



Handling Advanced Data Warehouse Scenarios in SSIS

John Welch, BI Architect @ Varigence

MVP – SQL Server

Agenda

- Late Arriving Dimensions
- Parent-Child Dimensions
- Type 3 and Type 6 Dimensions

About Me

- John Welch, BI Architect w/ Varigence
 - Varigence builds tools and frameworks that enable the creation and management of end-to-end business intelligence solutions with unprecedented ease and speed.
- Coauthor on
 - SQL Server MVP Deep Dives
 - Smart Business Intelligence Solutions with Microsoft SQL Server 2008
- Working with SQL Server since 1996
- SQL Server MVP
- My Contact Info
 - <http://agilebi.com/cs/blogs/jwelch>
 - Twitter: @john_welch
 - Email: john.welch@varigence.com

About This Presentation

- It's all about patterns
 - Patterns (like rules) are made to be broken
 - These are general outlines – you will need to modify them to fit specific scenarios
- The patterns are biased toward SSIS processing
 - Avoids tying the pattern to a specific data source



LATE ARRIVING DIMENSIONS

Scenario

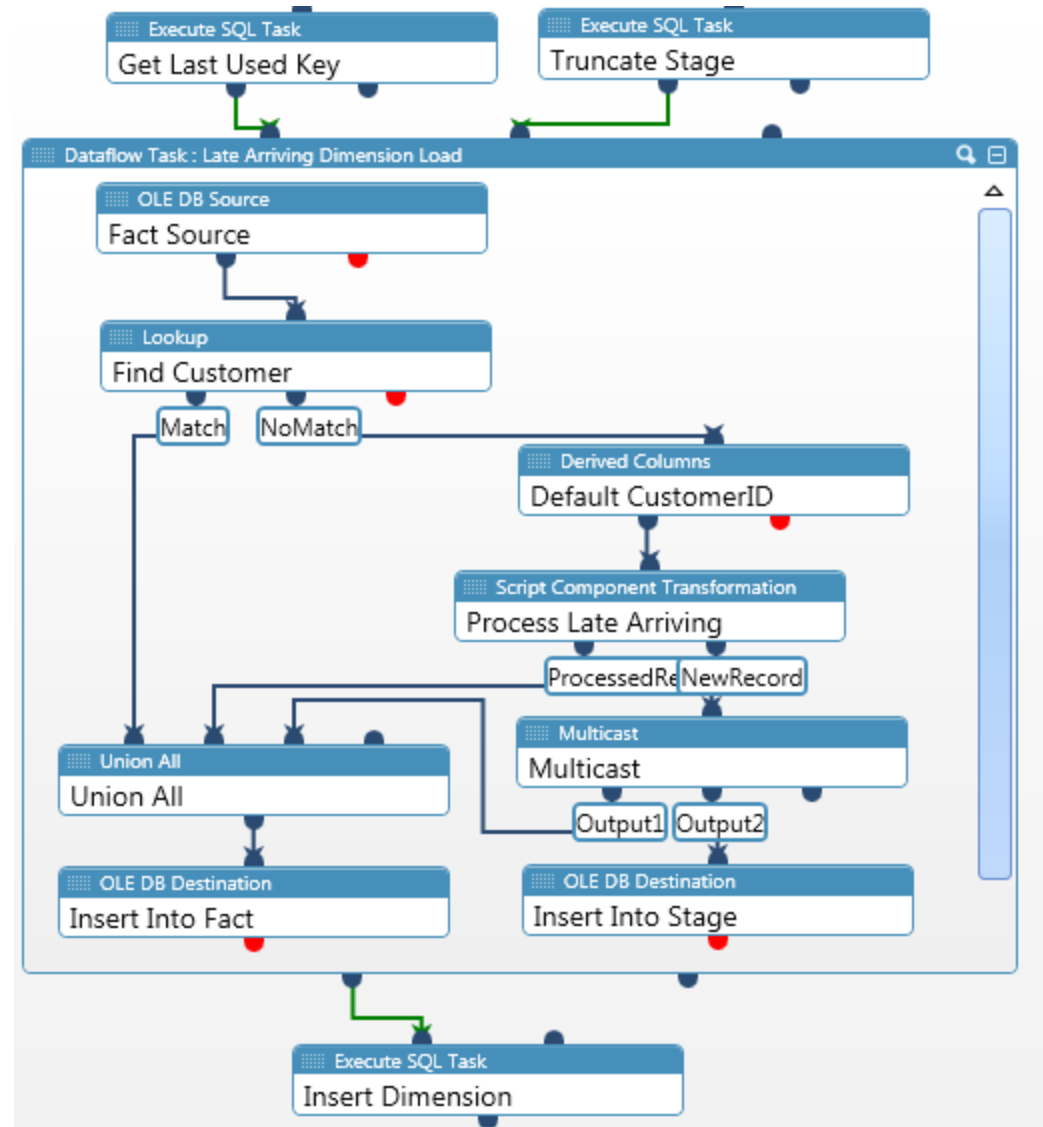
- Fact rows arrive before the dimensions rows
 - We need to record the fact data, but we don't have complete dimension data
- Constraints
 - Needs to perform well with high volumes of data
 - The dimension data will arrive eventually
 - The natural key represents a “real”, new dimension member

Potential Patterns

- Dimension Handling - Use the SCD Component's inferred member support
 - Quick and easy to set up
 - Performance is not great, and maintenance can be problematic
- Fact Handling - Use an OLE DB Command to insert the new row
 - Easy to set up, has performance problems
 - What if there are multiple instances of the new dimension value?

Recommended Pattern (Late Arriving)

- Does It Exist In Dimension?
 - Yes: Return ID
 - No: Continue
- Have we seen it before?
 - Yes: Get ID
 - No: Assign new ID
- Insert Fact Record
- Insert Dimension Record





PARENT-CHILD DIMENSIONS

Problem

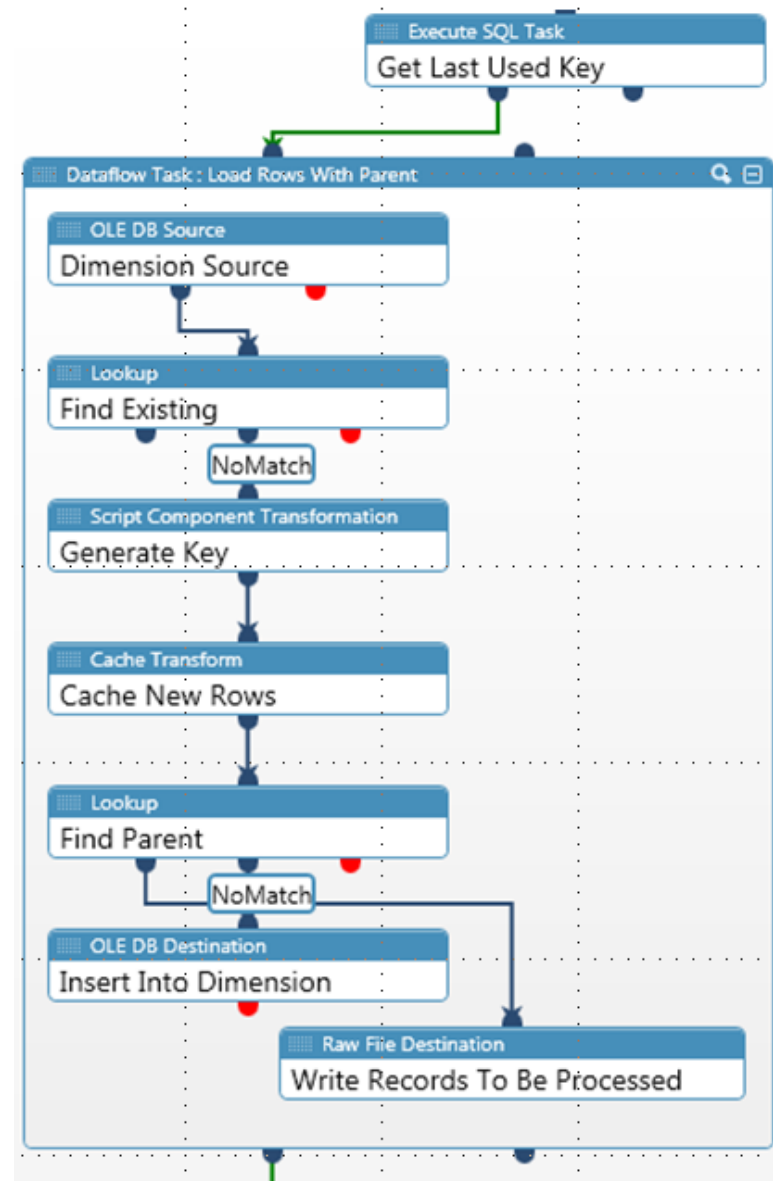
- The data represents a recursive hierarchy of information
 - Employees, Subsidiaries, etc.
 - A given rows needs to reference another row in the same table
- Constraints
 - Parent rows need to be inserted before child rows
 - Child rows have reliable references to parents
 - Needs to perform well

Potential Patterns

- Use SQL to order and determine level
 - Use CTEs, stored procs, etc
 - Doesn't work with non-SQL sources
- Loop over rows, inserting individually
 - RBAR mode – painfully slow

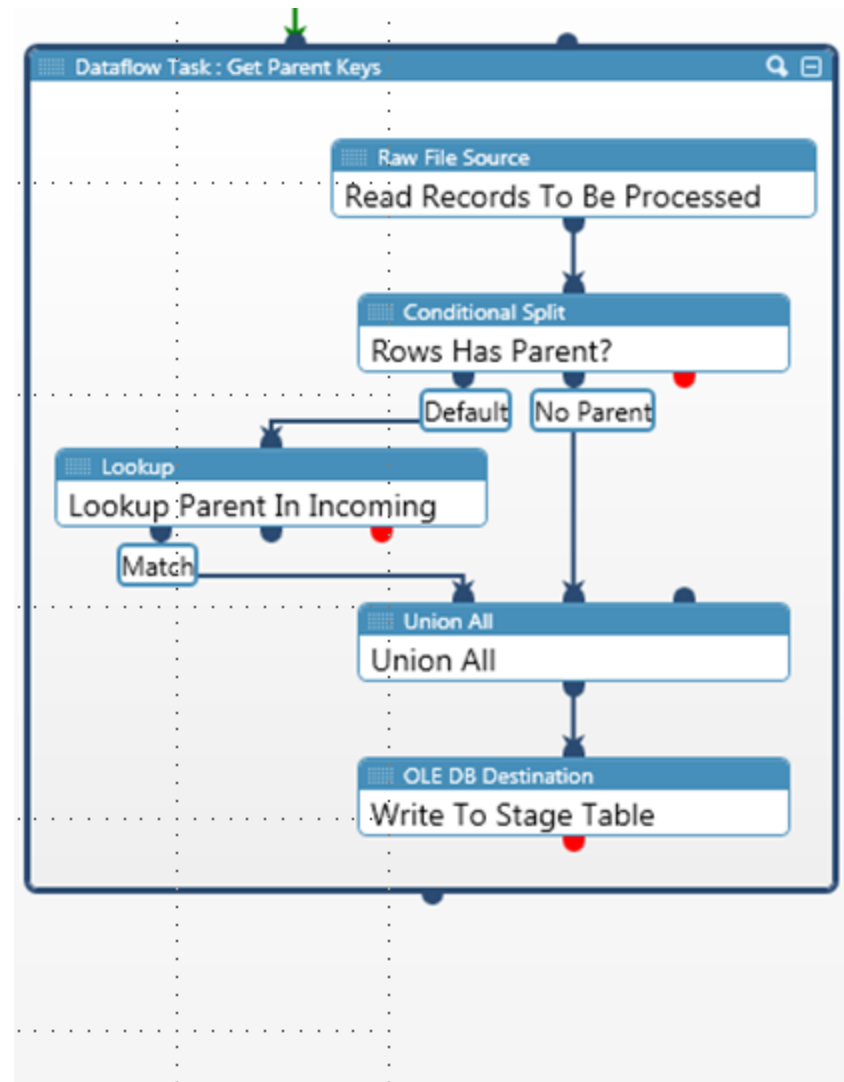
Recommended Pattern (Parent-Child Pt. 1)

- Does It Exist?
 - Yes: discard
 - No: continue
- Assign ID
- Cache New Rows
- Parent Exists?
 - Yes: Insert
 - No: Write to temp file



Recommended Pattern (Parent-Child Pt. 2)

- Read temp file
- Row Has Parent?
 - No: Write to Destination
 - Yes: Look for parent in new row cache
- Parent Exists in New Row Cache?
 - No: Discard
 - Yes: Write To Destination





TYPE 3 AND TYPE 6 DIMENSIONS

Dimension Types Defined

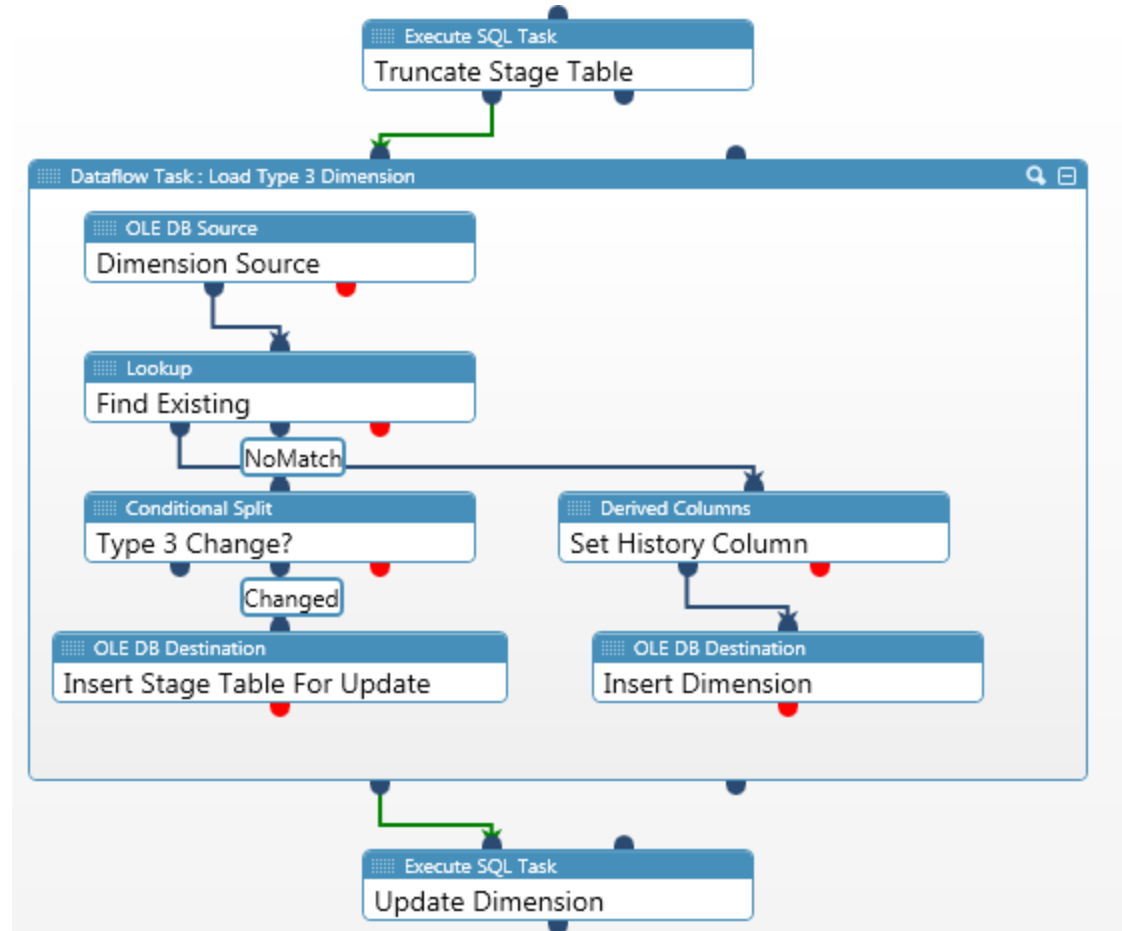
- Type 1 – changes are written in place
 - Tracks most recent value
- Type 2 – changes are recorded in a new row
 - Tracks historical values
- Type 3 – changes are recorded in a new column
 - Tracks limited historical and current value
- Type 6 – Type 1 + 2 + 3
 - Combines the tracking of all the above

Problem

- You need to track historical dimension changes while providing the ability to report history according to current values
 - Product changes categories, want to report sales both by historical category and by current category
- Constraints
 - Need to have understandable logic
 - Have differing update logic depending on scenario

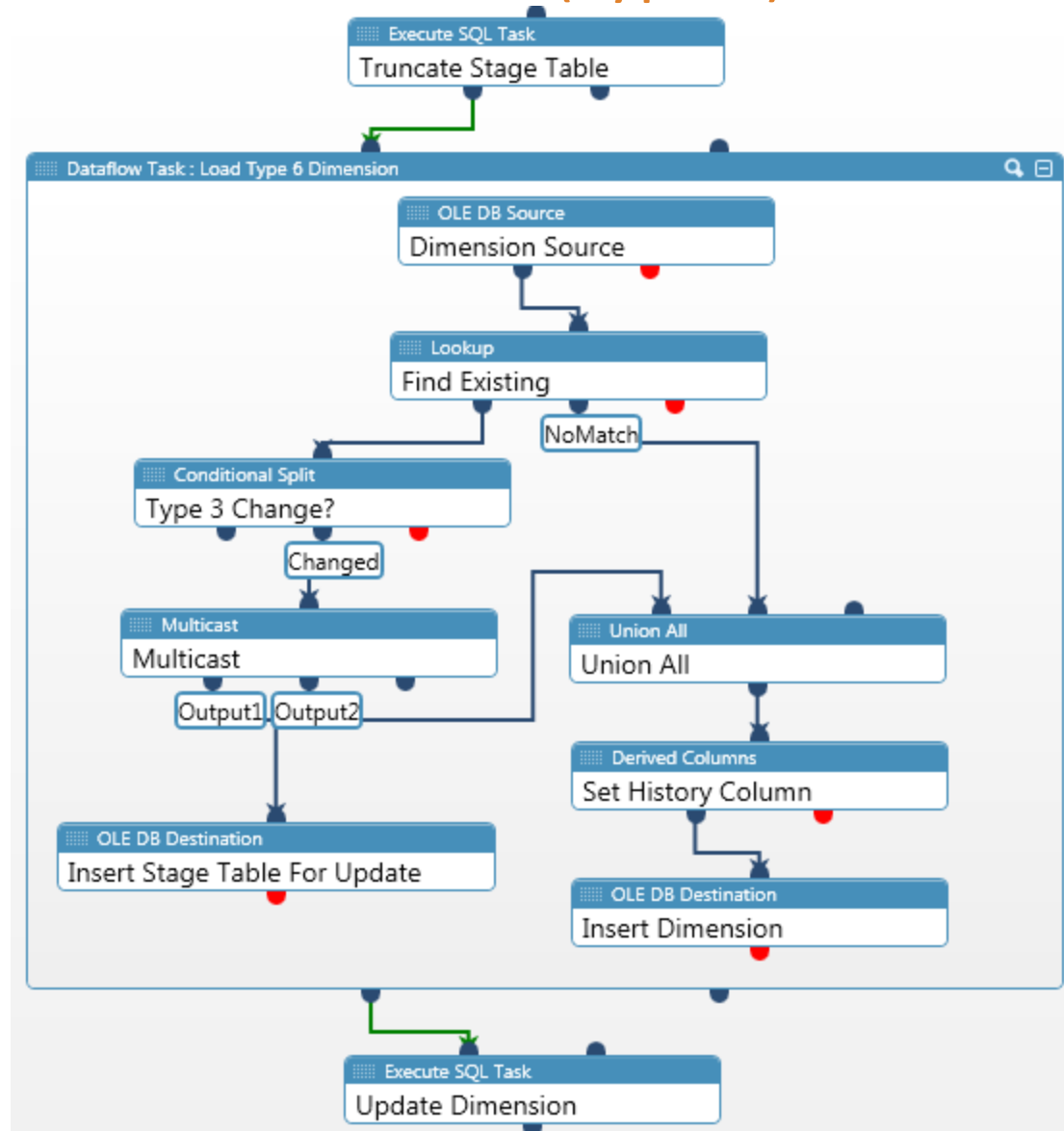
Recommended Pattern (Type 3)

- Rows Exists?
 - No: Set History and Insert
 - Yes: Check for Change
- Type 3 Change?
 - No: Discard
 - Yes: Update Dimension



Recommended Pattern (Type 6)

- Rows Exists?
 - No: Set History and Insert
 - Yes: Check for Change
- Type 3 Change?
 - No: Discard
 - Yes: Continue
- Update Dimension History
- Insert Dimension



Wrap Up

- Patterns are guidelines
- Consider different approaches when your DBMS platform permits
- Standardize patterns wherever possible

Resources

- SSIS MSDN Developer Center
 - <http://msdn.microsoft.com/en-us/sqlserver/cc511477.aspx>
 - Performance Tuning - <http://technet.microsoft.com/en-us/library/cc966529.aspx>
- My Blog
 - <http://agilebi.com/cs/blogs/jwelch>
 - Twitter: @john_welch
- MSDN Forums
 - <http://social.msdn.microsoft.com/forums/en-US/sqlintegrationservices/threads/>



Thanks for Attending

Any questions?

john.welch@varigence.com