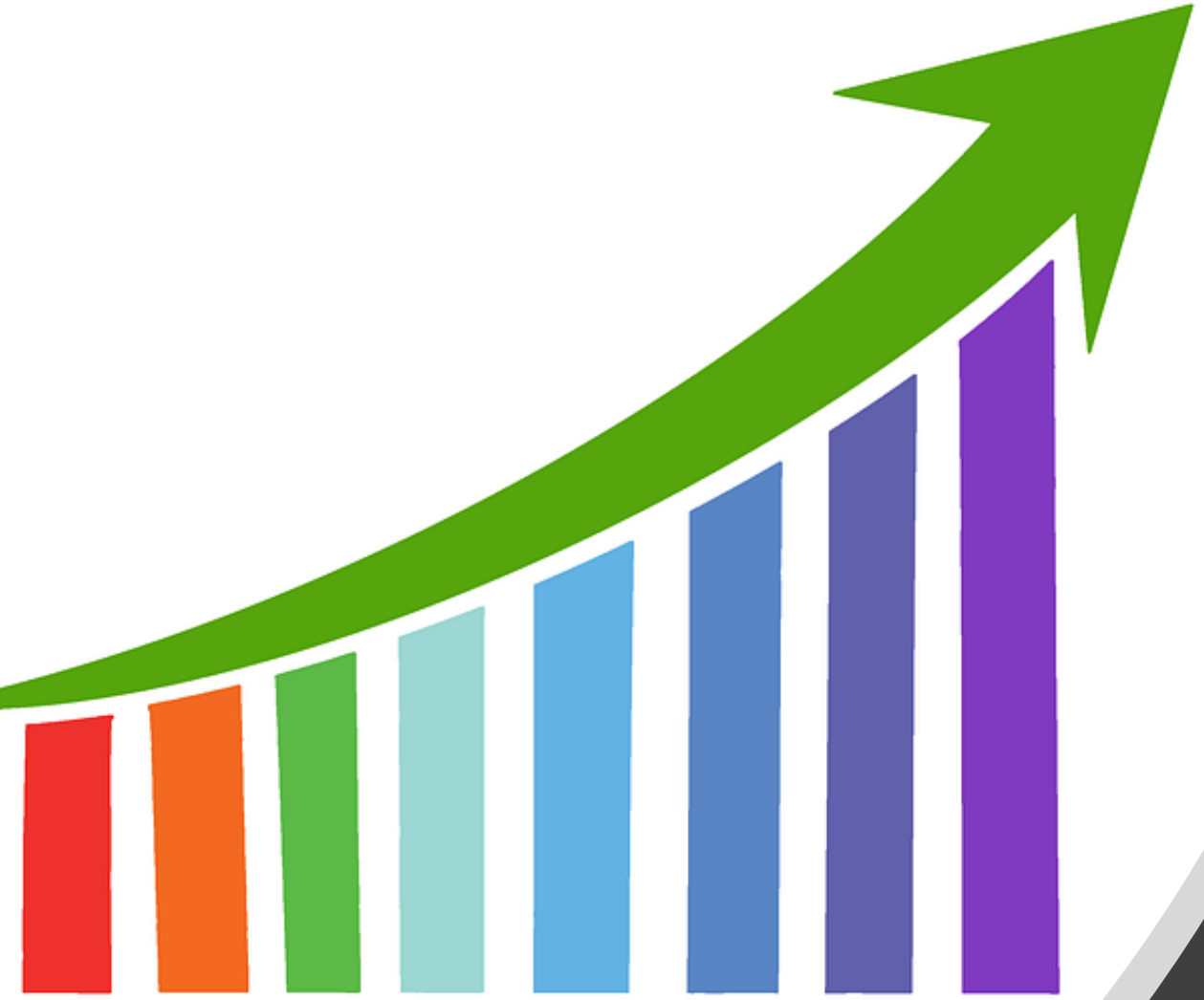


SQL Server Statistics

Who am I

- Steen Dybboe
- Database Developer
- Pragmatic BI
- Plays Golf
- No dogs


Længere ind i skoven

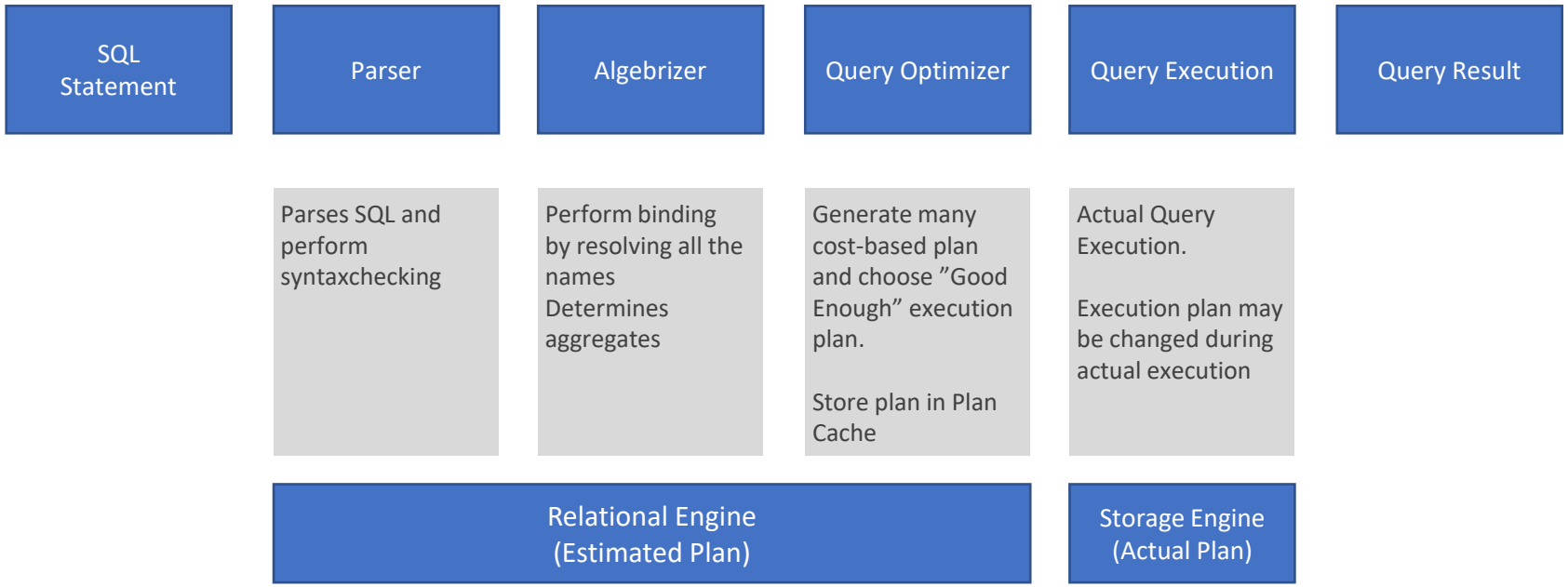
Statistics

- **Query Optimizer and Execution Plans**
- **Statistics**
  - **Anatomy of statistics**
  - **Statistics in action**
  - **Bad statistics**
- **Statistics maintenance**
- **Recommendations**

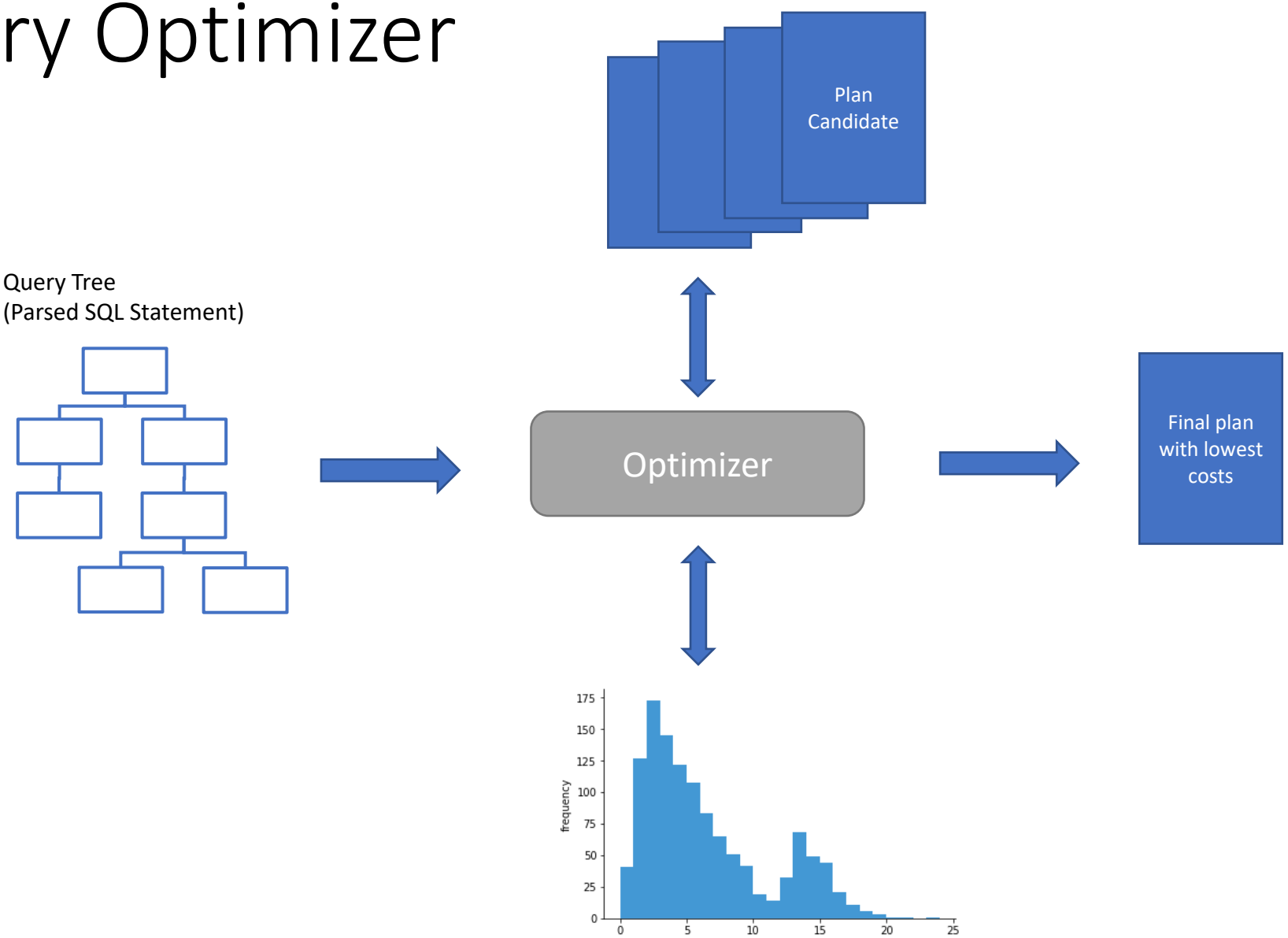
Query tuning isn’t just about using a single method but understanding how they all fit together

Query Optimizer

- Query Execution Flow

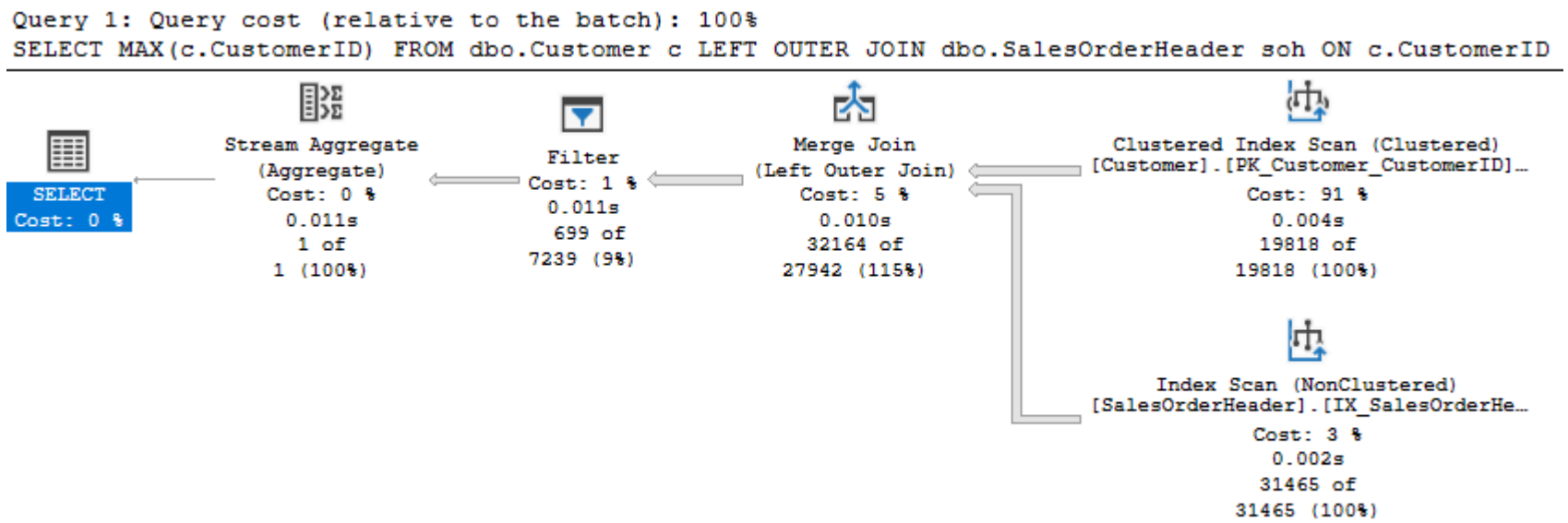


Query Optimizer



Execution plans

```sql
SELECT MAX(c.CustomerID)
FROM dbo.Customer c
LEFT OUTER JOIN dbo.SalesOrderHeader soh ON c.CustomerID = soh.CustomerID
WHERE soh.CustomerID IS NULL;
```



A Bad Plan is not the one which failed, but the one which succeeded at the Greatest Cost

~Anonymous DBA

Statistics – Cardinality Estimation

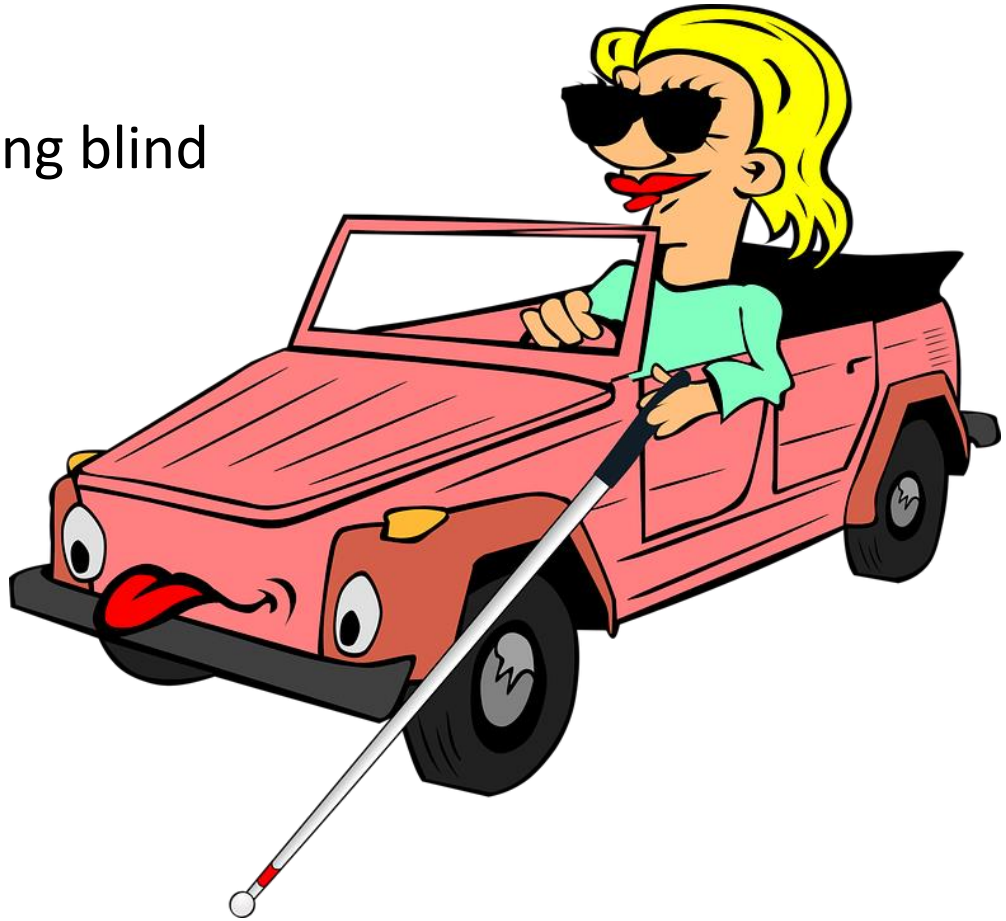
- SQL Server uses statistics to determine anticipated frequency and distribution of values within an index
  - This is referred to as ***Cardinality Estimation***
- Query Optimizer uses Cardinality to develop cost-based execution plans
  - Bad statistics results in bad plans !!
  - Having more accurate cardinality calculations means the optimizer will make better choices
  - How many rows needs to be fetched to Sort, Join, Group or Manipulate in your query (tasks, temp tables)
- Statistics assists in building the best plan
  - Plays no direct role in Query Execution

Statistics

- May be filtered
- Must be updated as often as “necessary”
  - “Ascending Date Key” pattern problem
    - When table holds LastModifiedTable column
    - Large chunk of data arrive
    - If you ask for newest time, this will not be part of Statistic → statistics returns “1 row only”
- Index rebuild may not be the reason for better performance
  - Indexes receive new Statistics AND then related execution plans are recompiled

Statistics – In Action

- The usefulness of an index is largely dependent on the statistics of the indexed columns
- Without statistics the Query Optimizer is driving blind

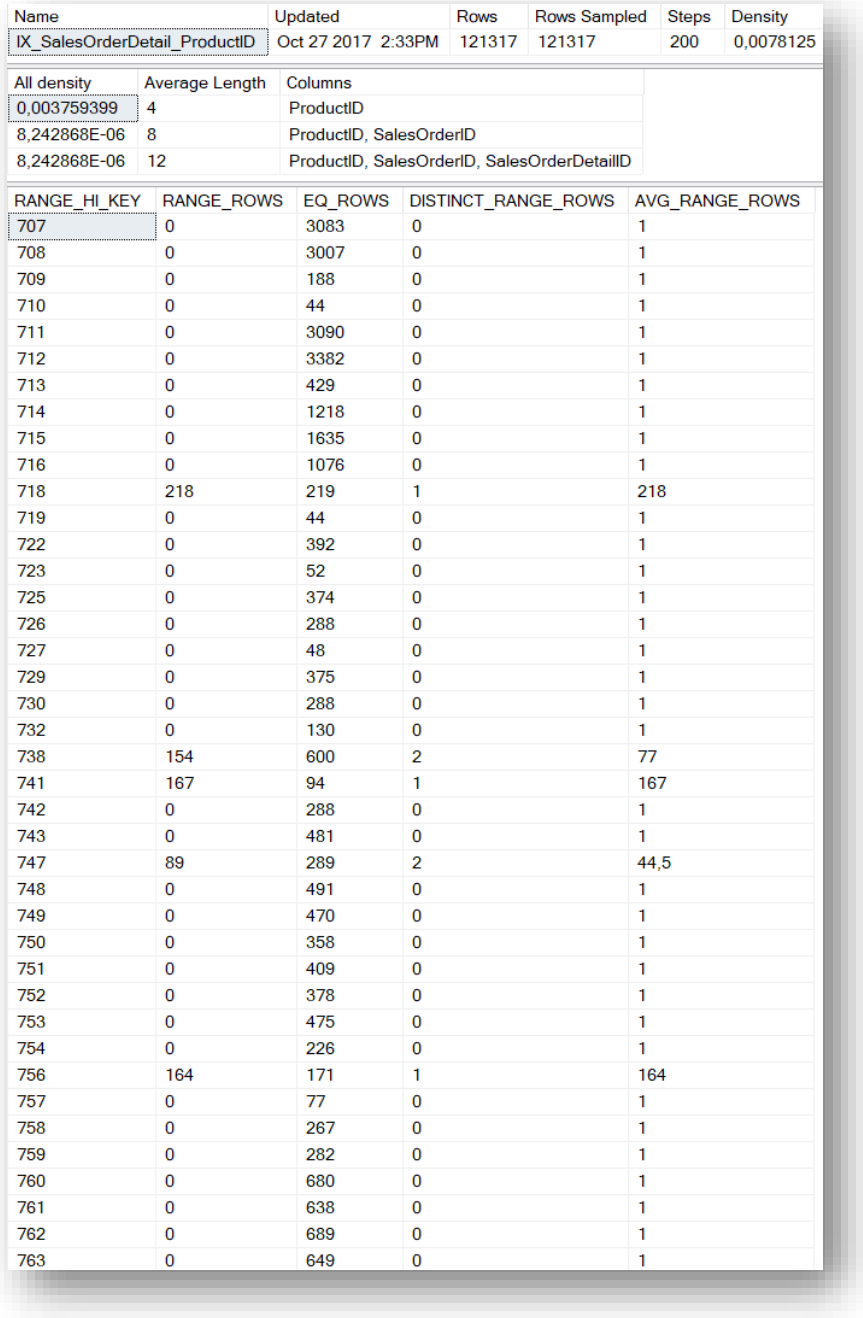


What are Statistics

- ***Small*** lightweight objects

- "Describes data in" = Types of statistics
  - An index (associated with)
    - Created automatically – "the statistical friend"
  - A Column
    - Usually created automatically
  - A Set of Column

- Contains
  - Information about data
  - Histogram, distribution
  - … And that's it

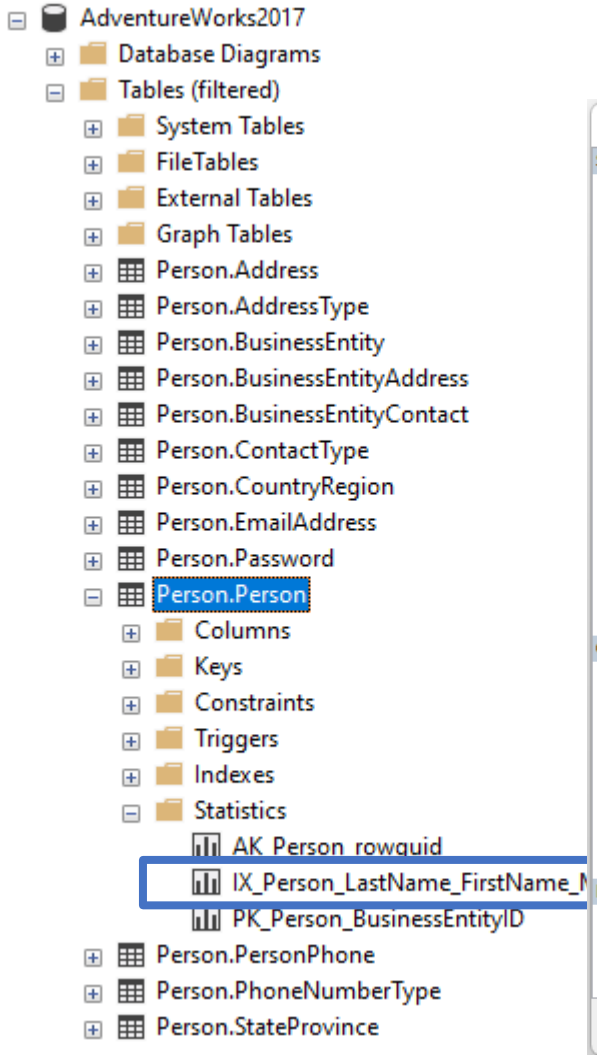


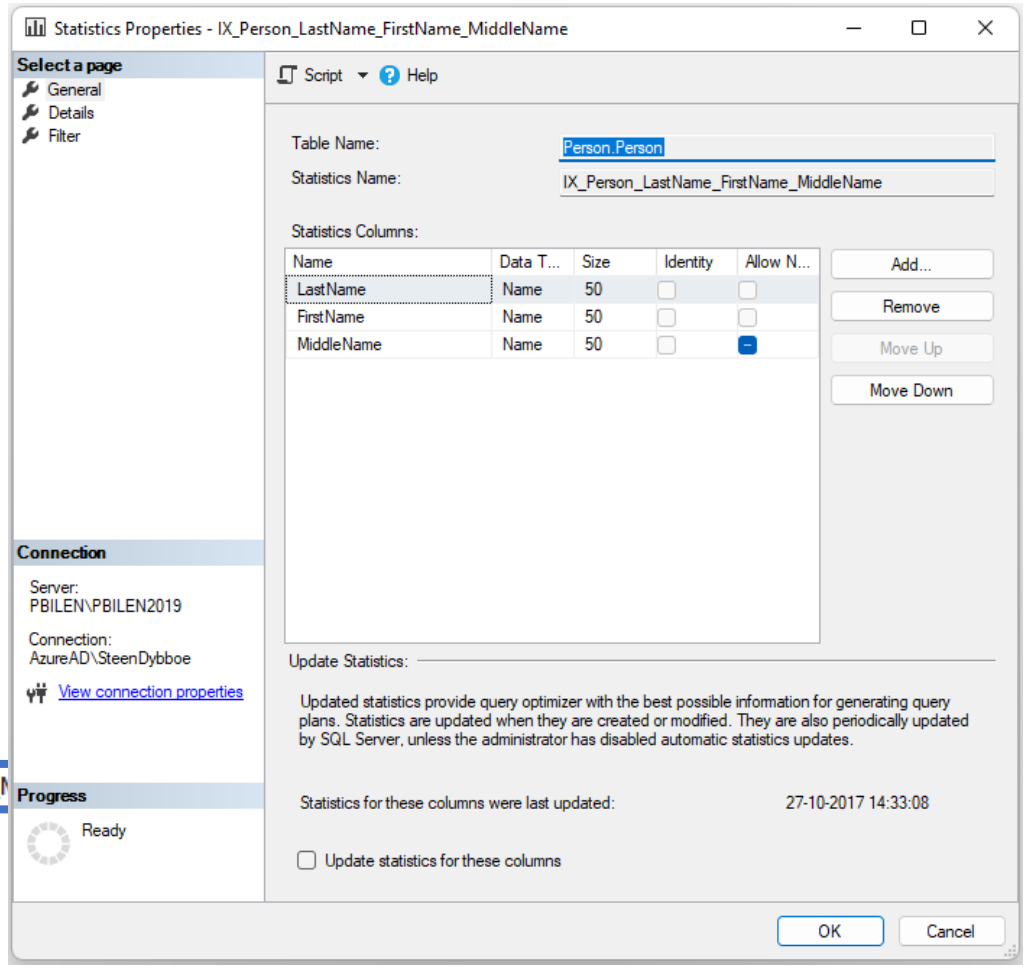
Name	Updated	Rows	Rows Sampled	Steps	Density
IX_SalesOrderDetail_ProductID	Oct 27 2017 2:33PM	121317	121317	200	0,0078125

All density	Average Length	Columns
0,003759399	4	ProductID
8,242868E-06	8	ProductID, SalesOrderID
8,242868E-06	12	ProductID, SalesOrderID, SalesOrderDetailID

RANGE_HI_KEY	RANGE_ROWS	EQ_ROWS	DISTINCT_RANGE_ROWS	AVG_RANGE_ROWS
707	0	3083	0	1
708	0	3007	0	1
709	0	188	0	1
710	0	44	0	1
711	0	3090	0	1
712	0	3382	0	1
713	0	429	0	1
714	0	1218	0	1
715	0	1635	0	1
716	0	1076	0	1
718	218	219	1	218
719	0	44	0	1
722	0	392	0	1
723	0	52	0	1
725	0	374	0	1
726	0	288	0	1
727	0	48	0	1
729	0	375	0	1
730	0	288	0	1
732	0	130	0	1
738	154	600	2	77
741	167	94	1	167
742	0	288	0	1
743	0	481	0	1
747	89	289	2	44,5
748	0	491	0	1
749	0	470	0	1
750	0	358	0	1
751	0	409	0	1
752	0	378	0	1
753	0	475	0	1
754	0	226	0	1
756	164	171	1	164
757	0	77	0	1
758	0	267	0	1
759	0	282	0	1
760	0	680	0	1
761	0	638	0	1
762	0	689	0	1
763	0	649	0	1

Management Studio





Statistics Creation

- Always on columns in an index
- Automatically when used in Joins or Queries
  - Synchronous or Asynchronous
  - When activated
- Explicit
  - CREATE STATISTICS

Statistics on a Multicolumn Index

- Statistics for a multicolumn index consist of
  - One histogram for the first column only
  - Multiple density values for the following
- Therefore, it it a good practice to put the more selective column first
  - The one with the lowest density
- May be created by an index or from manually created statistics
  - Never from the automatic statistics creation process

Statistics Tools

- DBCC SHOW_STATISTICS
  - Displays Current statistics for for table or index view
  - Displays the header, histogram, and density vector based on data stored in the statistics object

- STATS_DATE

```
DBCC SHOW_STATISTICS ('sales.invoices', 'PK_Sales_Invoices’)
WITH HISTOGRAM
```

Statistics Tools

- sys.stats
  - Add parameter “ALL” to get statistics for both stats created on index and columns
- sys.stats_columns

Statistics Tools – DMV's

- sys.dm_db_stats_properties (object_id, statistics_id)
  - Returns properties of statistics for the specified database object

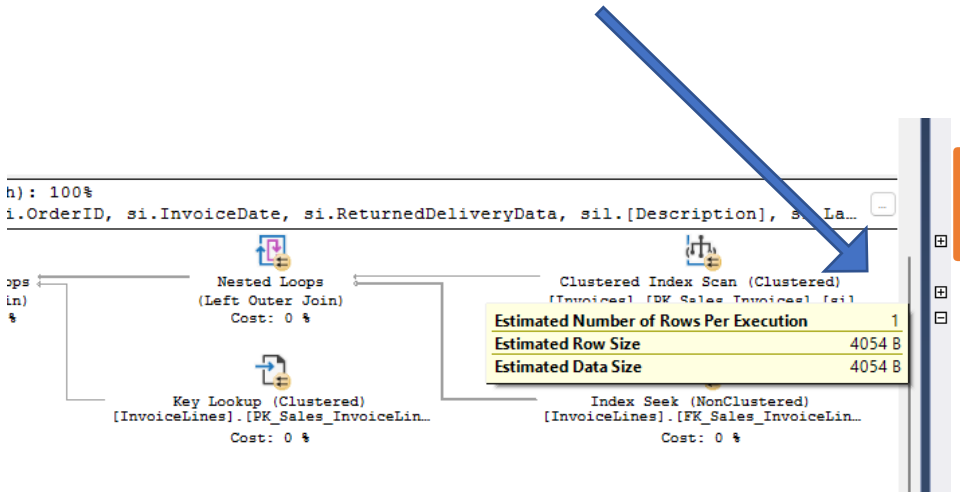
- sys.dm_db_stats_histogram (object_id, stats_id)
  - Relational output

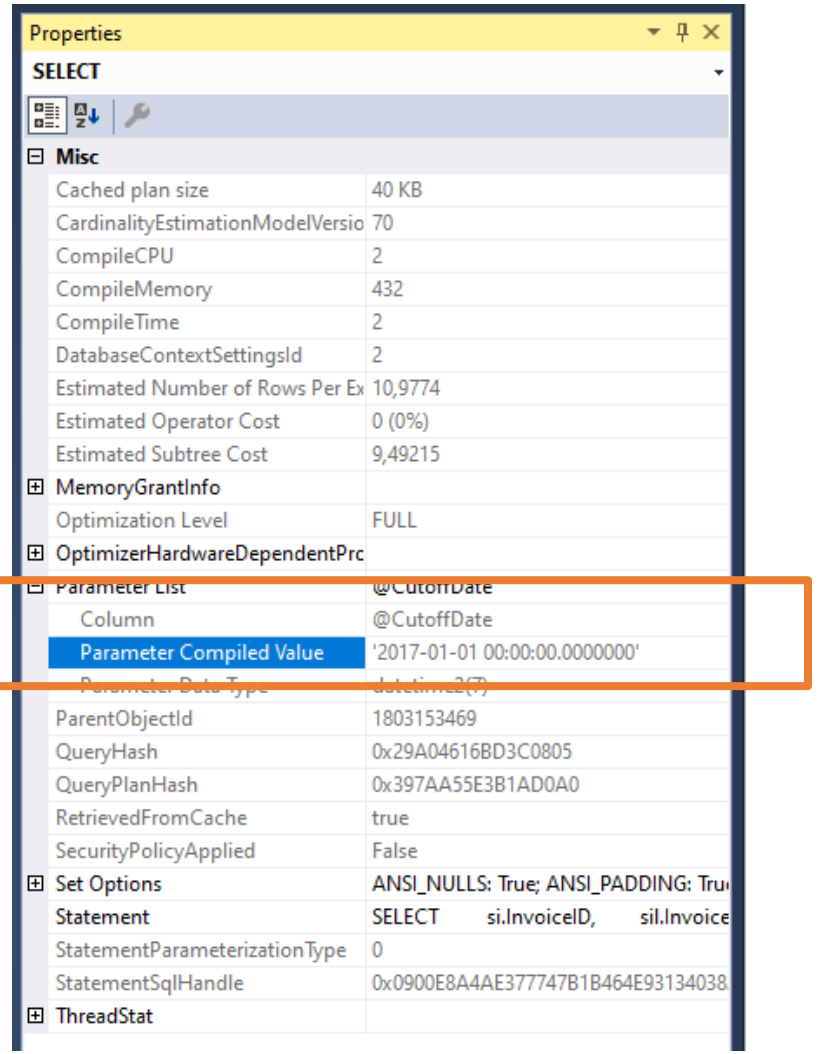
```
SELECT *
FROM sys.dm_db_stats_properties (228195863, 2)
```


DEMO

Properties Window

- Runtime ”2017-04-01”



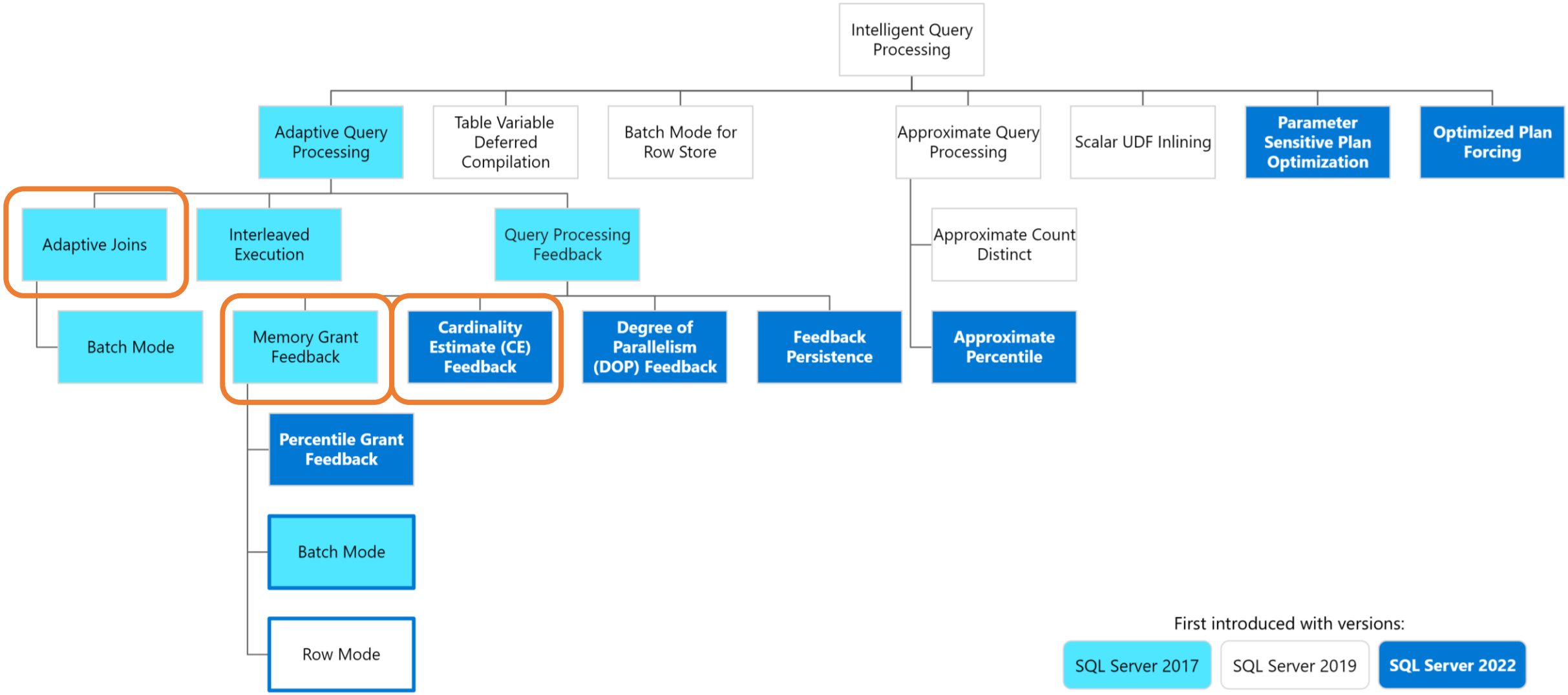


Statistics on a Filtered Index

- Creates a more LEAN statistic
  - Following the data in the filtered index
  - Slightly better (lower) density value

- Statistics are not automatically created on partitioned tables
  - Not possible to create using CREATE STATISTICS
  - But you may Create filtered indexes by partition and get statistics

Intelligent Query Processing (2022)



Statistics Maintenance - Automatic

- Four ways to control automatic maintenace
  - Auto create index: New statistics on unindexed columns
  - Updating existing statistics
  - Degree of sampling to generate index (need not be 100%)
  - Asynchronous updating of existing statistics

- The Auto Create Statistics setting is applicable for nonindexed columns only

Statistics Maintenance - Manually

- Higher degree of WORKLOAD predictability
- Stop statistics maintenance from occurring randomly
  - Though blocking is NOT an issue
- Manual interference relevant when
  - Experimenting
  - Often after server upgrades (Query Store)
  - When primary focus for a database is Ad-hoc querying (auto not required)
  - If data changes does not trigger a needed update
  - Bad Query plans signals bad statistics

Update Statistics

- Correct statitics are paramount to performance
- Performaing updates
  - Automatic
    - Used to be 20% of data in a table changed for tables larger than 500 rows
    - Problematic for large tables
  - UPDATE STATISTICS statement
- sp_updatestats [[@resample =] 'resample’]
  - Update only statistics for items in which there have been **sufficient modifications** to the data.
  - runs the UPDATE STATISTICS statement using the ALL option

UPDATE STATISTICS

```
UPDATE STATISTICS table_or_indexed_view_name
[
    {
        { index_or_statistics_name } | ( { index_or_statistics_name }
        [ ,...n ] )
}
]
[ WITH
    [ FULLSCAN
    [ [ , ] PERSIST_SAMPLE_PERCENT = { ON | OFF } ]
            | SAMPLE number { PERCENT | ROWS }
    [ [ , ] PERSIST_SAMPLE_PERCENT = { ON | OFF } ]
        | RESAMPLE
        [ ON PARTITIONS ( { <partition_number> | <range> } [, ...n] ) ] ]
        [ [ , ] [ ALL | COLUMNS | INDEX ]
        [ [ , ] NORECOMPUTE ]
        [ [ , ] INCREMENTAL = { ON | OFF } ]
        [ [ , ] MAXDOP = max_degree_of_parallelism ]
] ;
```

AUTO_UPDATE_STATISTICS

- Up to SQL Server 2014 (120) based on number of rows in table
  - n > 500
  - Modification Threshold = 500 + (n * 20%)
  - Problems with very large tables

- From SQL Server 2016 (130) based on number of rows in table
  - n > 500
  - Modification Threshold = MIN(500 + (n * 20%) , SQRT(1000 * n))
  - Compensates for very large tables

2.000.000	400.500	44.721

Update will trigger at 44.721

Intelligent Maintenance

- Complex decisions
  - Update when index is being reorganized
  - Significant data changes
- Update Column statistics
  - Might be expensive
  - One table scan for each column
    - thus, slower than Index Statistics scanning
- Should it depend on #modifications
  - Based on Counters
  - Therefore, use script from …

Frameworks

- If you NEED to manage STATISTICS UPDATE yourself, be structured about it
- Don’t invent a framework yourself
- Ola Hallengren
  - His framework is a gift to all SQL Server Admins
  - ... And it is FREE
  - ola.hallengren.com

Query Tuning and Statistics

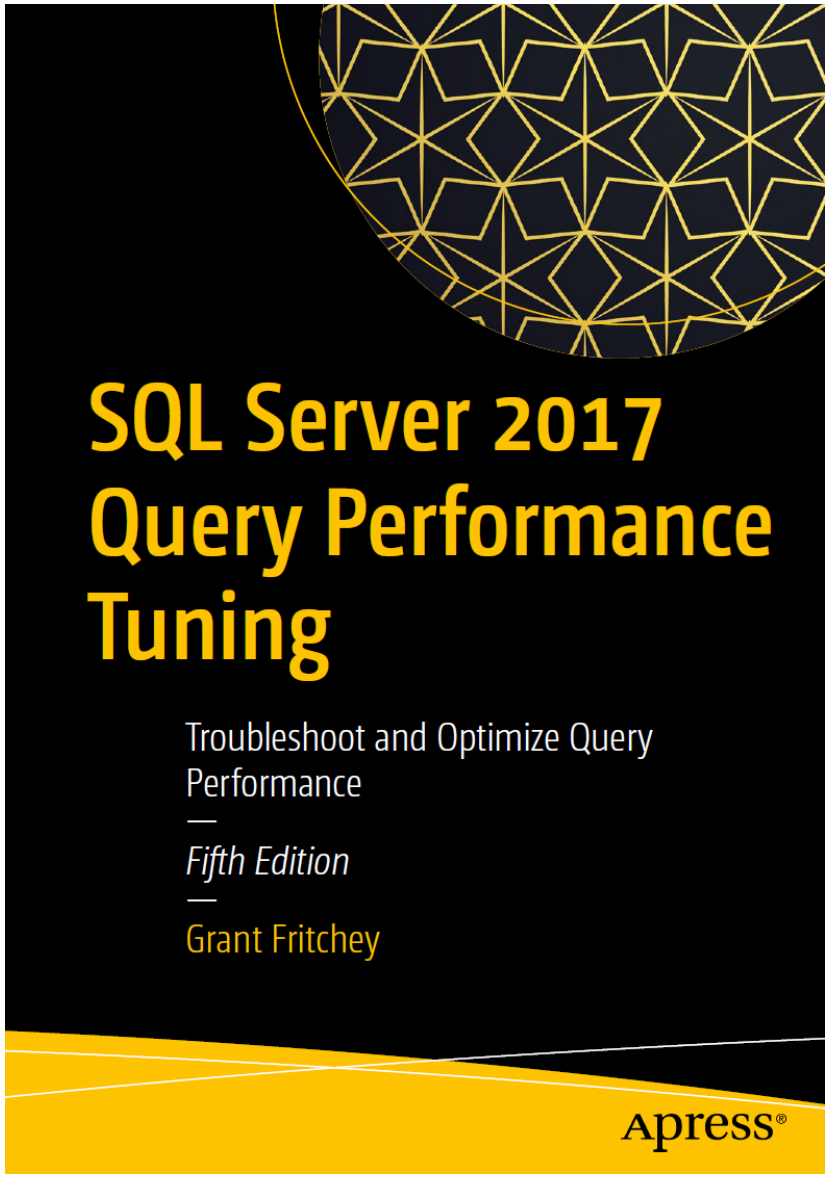
- Indexes are available on the columns referred to in the filter and join criteria
- For non-indexed columns used in filter and join make sure statistics exists
  - Either by automatic or manual process
- Make sure all your statistics are updated
  - The easy-win with complex queries

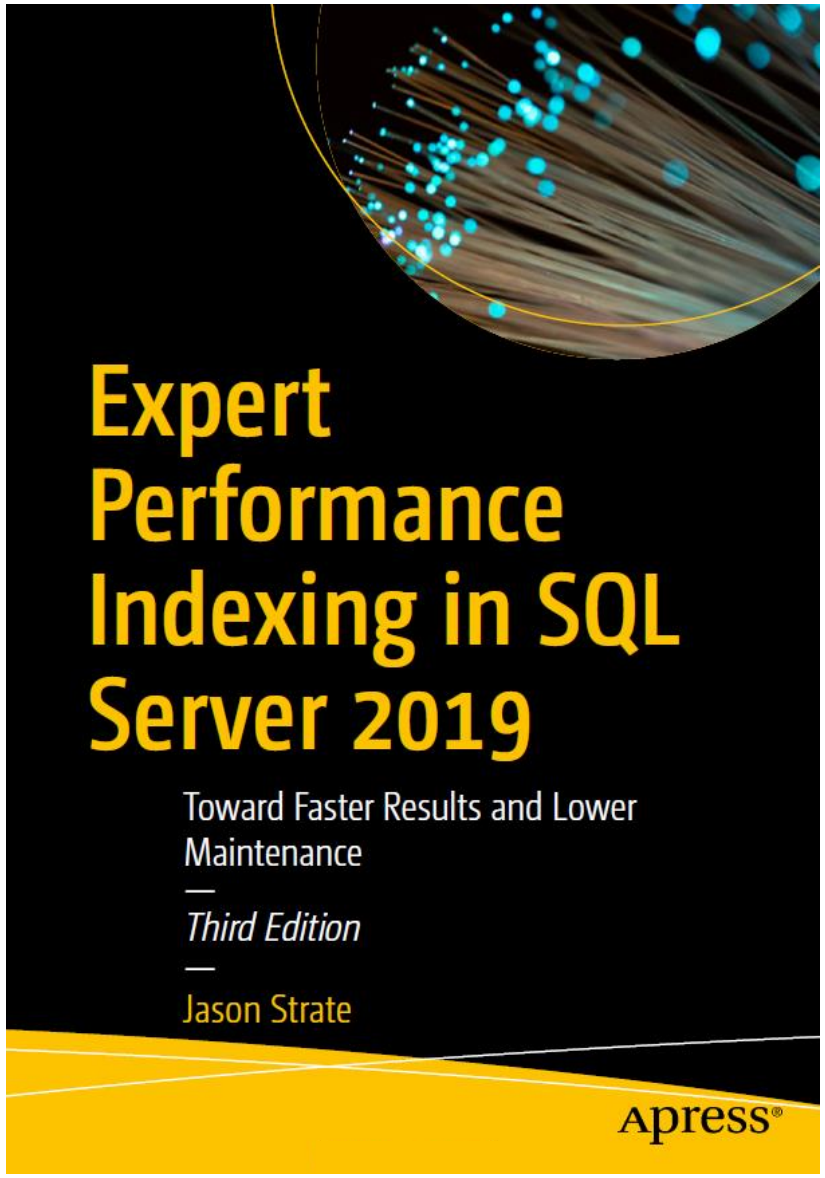
Recommandations

- Auto CREATE Statistics
- Auto UPDATE statistics
- Auto UPDATE statistics Asynchronously
- Beware of Sampling Size
  - Less than FULLSCAN may yield problematic statistics on skewed data
  - FULLSCAN may drain your resources

The Central Question

Is my stats out of date?


SQL Server 2017
Query Performance
Tuning
Troubleshoot and Optimize Query Performance
—
Fifth Edition
—
Grant Fritchey
Apress®


Expert
Performance
Indexing in SQL
Server 2019
Toward Faster Results and Lower Maintenance
—
Third Edition
—
Jason Strate
Apress®

Questions

Pragmatic BI

Pragmatic BI Aps
Åhusene 4,9,3
8000 Arhus C

Telefon +45 42 606 202
E-mail steen@prabi.dk
Internet www.prabi.dk
CVR nr. 36974133