

# Validating Assumptions

**Joe Sack**

[Joe@SQLskills.com](mailto:Joe@SQLskills.com)

<http://www.SQLskills.com/blogs/Joe>

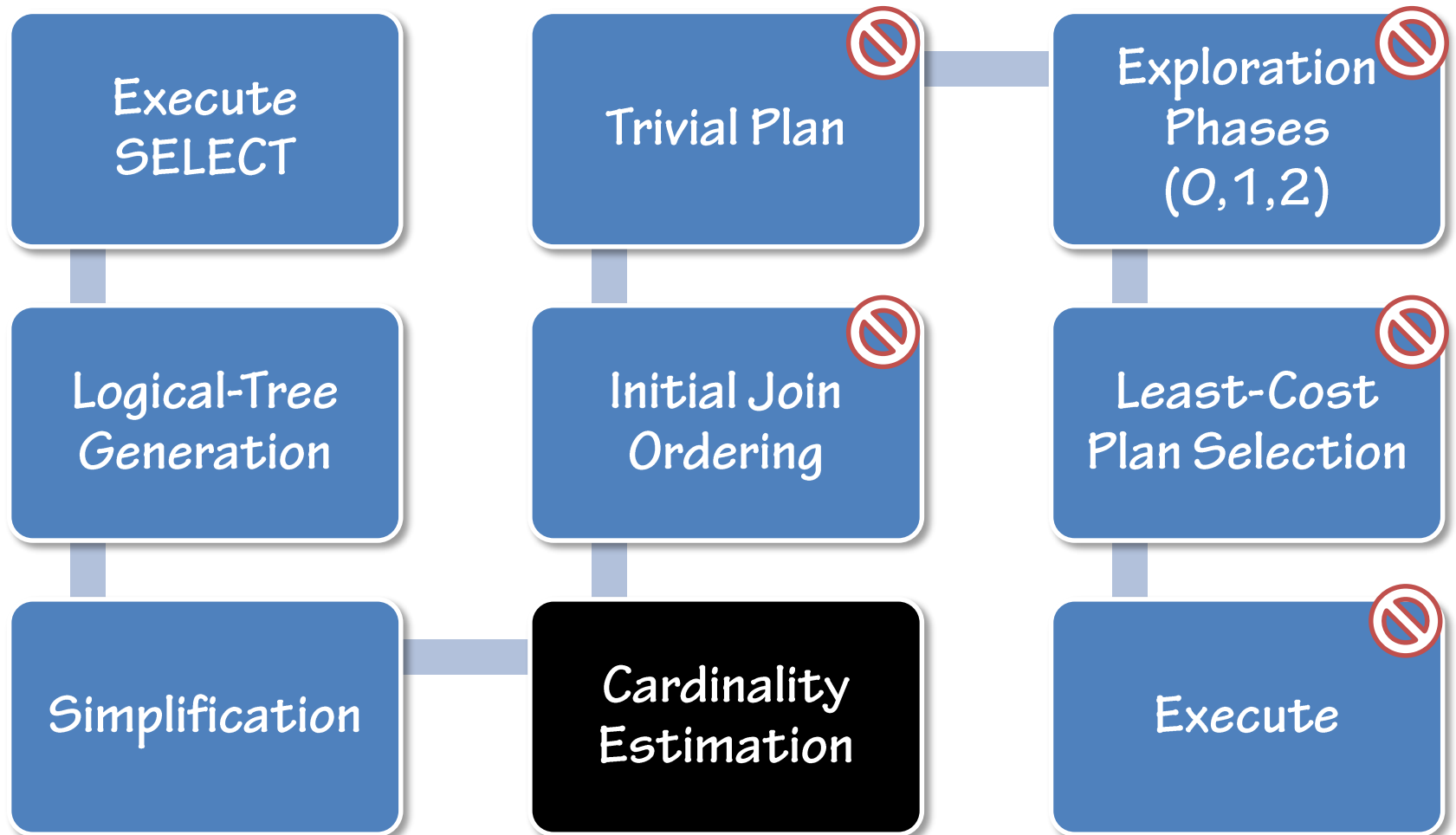


**pluralsight**   
hardcore developer training

# Module Introduction

- **A critical step when tuning a poorly-performing query is to ensure that the original assumptions being made by the query optimization process are actually correct**
  - When not correct, you could be troubleshooting downstream side effects and NOT the actual root cause
- **In this module we'll walk through how to validate query execution assumptions in order to ensure you're troubleshooting the true root cause of a query performance issue**

# Downstream Impact of Bad Assumptions

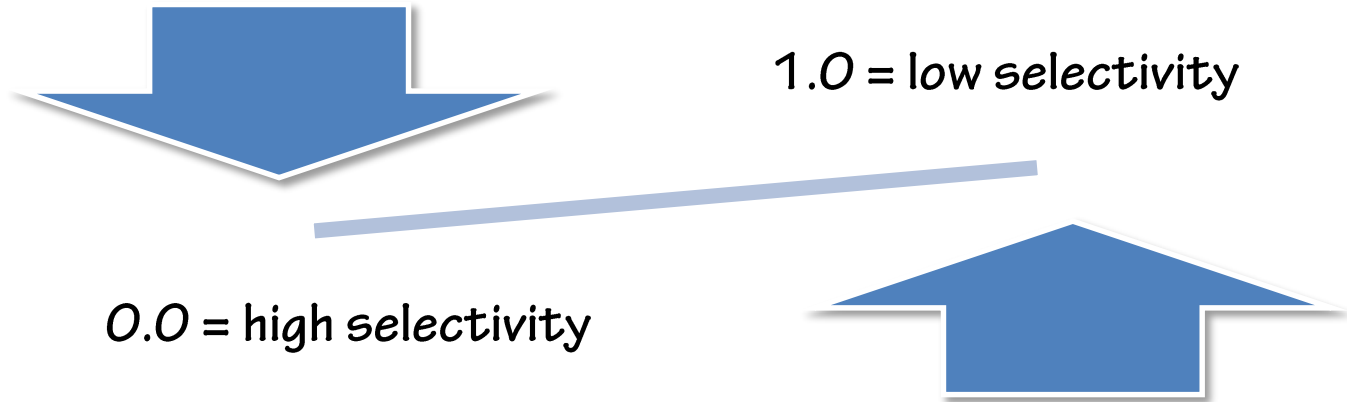


# Cardinality Estimation and Predicates

- Cardinality estimates are row count estimates calculated for each operator within a query execution plan
- The cardinality estimation process is interested in the *predicates* you have defined in a T-SQL query
  - Predicates are expressions that evaluate to TRUE, FALSE or UNKNOWN
    - Filter predicates are used in a search condition via a WHERE and/or HAVING clause of a T-SQL query
    - Join predicates are used in JOIN conditions of the FROM clause
- The cardinality estimation process is interested in determining the *selectivity* of a predicate

# Selectivity

Number of rows passing the predicate / Total number of rows



How many products  
have a “red” color?

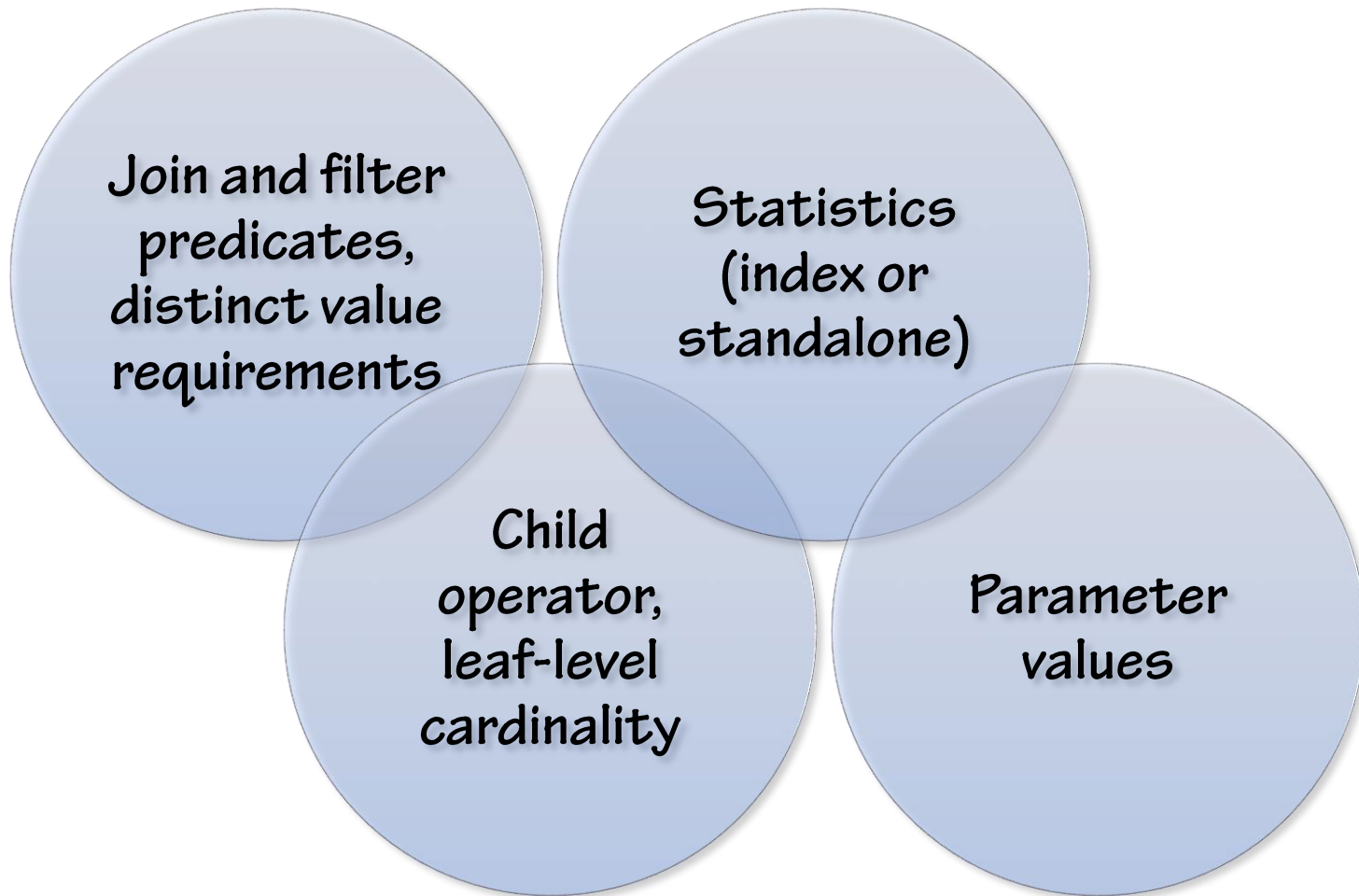
432/1,600,000  
rows

0.00027  
selectivity

# Query Cardinality Questions

- If you're using WHERE, HAVING, JOIN, GROUP BY, and/or DISTINCT constructs then you're giving the Cardinality Estimator questions it will try to answer
  - If it answers them incorrectly, depending on *how* incorrectly, the query performance will suffer
- Therefore an effective query tuning technique is to start with improving the quality of answers to the following questions:
  - What is the selectivity of a query with a single filter predicate?
  - What is the selectivity of a query with multiple filter predicates?
  - What is the selectivity of a join predicate between two tables?
  - How many distinct values do we expect from a specific column?
  - How many distinct values do we expect from a set of columns?
    - Said another way: how many unique combinations do we expect?

# Cardinality Estimate Inputs

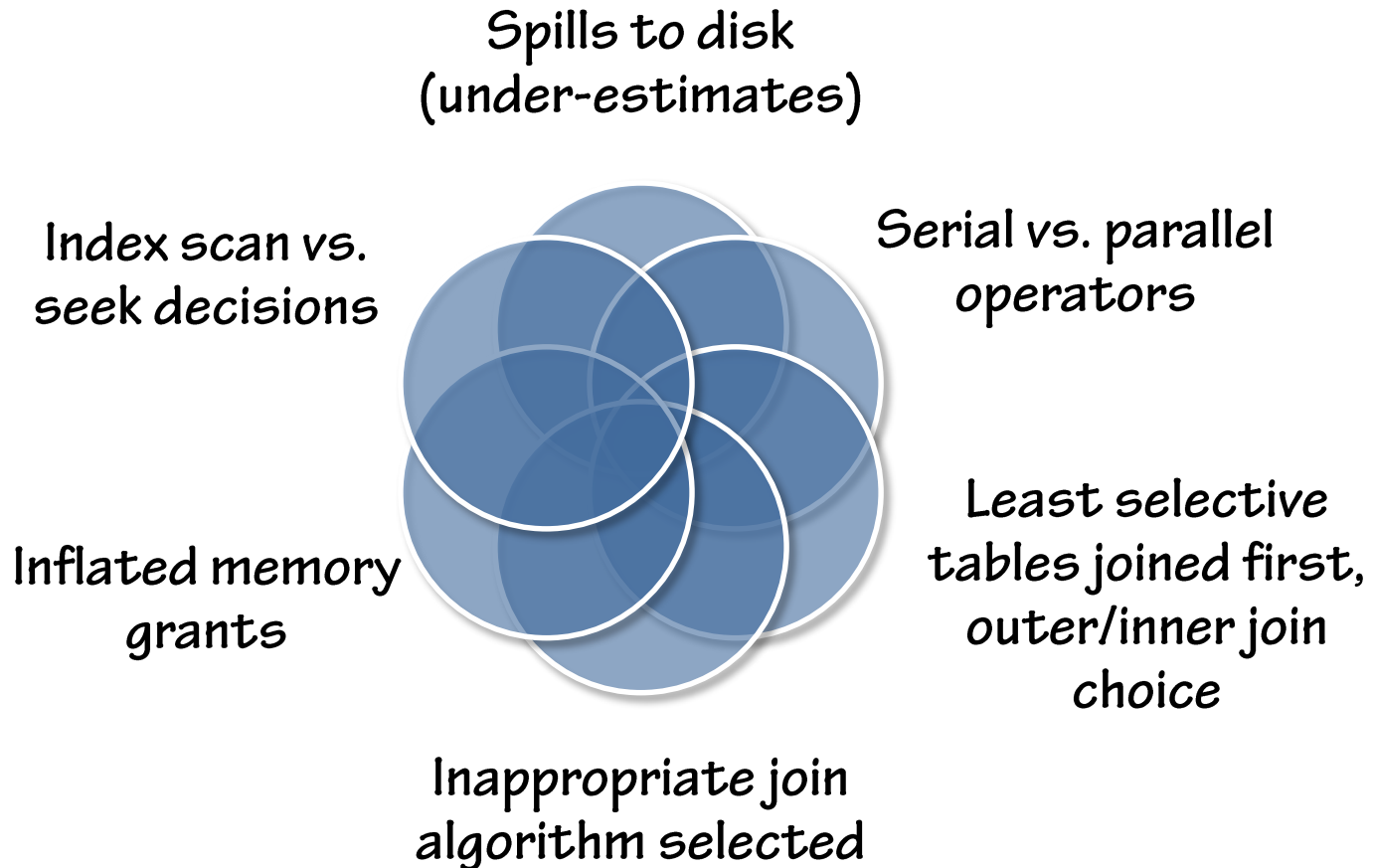


# Statistics

- Statistics drive cardinality estimates
- If the statistics don't reflect reality, your query execution plan is unlikely to include the most optimal choices
- Understanding how to analyze statistics objects (index or standalone) is a critical skill when used for tuning queries in SQL Server
- You'll want to be very familiar with DBCC SHOW\_STATISTICS and other methods for finding out which statistics were used for a specific query



# Impact to Query Execution and Performance



# Identifying Bad Assumptions (1)

- The estimated plan will NOT help you in diagnosing CE issues – but the actual plan will:
  - SET STATISTICS XML
  - SET STATISTICS PROFILE (deprecated)
  - Graphical Showplan (“Include Actual Execution Plan”)
  - SQL Sentry Plan Explorer
    - See Jonathan Kehayias’ course *SQL Server: Using SQL Sentry Plan Explorer*, <http://bit.ly/PSPlanE>



Operation	Object	Est Cost	Est Subtree Cost	Actual Rows	Est Rows
SELECT		100.0%	100.0%	15,554	1
Nested Loops (Inner Join)		0.0%	100.0%	15,554	1
Table Scan	[Credit].[dbo].[payment].(Heap)	25.1%	33.4%	15,554	1
Clustered Index Seek	[Credit].[dbo].[member].[member_iden...]	74.9%	99.5%	15,554	3

SQL Sentry Plan Explorer

# Identifying Bad Assumptions (2)

- `query_post_execution_showplan` Extended Event and equivalent SQL Trace events \*
- SQL Server 2014's `sys.dm_exec_query_profiles` dynamic management view for real-time query execution progress-monitoring
- Root-level skew via `sys.dm_exec_query_stats`



*\* Be careful of “observer overhead”!*

# What Variance is Problematic?

- 100 rows? 1,000 rows? 1,000,000 rows?
- **Not all skews matter, so ask these questions:**
  - Does the row estimate skew affect plan choices?
    - Requires a solid understanding of query execution plans
    - See the *SQL Server: Query Plan Analysis* course <http://bit.ly/SackQPA>
  - Do the plan choices result in excessive resource consumption or excessive execution time?

# **Module Summary**

- **In this module we walked through the importance of validating the assumptions made during the query optimization process**
- **You can waste quite a bit of time troubleshooting a query that is performing poorly due to incorrect assumptions**
- **You learned about some of the key areas where you can validate issues with the underlying assumptions**
- **In the next module we'll look at some of the common problems that can occur due to bad assumptions and look at some of the viable associated solutions to help improve query performance**