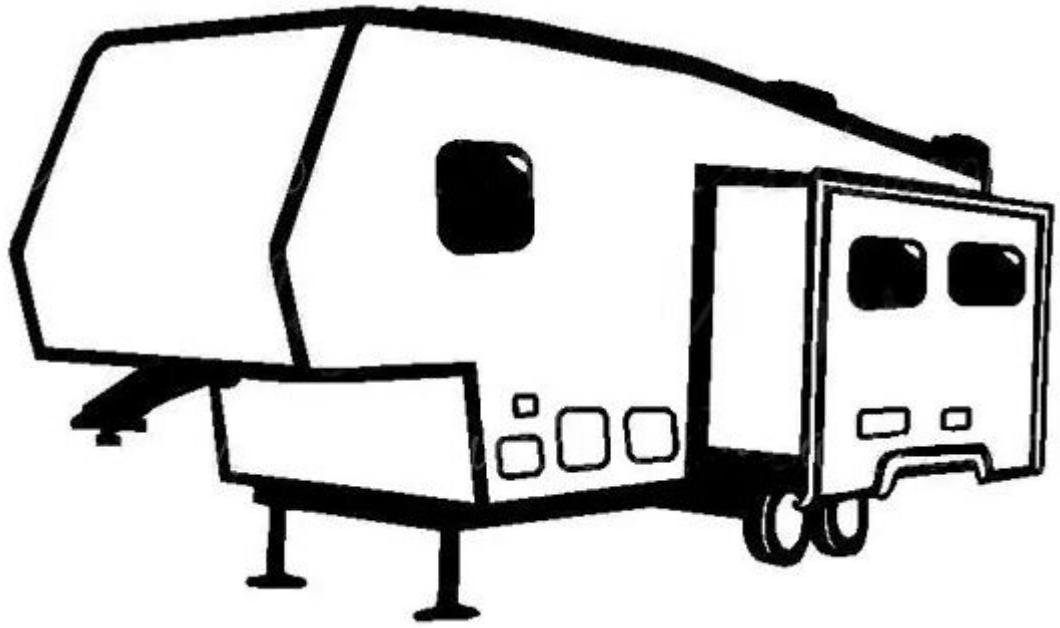
Aaron N. Cutshall, DHA, MSHI

SQL RV http://www.sqlrv.com



“There Be Whales Here!”
– Big Data on SQL Server

Just who is this guy?

SQL RV

healthfirst
Health Insurance for New Yorkers

Enterprise Data Architect

ENTERPRISE DATA WORLD
PASS SQLSATURDAY

Speaker – various events

B.S. Computer Science

M.S. Computer Information Systems

M.S. Health Informatics

Doctor of Healthcare Administration

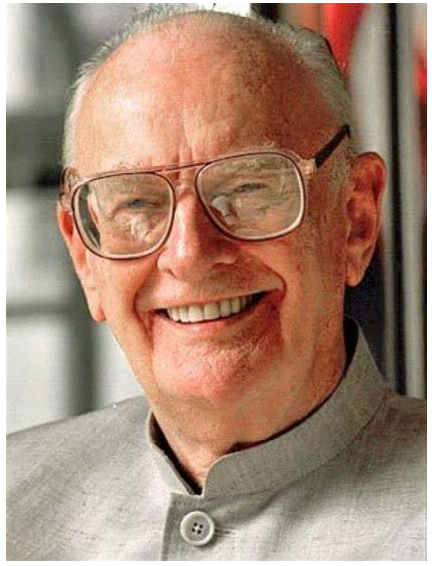
35+ Years

Copyright © 2023, SQL RV - All Rights Reserved

“Admiral! There be whales here!”

Copyright © 2023, SQL RV - All Rights Reserved

Something to consider…



credit

"Any sufficiently advanced technology is indistinguishable from magic."

– Sir Arthur Charles Clarke (1917 – 2008)

A celebrated author of many science-fiction classics

Copyright © 2023, SQL RV - All Rights Reserved

Agenda

- What is Big Data?
  - Some Definitions
  - Common Myths
- When Big Data Goes Bad
  - SQL Server vs MongoDB
  - SQL Server vs Hadoop
  - Observations
- Big Data for SQL Server
  - CoreAnalytics Overview
  - Big Data Techniques
  - Alternative Options
- Closing Thoughts

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – Some Definitions

- Oxford English Dictionary:

  "Extremely large data sets that may be analyzed computationally to reveal patterns, trends, and associations, especially relating to human behavior and interactions."

- Keywords:
  - Large
  - Analyze
  - Patterns
  - Trends
  - Associations

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – Some Definitions

- Merriam-Webster Dictionary:

  "An accumulation of data that is too large and complex for processing by traditional database management tools."

- Keywords:
  - Accumulation
  - Large
  - Complex

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – Some Definitions

- Gartner Research:

  “High-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.”

- Keywords:
  - High-Volume
  - High-Velocity
  - High-Variety
  - Cost-Effective
  - Innovative
  - Enhanced Insight
  - Decision Making
  - Process Automation

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – Some Definitions

- MIT Technology Review

  "Big data is a term describing the storage and analysis of large and or complex data sets using a series of techniques including, but not limited to: NoSQL, MapReduce and machine learning."

- Keywords:
  - Storage
  - Analysis
  - Large
  - Complex
  - NoSQL
  - MapReduce
  - Machine Learning

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – Some Reality

- The definition of Big Data is a moving target
- Berkeley University surveyed 40 industry thought leaders: "What is Big Data" – interestingly, no one agreed
  - Most people think they understand what Big Data is
  - Everyone's definition is skewed by their perspective
- Big Data Characteristics:
  - The Three V's: Volume, Variety, Velocity
  - The Four Additional V's: Veracity, Variability, Visualization, Value
  - Structured vs. Unstructured: Huge disagreements on the definition

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – MythBusters

- Big data is a new phenomenon that only came about in recent years
  - The Genome Data Base uses an RDBMS for most of its data in mapping human DNA
  - The Astronomical Data Model is on an RDBMS system used for many years before PCs
- To process big data requires totally new platforms and thought processes
  - Marketing articles targeted toward C-level executives
  - Promotes the myth that companies will not survive without embracing Big Data

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – MythBusters

- Because big data is so large it can be difficult to process
  - Complexity is due to how data is organized
  - It's a myth that only NoSQL platforms can handle big data
  - The assumption is that scaling can only be achieved by sharding
- NoSQL vendors and training companies promote myths: "Like Hobbes is to Calvin, Hadoop is to Big Data."
- Big Data applications require little or no performance tuning.
  - ***ALL*** data systems require tuning for proper performance
  - Remember how "Plug & Play" became "Plug & Pray"?

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – MythBusters

- Analytics requires Big Data
  - Analytics has been performed for years in RDBMS environments
  - Insurance companies always had “data scientists” called actuaries
- Big Data is only unstructured data
  - Assumption that all Big Data is unstructured or semi-structured
  - Big confusion between unstructured and non-discrete
  - Structure is imposed in storage (RDBMS) or retrieval (NoSQL)
- Big Data is too big for SQL
  - All MPP platforms support parallel processing & SQL
  - Netezza, Teradata, Exadata, Azure, RedShift, Snowflake, etc.
  - Most NoSQL environments now have SQL interfaces

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – Overheard

- "Big Data is anything over 500GB"
  - Lots of SQL Server environments have much mor data
  - Remember: Size is not everything!
- "Only NoSQL systems can scale"
  - Most consider scaling to involve multiple servers
  - Microsoft Azure, Snowflake, and AWS Redshift are quite scalable
- "NoSQL allows faster development"
  - Many NoSQL implementations begin with little planning
  - Planning is required to design proper databases
  - Do not confuse *activity* with *productivity*!

Copyright © 2023, SQL RV - All Rights Reserved

What is Big Data? – The Results

- Gartner survey 2015 analysis:
  - Plans for Hadoop implementations fell to 46%
  - Only 17% of Hadoop deployments made it to production
  - Only 15% of the production deployments were successful
- Why?
  - Hadoop was overkill for most business problems
  - Lack of maturity and an enterprise focus
  - Inconsistent query results with the same dataset
- To survive, Cloudera & Hortonworks merged in Jan 2019
- MapR acquired by Hewlett Packard Enterprise in Aug 2019

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – MongoDB

- Scenario (circa 2012)
  - Clinical Quality Measures
  - Denominator – What is to be measured (ex. diabetic patients)
  - Numerator – The measure itself (pass/fail) (ex. retinal eye exam in 12 months)
  - Clinical data identified by taxonomy and term
  - Data per patient contains extensive records:
    - Encounters
    - Diagnoses
    - Procedures
    - Medications
    - Patient History
    - Allergies & Interactions
    - Lab Orders & Results
    - Vital Signs

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – MongoDB

- MongoDB version
  - 40 Clinical Quality Measures of varying complexity
  - Data for 750,000 patients
    - Stored in Clinical Document Architecture (XML) documents
    - One document per patient with all medical history
  - Data sharded across 10 MongoDB servers
  - Used Ruby-on-Rails for processing logic
  - Results generated in 8 hours

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – MongoDB

- SQL Server version
  - Same 40 Clinical Quality Measures
  - Data for 2.5 million patients (stored on my laptop)
  - Data normalized
    - Broken into Elements and Attributes
    - Data organization optimized for measure calculations
  - Used T-SQL for processing logic
  - Results generated in 45 minutes – on a laptop!!
  - Results generated in ~ 20 minutes on a server

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – MongoDB

- Why?
  - MongoDB version relied on procedural processing
  - Data in XML documents per patient
    - No correlation from one patient to another
    - Could not search across documents
    - Required examining each document for each measure
  - Healthcare data is highly structured and relational
  - SQL Server version
    - Used proper data storage techniques
    - Used set-based processing
    - Calculated each measure for all patients

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – Hadoop

- Scenario (circa 2016)
  - Analyze claims for billing errors, fraud, waste, and abuse
  - Claims must be paid within 24 hours of receipt
  - Claims analyst reviewed the generated recommendations before submitting for payment
  - Approx. 150 analytical concepts of varying complexity
    - Each looks at a different scenario
    - Each suspect claim has one or more reference claims
  - Ex: Cataract surgery allowed in each eye only once

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – Hadoop

- SQL Server version
  - Data copied from claims processing server once per day
  - Data architecture was copied directly from claims processor
    - Large, wide records for member, claim header and detail
    - Most data was stored as text including primary and foreign keys
    - Data structure was not optimized for analysis
  - Stored procedure for each scenario (concept)
    - Written by business analysts (non-developers)
    - Inconsistent, non-optimized logic
  - Procedures took considerable time to execute
    - Few completed within an hour
    - Some did not complete in the required 24-hour timeframe

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – Hadoop

- Hadoop version
  - Data captured as a live feed from the original server ✓
  - Data structure kept in original non-optimized format
  - Concept logic processed in SQL
    - Stored procedure logic kept mostly as-is
    - No revisit of business logic (no critical “why” questions)
    - Work-arounds for lack of update and delete functionality
  - Only simple concepts converted initially
    - Resulted in some minimal time savings
    - Could not get more than one execution per day

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – Hadoop

- Why?
  - Data architecture not optimized for analysis
  - SQL Server procedures not optimized for performance
    - Not created by developers who understood T-SQL
    - Lack of direction in business logic – little documentation
  - Hadoop chosen for “speed”
    - SQL was used for application interface – not optimized
    - Data storage as parquet files is not optimal for searches
  - Had resources been applied to the original SQL Server solution:
    - Completed sooner
    - Cost less

Copyright © 2023, SQL RV - All Rights Reserved

When Big Data Goes Bad – Observations

- The platform chosen was thought to be “THE” solution
  - No analysis performed on other approaches
  - Management picked the platform
    - High visibility – had to be impressive
    - High impact if successful
  - “But, it’s FREE!” – The hidden cost of free
- No root cause analysis during rollout
  - “Problem must be in the data or programming”
  - All-day meetings were created to “improve” productivity
- Wrong tool for the job and/or tool used incorrectly!!

Copyright © 2023, SQL RV - All Rights Reserved

Mid-Point Review

- We we've covered so far:
  - What is Big Data?
    - Some Definitions
    - Common Myths
  - When Big Data Goes Bad
    - SQL Server vs MongoDB
    - SQL Server vs Hadoop
    - Observations
- What's coming up next:
  - Big Data for SQL Server
    - Table Structures and Partitioning
    - Keys and Indexes
    - Querying Large Data Volumes
    - Stored Procedure Techniques

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – CoreAnalytics

- Healthcare Measures (circa 2015)
  - Clinical Quality – quality of patient care and adherence to quality standards
  - Utilization – cost or performance efficiency of a healthcare facility
- Clients were major healthcare provider systems
  - Hospital Networks with data for about 70+ hospitals
  - Hundreds of millions of patients and their clinical data
  - ~20+ Clinical Quality Measures
  - ~10+ Utilization Measures
  - Calculation time: ~1-2 hours

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – CoreAnalytics

- Measure Calculation Engine
  - Measures created/edited by non-programmers
  - Stored as XML documents
  - Compiled into stored procedures
- Client data imported from EHR/EMR systems
  - Data mapped to standardized taxonomies & terms
  - Data transformed into known data structure
    - Elements – primary data (encounter, patient, diagnosis, etc.)
    - Attributes – characteristics of Elements (additional data)
  - Primary steps:
    - ETL – Extract/Transform/Load (typically the largest portion of time)
    - Calculation ~1-2 hours
    - Result Analysis (preparation for reporting including aggregates)

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – CoreAnalytics

- What this is NOT:
  - A treatise on SQL Server internals
  - A set of step-by-step instructions
  - A tome of DBA black magic
  - An expectation that every tip will work in every situation
- What this IS:
  - Lessons learned from the trenches
  - Data architecture techniques
  - DB programming techniques
  - A good starting point in the right direction

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Tables

- Table Structure
  - It's OK (in fact, preferred) to restructure data for analysis purposes
  - The right level of normalization is subjective
  - Minimize data to essentials
    - Only what data is needed
    - Number of tables involved
    - Number and size of columns
- Table Keys
  - Use BIGINT surrogate keys
  - Avoid varchar keys
  - Avoid GUID keys (Jeff Moden disagrees)
  - Replace key text search values with numeric ID (Taxonomy Term Pair)

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Tables

- Taxonomy/Term Pair
  - Taxonomy is a coding system identification (CPT, SNOMED, etc.)
  - Each term is a value in a specific coding system (CPT: 123.45)
  - Both are typically text strings
  - Store the pairs in a table with a unique ID for each combo
  - Used the ID (TaxTermPairId) in all search criteria
- Value Set
  - A collection of Taxonomy/Term pairs uniquely named
  - Used to identify a given procedure, diagnosis, medication, etc.
  - Can be used for multiple measures

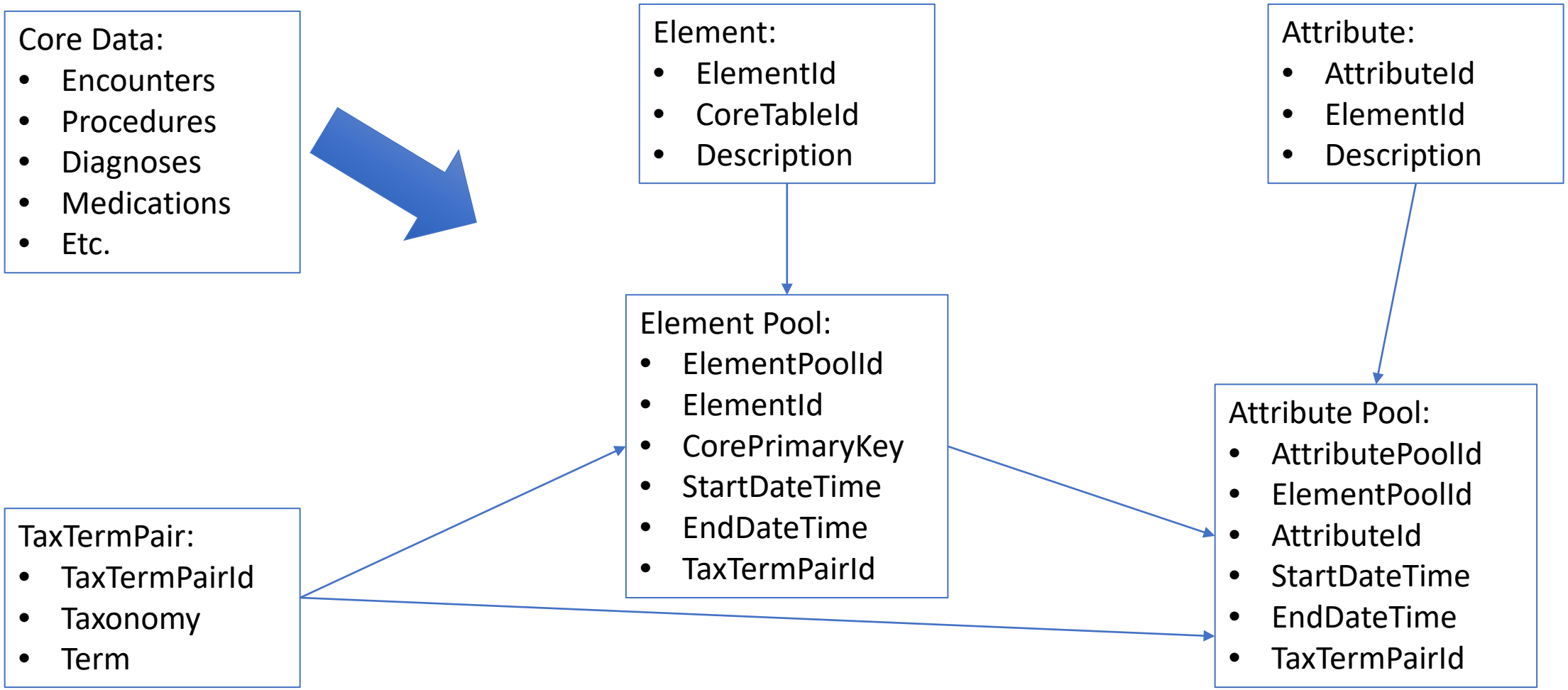
Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Tables

- Primary Data Object – Element
  - Each Element known by a specific Element Type
  - An Element may also be identified by a TaxTermPairId
  - Each Element has zero or more Attributes
  - All instances of an Element exist in the Element Pool
  - Each Element instance is identified by an ElementPoolId
- Secondary Data Object – Attribute
  - Each Attribute known by a specific Attribute Type
  - An Attribute may also be identified by a TaxTermPairId
  - An Attribute Type can be used by more than one Element Type
  - Each Attribute instance is specific to a particular Element instance

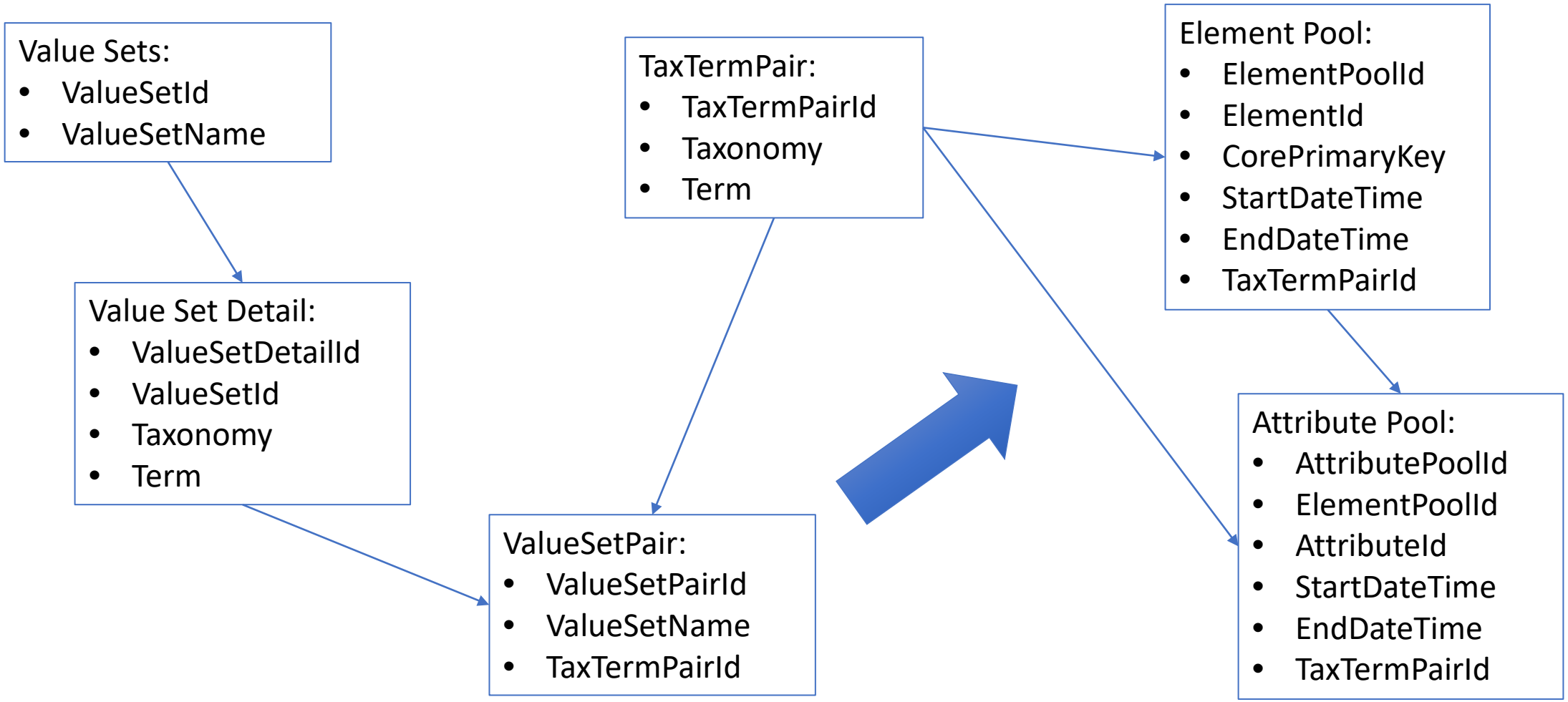
Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Tables



Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Tables



Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Queries

- Use set-based queries
  - Avoid scalar User-Defined Functions as they break set operations
  - Inline Table-Value Functions preferred
  - Avoid cursors – most can be avoided by using sets
  - Use Tally Table (or functional equivalent) to replace many instances of loops
  - Some loops can’t be avoided but OK for small iterations
- Windowing Functions
  - Your best friend by avoiding loops and cursors
  - Characteristics of GROUP BY without losing detail
- Avoid complex views
  - Optimizer can only do so much with lots of nesting
  - Hidden operations, duplicated efforts, etc.

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Queries

- Watch for Extremely Complex Queries
  - Be mindful of memory requirements with large data sets
  - Break up queries into manageable chunks
  - Take advantage of set operators: UNION (ALL), INTERSECT, EXCEPT [Bonus: Encourages parallelism]
  - Break OR conditions into separate queries then UNION results for better index use
- Avoid the use of DISTINCT in queries
  - Expensive in query processing (implicit sort)
  - Hides poorly written queries that return duplicate rows
  - Better options such as ROW_NUMBER then select row 1

Copyright © 2023, SQL RV - All Rights Reserved

Example: OR condition split

From:

```
INSERT INTO dbo.TableName (Col1, Col2, Col3)
  SELECT Col1, Col2, Col3
  FROM dbo.Table1
  WHERE (Col4 IS NULL OR Col4 = 'Value1’)
```

To:

```
INSERT INTO dbo.TableName (Col1, Col2, Col3)
  SELECT Col1, Col2, Col3
  FROM dbo.Table1
  WHERE Col4 IS NULL
UNION ALL
  SELECT Col1, Col2, Col3
  FROM dbo.Table1
  WHERE Col4 = 'Value1';
```

Big Data for SQL Server – Queries

- Table Joins
  - Rework LEFT JOINS into separate queries if possible, especially if conditional
  - Query hints *may* be needed to encourage HASH JOINs
  - Avoid the use of OUTER JOINs
    - Poor performance on large tables
    - Often pull extra data that must then be filtered
    - Rework with multiple queries UNIONed together
- Avoid calculated fields or functions in JOIN and WHERE clauses
  - Could cause indexes to not be used as expected (ex. LEFT(City,1) = 'M')
  - Causes extra processing for each calculation
  - Use pre-calculated data from a table or CTE

Copyright © 2023, SQL RV - All Rights Reserved

Example: LEFT JOIN split

From:

```
INSERT INTO dbo.TableName (Col1, Col2, Col3)
  SELECT t1.Col1, t1.Col2, t1.Col3
  FROM dbo.Table1 t1
  LEFT JOIN dbo.Table2 t2
    ON t2.Col4 = t1.Col4
    AND t2.Col5 = 'Value’
  WHERE t1.Col4 IS NULL OR t2.Col4 IS NOT NULL;
```

To:

```
INSERT INTO dbo.TableName (Col1, Col2, Col3)
  SELECT t1.Col1, t1.Col2, t1.Col3
  FROM dbo.Table1 t1
  WHERE t1.Col4 IS NULL
UNION ALL
  SELECT t1.Col1, t1.Col2, t1.Col3
  FROM dbo.Table1 t1
  INNER JOIN dbo.Table2 t2
    ON t2.Col4 = t1.Col4
    AND t2.Col5 = 'Value';
```

Big Data for SQL Server – T-SQL

- Explicit Transactions
  - Break processes into logical units of work
  - Minimally nested transactions help system resources
- Stored Procedure calls
  - Avoid deeply nested stored procs (its not C# folks!)
  - Avoid complicated parameters (memory impact)
  - Minimize number and size of OUTPUT parameters
- Execution Plans
  - Heavily dependent on data volume
  - Expect changes from dev/test to prod

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Indexes

- Partitions
  - Identify discrete, non-overlapping data
    - Segments data into pre-defined chunks
    - ElementId (encounter, diagnosis, etc.) for Element Pool
  - Allows data to be placed where best optimized
  - Improves maintenance of data (inserts/deletes)
- Proper use of indexes
  - Use covering indexes where appropriate
  - Avoid numerous indexes – not every combo is needed
  - Base indexes on JOIN criteria and WHERE clauses
  - Have plans to periodically re-evaluate index usage
  - No indexes or FK constraints on worktables

Copyright © 2023, SQL RV - All Rights Reserved

Big Data for SQL Server – Alternatives

- Columnstore Indexes
  - Great idea, but not available on production version 2008
  - “Compression” (de-duplication) makes great use of space
  - Did use a form of de-duplication with measure results
- Why not another platform?
  - Hadoop: Data is highly relational; needs updates and deletes
  - MPP: Ported to Netezza; Teradata, Redshift, or Snowflake possible
  - Multiple SQL Server instances (pre-Azure days):
    - Data could be replicated, and measures split across servers
    - Results could be combined on a reporting server
    - Overhead would probably not be worth the hassle

Copyright © 2023, SQL RV - All Rights Reserved

Closing Thoughts

- NoSQL “Big Data” environments
  - Relies upon sharding and parallelism for performance/scalability
  - Unique data considerations
    - Primarily stores and fetches data – not for updates or deletes
    - Effective on non-discrete (unstructured) data
  - No built-in query tool or language
    - Data processing is performed outside of the database
    - Allows for complex procedural algorithms and programming
    - Multiple tools available to process data
  - Not a relational database engine
    - SQL interfaces are for convenience – not designed for production
    - Tools are improving but not optimized for relational queries

Copyright © 2023, SQL RV - All Rights Reserved

Closing Thoughts

- SQL Server
  - True relational database engine
    - Designed to perform updates and deletes
    - Better performance when set-based logic is used
  - Works well with structured high-volume data
    - Requires proper architecture and tuning
    - Data movement operations should be scrutinized
  - T-SQL performs data processing close to data
  - Is limited to a single server
- Azure is a multi-server SQL Server in the cloud
- Introducing Microsoft SQL Server 2019 Big Data Clusters

Copyright © 2023, SQL RV - All Rights Reserved

Questions & Comments

BONUS:
A **TON** of free eBooks from Microsoft, RedGate, and SentryOne!

PRESENTATION FEEDBACK:
- Your thoughts needed
- Improve presentations
- Make this event even more valuable!!!

Aaron N. Cutshall

@sqlrv
www.linkedin.com/in/sqlrv
aaron@sqlrv.com
www.sqlrv.com
www.youtube.com/@sqlrv

Copyright © 2023, SQL RV - All Rights Reserved