Tune the query, not the plan

SQL Saturday 730 Manchester – 14 July 2018

Kevan Riley

Principal Consultant, Riley Waterhouse

Kevan Riley

rileywaterhouse

Freelance SQL Server consultant

19+ years from SQL Server v7.0

kevan.riley@rileywaterhouse.co.uk

@kevriley

Microsoft CERTIFIED
Solutions Expert
Data Management and Analytics

Execution plans

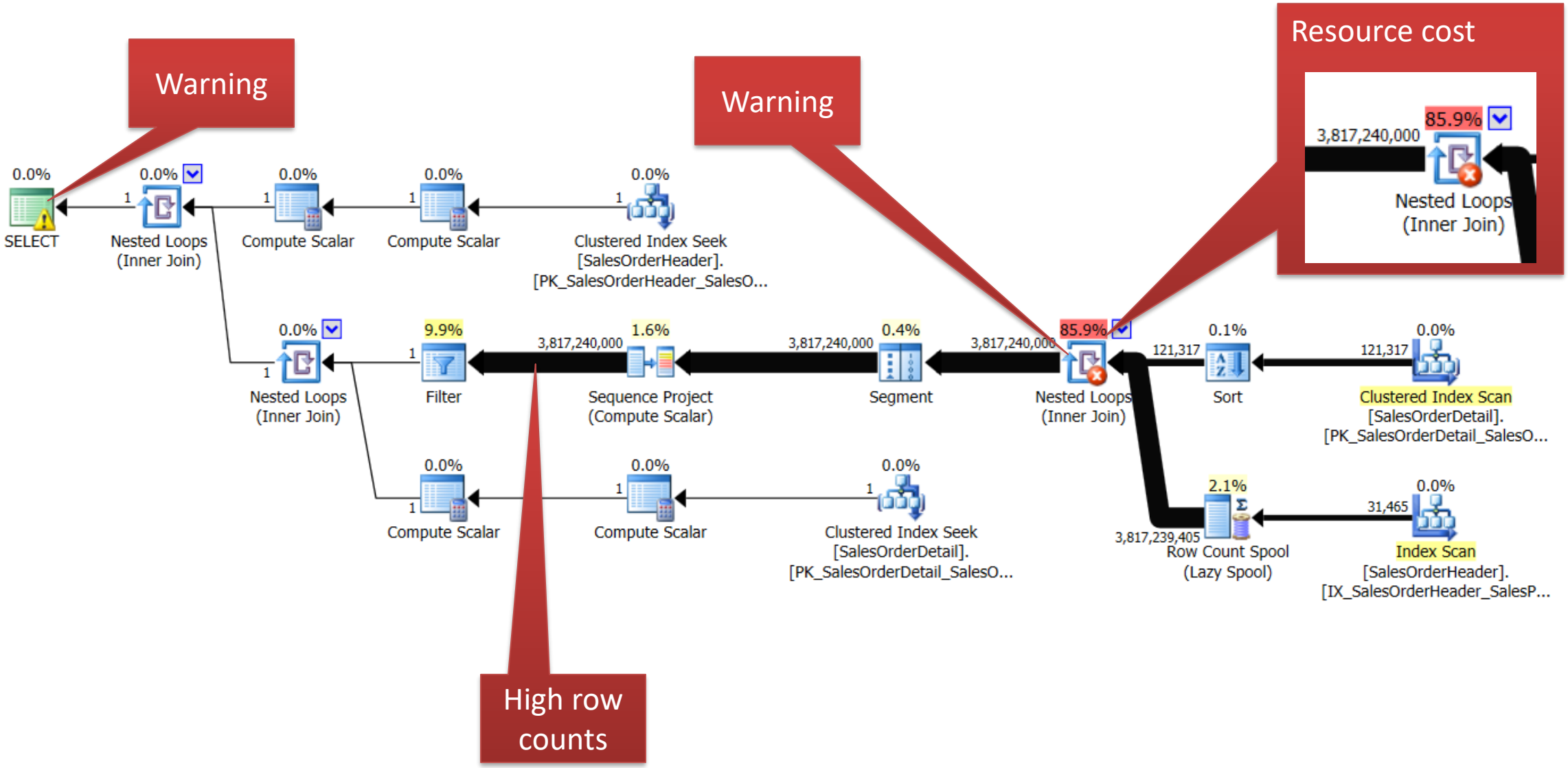
- Generated by Query Optimizer
  - Cardinality Estimator
  - Statistics
  - YOUR design
- How SQL will execute the query
  - Table access
  - What order
  - Join algorithm
- VERY useful in performance tuning

¯_(ツ)_/¯

So what's the problem?

Your time is precious

- DBA firefighting
- Looking for quick wins
- Focus on ‘bad’ operators
  - Table Scans
  - Scan operations in general (prefer seek)


Warning
Warning
Resource cost
High row counts
SELECT
Nested Loops (Inner Join)
Compute Scalar
Compute Scalar
Clustered Index Seek [SalesOrderHeader]. [PK_SalesOrderHeader_SalesO...
Nested Loops (Inner Join)
Filter
Sequence Project (Compute Scalar)
Segment
Nested Loops (Inner Join)
Sort
Clustered Index Scan [SalesOrderDetail]. [PK_SalesOrderDetail_SalesO...
Compute Scalar
Compute Scalar
Clustered Index Seek [SalesOrderDetail]. [PK_SalesOrderDetail_SalesO...
Row Count Spool (Lazy Spool)
Index Scan [SalesOrderHeader]. [IX_SalesOrderHeader_SalesP...
3,817,240,000
85.9%
Nested Loops (Inner Join)

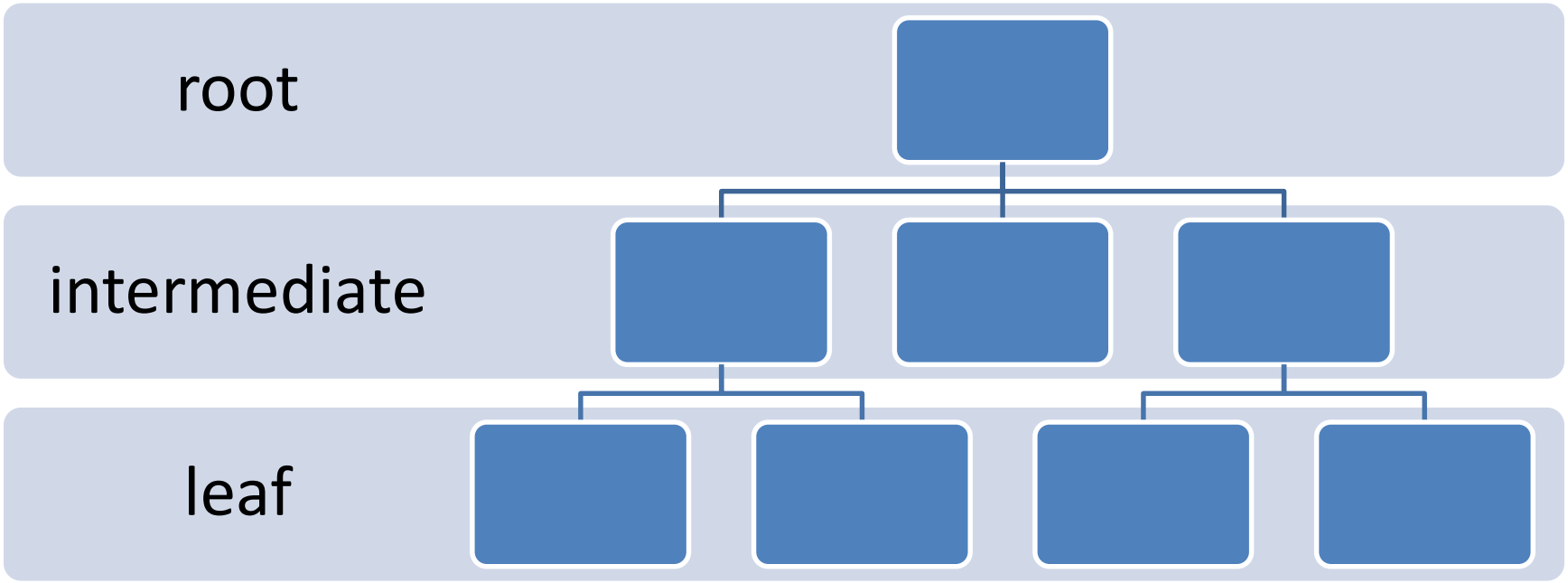
Performance tuning?

- Not just the plan
- Time
- CPU
- Reads/Writes

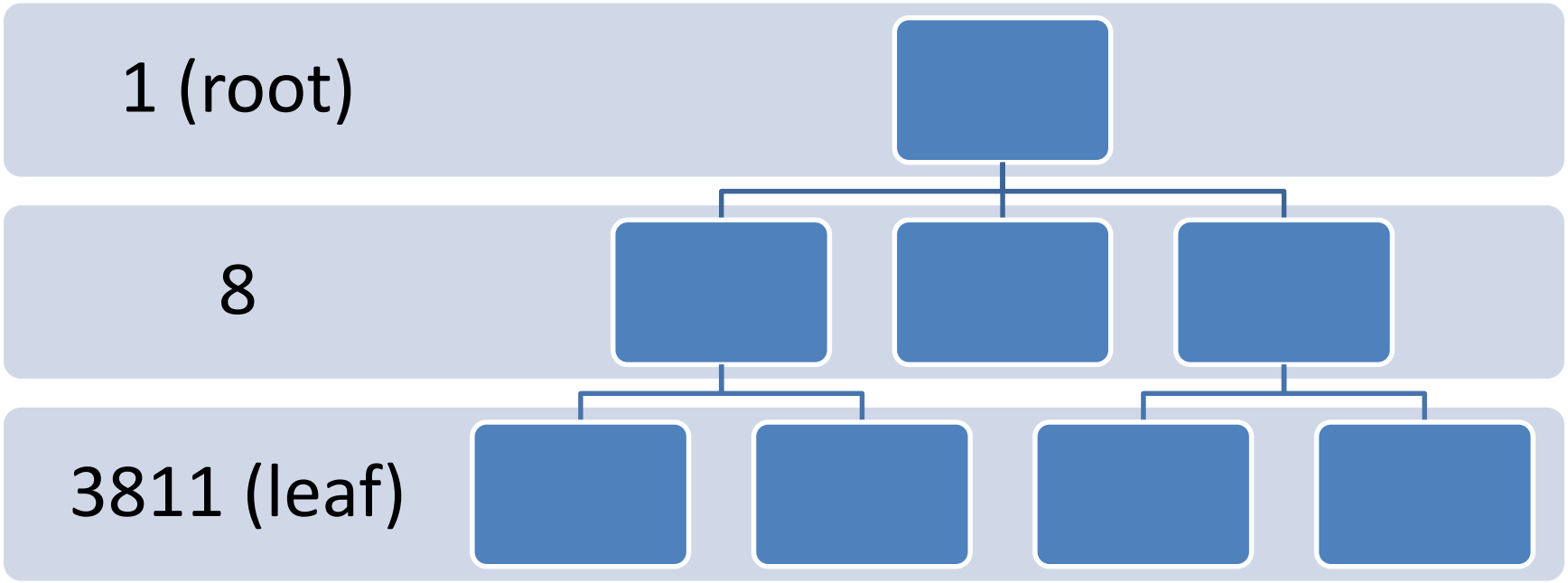
DEMO

- Table Scans – removing from plan
- Index Scans vs Index Seeks
  - Does a scan always read all the table?
  - Does a seek only return a few rows?
- How to confuse/coerce the Optimizer
- Constraints

Index structure



Person.Person
PK_Person_BusinessEntityID



==========

3820

Conclusions

- Replacing Table Scans with CIX Scans is not a perf gain
- Scans do not always read the entire table
- Seeks can return the entire table
- Scan vs Seek – not comparable
  - Seek gives a starting point – needs a predicate on the key
  - Scan starts at one end of an index
- Design of query and schema is important

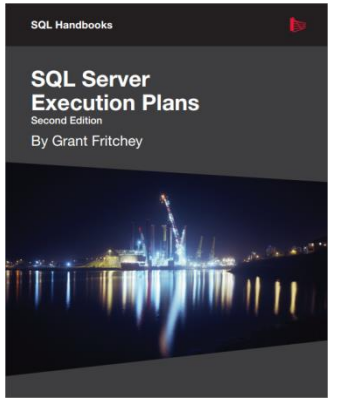
Performance tuning – what should we use?

- set statistics io
- set statistics time
- execution plans
  - understand why you are getting certain operators
  - is there a better way?
- wait stats

Execution Plan Resources

- SentryOne Plan Explorer
  https:/www.sentryone.com/plan-explorer

- SQL Server Execution Plans – Grant Fritchey
  https://www.red-gate.com/library/sql-server-execution-plans-2nd-edition



- SQL Server Execution Plan Reference – Hugo Kornelis
  https://sqlserverfast.com/epr

Useful links

- Constraints
  - https://rileywaterhouse.co.uk/helping-the-query-optimizer-with-constraints/
  - https://rileywaterhouse.co.uk/trusting-constraints/
- SET STATISTICS
  - https://docs.microsoft.com/en-us/sql/t-sql/statements/set-statistics-io-transact-sql
  - https://docs.microsoft.com/en-us/sql/t-sql/statements/set-statistics-time-transact-sql
  - https://blogs.msdn.microsoft.com/sqlserverfaq/2010/09/14/scan-count-meaning-in-set-statistics-io-output
- Waits
  - https://www.sqlskills.com/blogs/paul/wait-statistics-or-please-tell-me-where-it-hurts/
  - https://www.brentozar.com/sql/wait-stats/

Thanks for listening

kevan.riley@rileywaterhouse.co.uk
rileywaterhouse.co.uk
@kevriley