



Design Studio

Data Flow performance optimization



Performance optimization

- Overview
- Plan sources
- Plan sinks
- Plan sorts
- Example
- New Features
- Summary
- Exercises



Performance optimization

- **Introduction**
- Plan sources
- Plan sinks
- Plan sorts
- Example
- New Features
- Summary
- Exercises



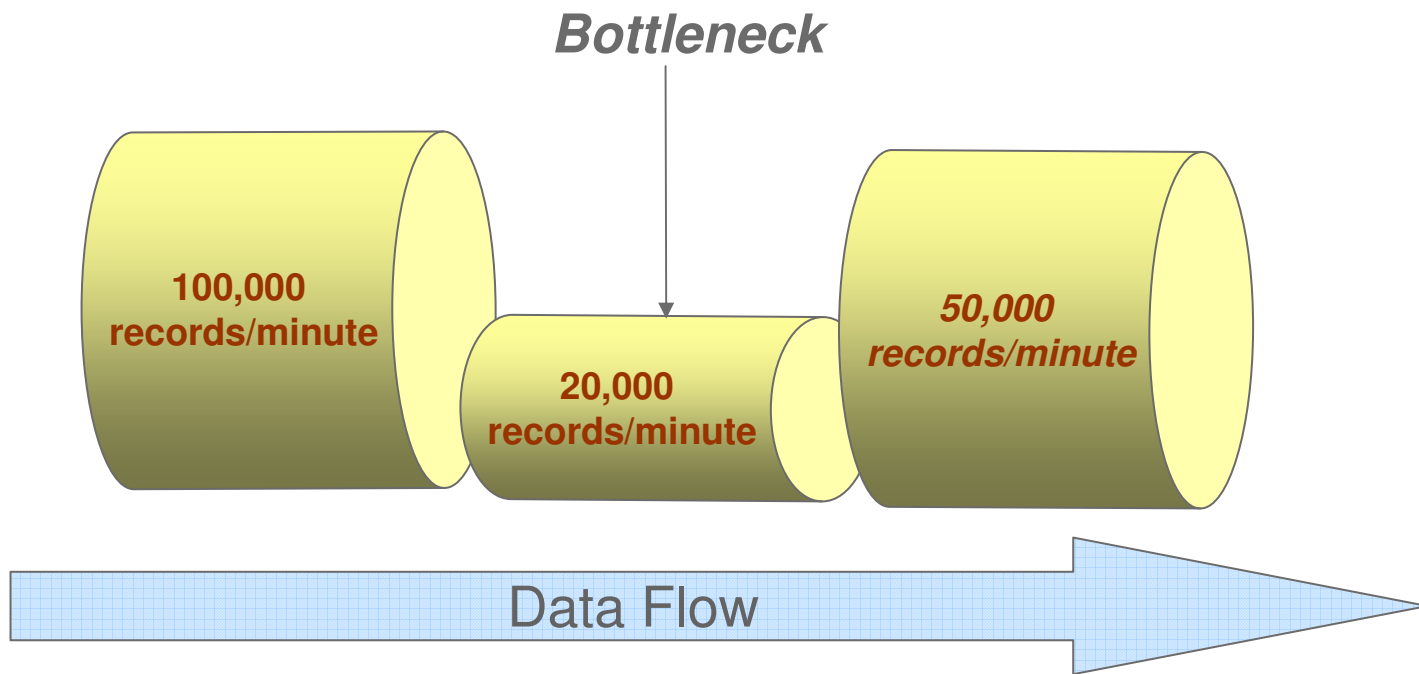
Performance optimization

Introduction

- Design Studio = Development Tool
As with other application development, ETL development should follow a methodology which includes tasks for both function and efficiency testing
- Data Flow Server works with a Pipeline Architecture
For data load plans processing time is dependent on the rate that the data can flow through this pipeline

Performance optimization

Introduction – Pipeline Architecture





Performance optimization

Introduction – Analyzing a plan

Look for the three “S”:

- Sources – File Transforms and SQL Queries
- Sinks – File Sinks, SQL Sinks and Batch Loaders
- Sorts – Sort Transforms, Join, Pivot, SubPlans, Sorted Union and Rank Transforms

Performance optimization

- Introduction
- **Plan sources**
- Plan sinks
- Plan sorts
- Example
- New Features
- Summary
- Exercises





Performance optimization

Plan sources – Overview

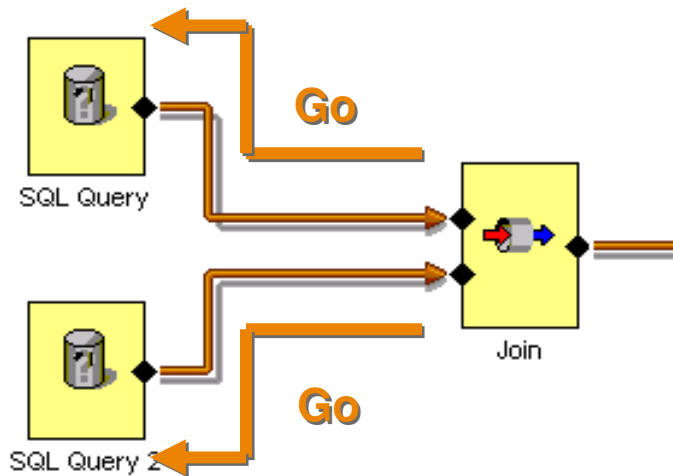
Begin by looking at the performance of Plan sources:

- Use a terminal sink or monitor transform to look at source transforms performance
- If the source transform takes a large percentage of the overall Plan execution time, look at ways to speed up the source:
 - Modifying the SQL or adding indexing to tables.
 - Using multiple source transforms to read the source in parts.
 - Reduce the overhead of translating data from file sources.

Performance optimization

Plan sources – Sort related transforms 1

Try and push sorts back to the source whenever possible!

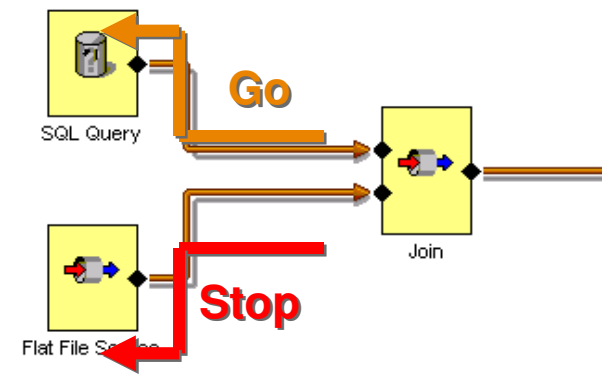


In some cases, Sagent can push sorting back to the source on its own.

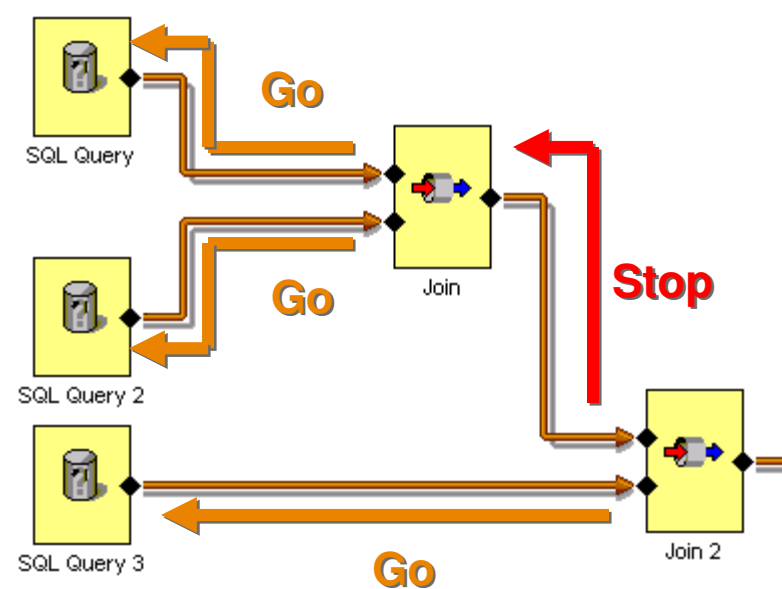


Performance optimization

Plan sources – Sort related transforms 2



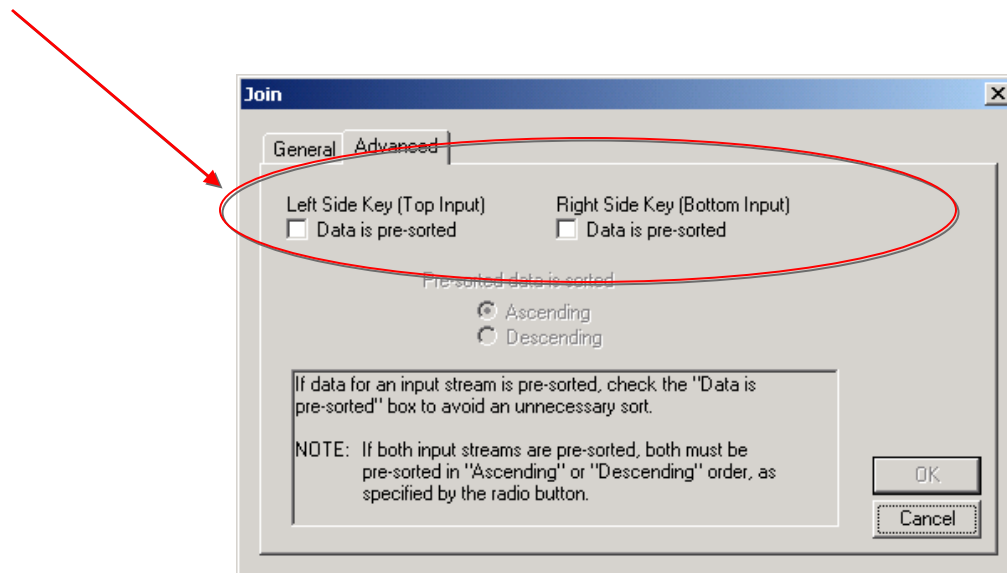
In other cases Sagent can't push sorting back to the source automatically.



Performance optimization

Plan sources – Sort related transforms 3

- Analyze the plan to see if the blocked input is going to be in proper sort order.
- Use the “Data is pre-sorted” feature to override Sagent disk sorts.





Performance optimization

- Introduction
- Plan sources
- **Plan sinks**
- Plan sorts
- Example
- New Features
- Summary
- Exercises



Performance optimization

Plan sinks – Overview

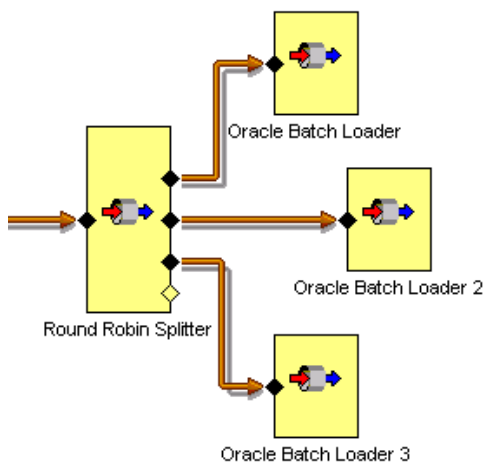
- Use batch loaders where appropriate
- Parallelize the sinks where possible to improve performance
 - Use Round Robin Splitter or segmented Data Flows to achieve parallelism



Performance optimization

Plan sinks – Round Robin Splitter

- Purpose:
The Round Robin Splitter provides a way to distribute data in the data flow for processing more efficiently. It divides up the data and the processing so that multiple sets can be processed at the same time





Performance optimization

Plan sinks – Database

- Use existing database features wherever feasible (e.g. Oracle direct path loading)
- Use RAID 0 where appropriate for temporary files to improve performance
- Make sure, the hardware of the database server matches your requirements



Performance optimization

Plan sinks – Other items to know

- Reduce the impact of copy transforms by reusing existing columns
- Use Round Robin Splitter with intermediate transforms such as Pivot and Expression Calculator
- Increasing Data Flow Block Sizes can affect performance in some cases
- Leave Key Lookups for the end of the plan



Performance optimization

- Introduction
- Plan sources
- Plan sinks
- **Plan sorts**
- Example
- New Features
- Summary
- Exercises

Performance optimization

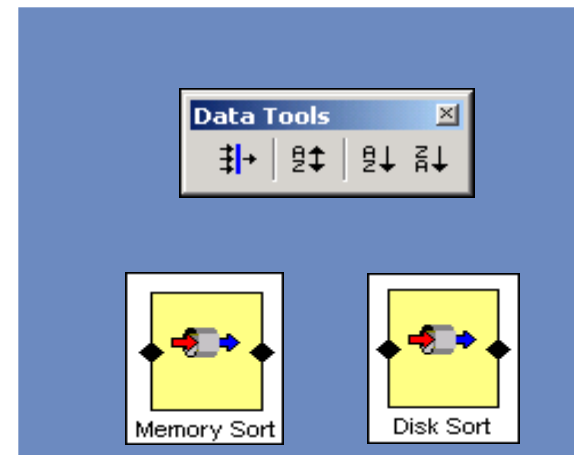
Plan sorts - Overview

Sorting is any process of arranging items in some sequence and/or in different sets, and accordingly, it has two common, yet distinct meanings:

- ordering: arranging items of the same kind, class, nature, etc. in some ordered sequence
- categorizing: grouping and labeling items with similar properties together (by sorts)

Sort - to put in a certain place or rank according to kind, class, or nature.

- Merriam Webster Online





Performance optimization

Plan sorts - User specified Sorts

Sequences the input records based on a specified column or set of columns:

- SQL Query
 - Uses resources on the RDBMS
- Memory Sort
 - Uses memory resources on the Data Flow Server. Use for small amounts of data
- Disk Sort
 - Creates and uses temporary files stored on the Data Flow Server. Use for large amounts of data



Performance optimization

Plan sorts - Internal Sorts

- Sequences the input records based on columns which are determined by Transform specifications, not user-defined sort parameters.
- Unless otherwise specified by the user, the Data Flow will perform an internal sort when executing these transforms:
 - Join
 - Comparison
 - Sorted Union
 - Pivot
 - Rank
- The user has the option to override the Internal Sort on all of these transforms.
- Use the Advanced tab to avoid sorting data that is already sorted.
- Eliminating unnecessary sorting can increase performance.



Performance optimization

Plan sorts - Sort Pushback 1

- Each database system has its own way of sorting data internally
- Sybase and MS SQL Server use the same sorting mechanism
- Sagent Dataflow uses the American way of sorting. Make sure your database clients are set to American in an Oracle environment
- There is no universal sort mechanism, even in one country there are different ways of sorting.
- In case of doubt let dataflow do the sorting rather than the database system



Performance optimization

Plan sorts - Sort Pushback 2

- In order to optimize plan performance, Data Flow will perform (if possible) what is known as a “sort pushback” to the RDBMS whenever there is a sort required by the plan
- The main attributes of sort pushback prior to version 5.5 are:
 - If Data Flow can “push” a sort onto the SQL Query step, it will do so
 - If Data Flow cannot push the sort back to the SQL Query step and all the sinks in the plan are “Client Sinks” (i.e.; display sinks), Data Flow will perform an internal memory sort
 - If Data Flow cannot push the sort back to the SQL Query step and at least one of the sinks in the plan are “Non-Client Sinks” (i.e.; non-display sinks), Data Flow will perform an automatic disk sort



Performance optimization

Plan sorts - Sort Pushback 4

- **Sort pushback is attempted with the following transforms:**
 - Join
 - Comparison
 - Sorted Union
 - Pivot
 - Rank
- **Use the Advanced tab to bypass all sorts, internal or pushback**
 - If the data are already sorted, you can ask Data Flow to bypass a sort pushback or an internal sort by specifying that the data are pre-sorted in the Advanced tab of these transforms



Performance optimization

Plan sorts - Sort Pushback 5

- **Data Flow will attempt to pushback a sort to the RDBMS if:**
 - The SQL Query does not already contain an ORDER BY clause
 - The SQL Query is not user-entered SQL
 - No intermediate step alters any of the sort fields
 - No intermediate step requires a sort of its own
 - No intermediate step adds or deletes records
 - No sort field originated in an intermediate step
 - No intermediate step has multiple inputs or outputs

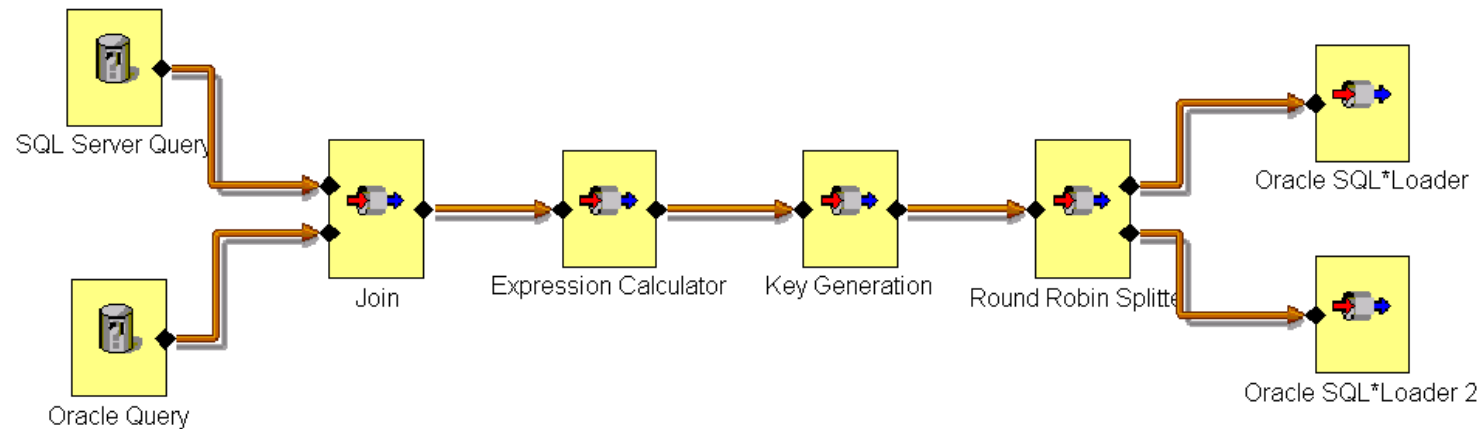


Performance optimization

- › Introduction
- › Plan sources
- › Plan sinks
- › Plan sorts
- **Example**
- › New Features
- › Summary
- › Exercises

Performance optimization

Example – Introduction



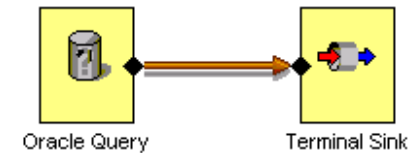
Data Load Plan

- **Read data from two disparate data sources**
- **Transform data according to business rules**
- **Add a unique record identifier**
- **Load a dimension in the data mart**
- **Plan taking over 4 hours to complete**

“Why doesn’t the Round Robin Splitter improve Performance?”

Performance optimization

Example – Steps to analyze plan

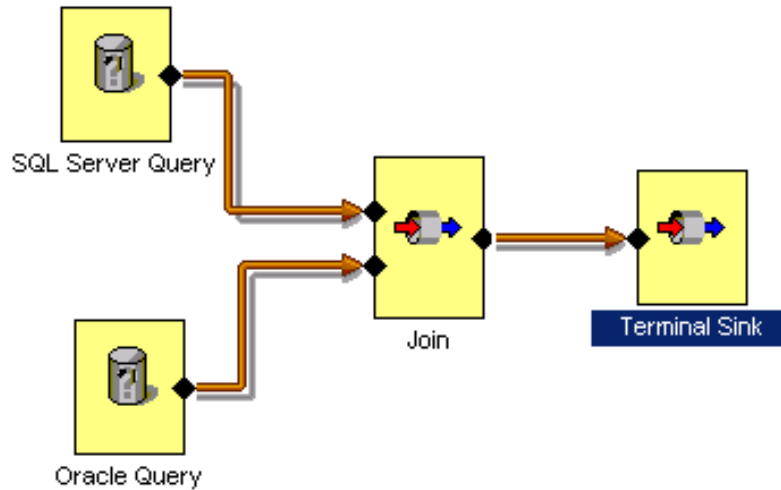


Start at target and test for processing time of each query using terminal sink transform

- Oracle query step took almost 4 hours
- Indexing was added to reduce Oracle query to less than 20 minutes (~42,000 rec/min)

Performance optimization

Example – Analyze step by step

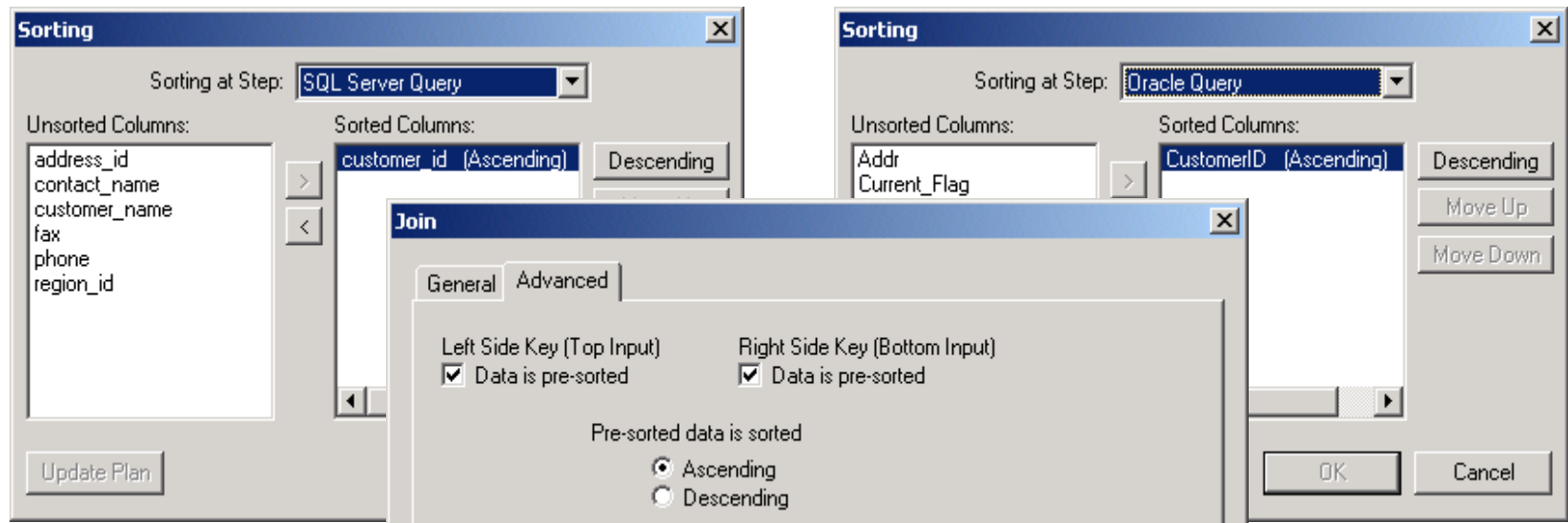


Move the Terminal Sink after the Join Transform

- Completes in 65 minutes
- Join transform incrementally increases processing time by 45 minutes

Performance optimization

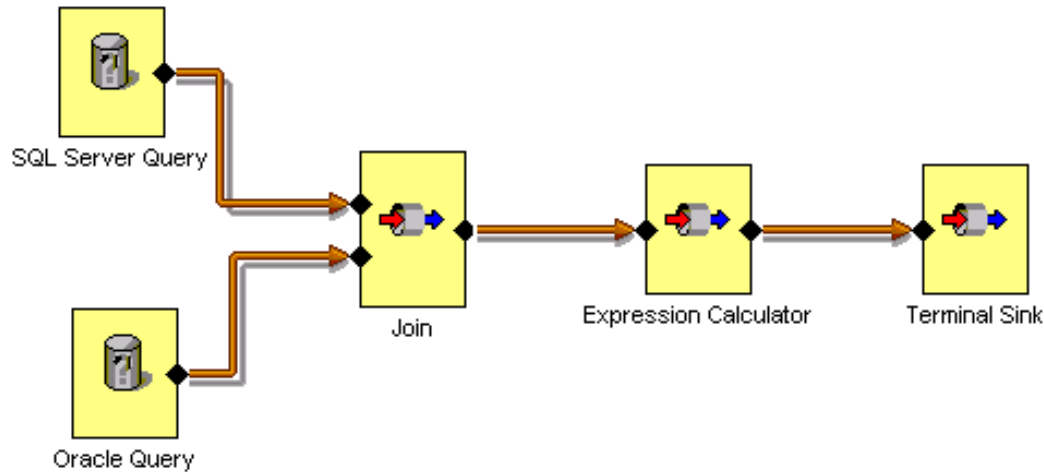
Example – Pre-Sort both data sets



- **Sort both Source Queries by common data field(s)**
- **Update Join transform to indicate that both sets of data are presorted**
 - Eliminate automatic memory sort of both sets of data
 - Plan now completes in 40 minutes (reducing this portion of processing by 25 minutes)

Performance optimization

Example – Analyze step by step



Move the Terminal Sink after the Expression Calculator Transform

- Completes in 50 minutes
- Expression Calculator incrementally increases processing time by 10 minutes

Performance optimization

Example – Update data in flow

The image displays two side-by-side screenshots of the 'Expression Calculator' dialog box, illustrating a performance optimization technique. Both screenshots show the same input and output columns, with the 'Include New Columns by Default' checkbox checked. The 'Expressions' table in both is identical, showing a single expression: 'New Name' (string, length 10) with the expression 'ToUpper (Name)'. The key difference is in the 'Output Column' settings at the bottom right. In the left screenshot, the 'Replace Existing' checkbox is unchecked. In the right screenshot, the 'Replace Existing' checkbox is checked, and the 'Name' dropdown menu is highlighted with a red box, indicating that the existing column is being updated instead of a new one being added.

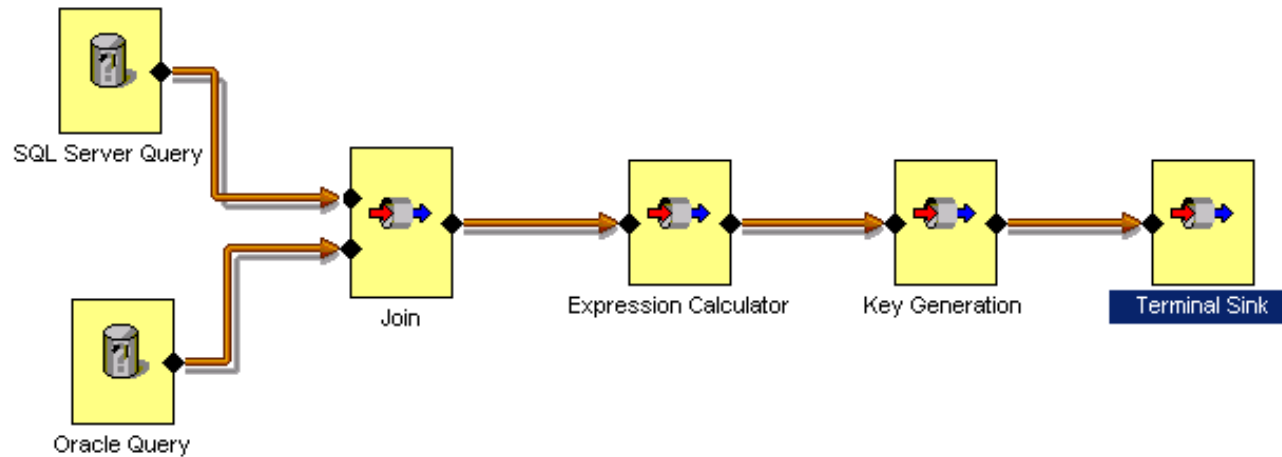
Output Column	Type	Length	Expression
New Name	string	10	ToUpper (Name)

Update an existing column rather than adding a new column

- Plan now completes in 45 minutes (reducing this portion of processing by 5 minutes)

Performance optimization

Example – Continue step by step analysis

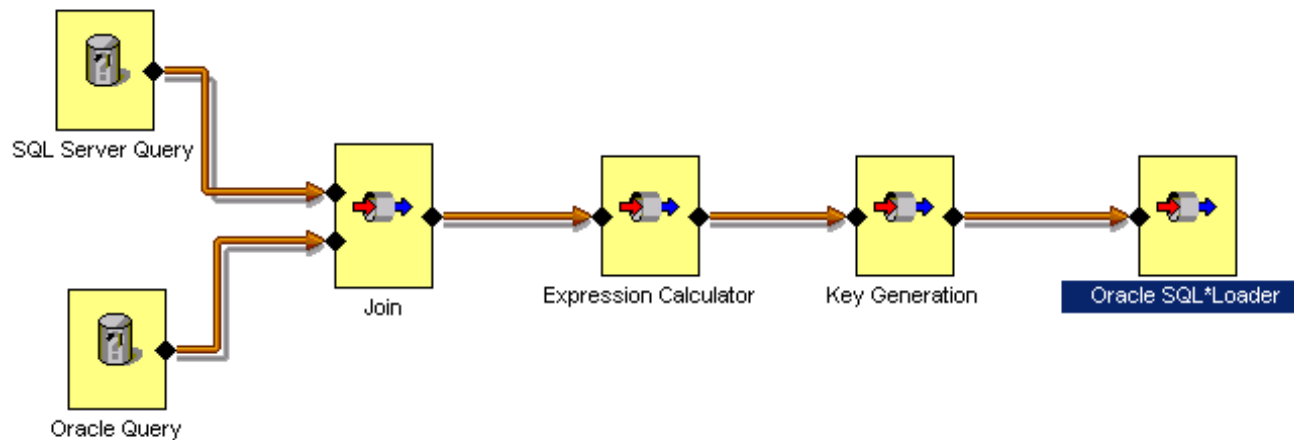


Move the Terminal Sink after the Key Generation Transform

- Completes in 50 minutes
- Key Generation incrementally increases processing time by 5 minutes

Performance optimization

Example – Load the data

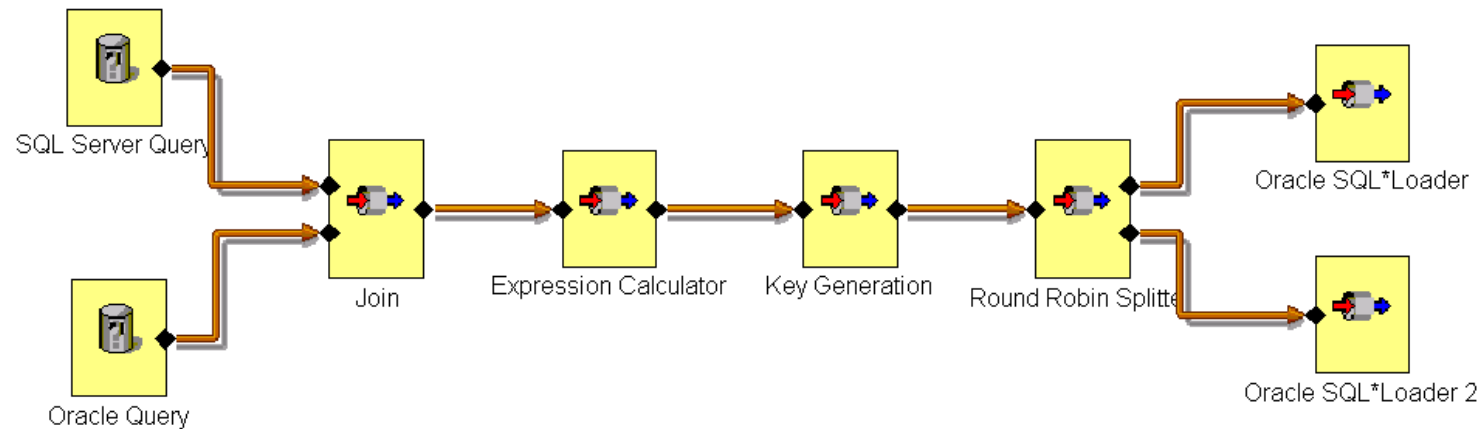


Add Oracle SQL Loader to update the data mart

- Completes in just under 3 hours (180 minutes)
- SQL Loader incrementally increases processing time by 2 hours
- Loads 1M records in 120 minutes (~8,300 rec/min)

Performance optimization

Example – Add additional loaders



Add Round Robin Splitter and additional Oracle SQL Loader

- Completes in just under 2 hours (120 minutes)
- Two SQL Loaders incrementally increases processing time by 1 hour
- Loads 1M records in 60 minutes (~16,700 rec/min)
- Continue to add Loaders until you reach a diminishing rate of return

Performance optimization

- Introduction
- Plan sources
- Plan sinks
- Plan sorts
- Example
- **New Features**
- Summary
- Exercises

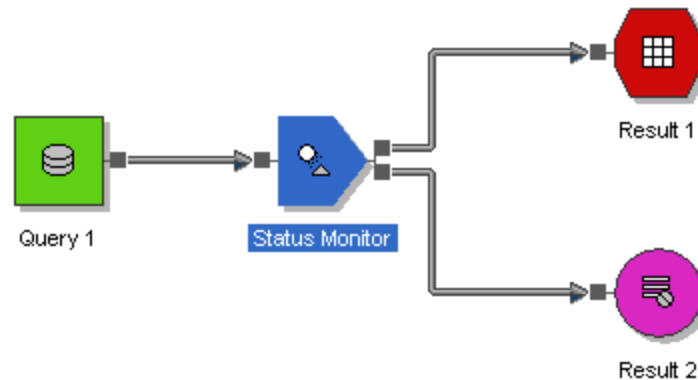




Performance optimization

New Features – Status Monitor 1

- Purpose:
 - Use the Status Monitor Transform to track processing performance in a data flow.



Performance optimization

New Features – Status Monitor 2

- Parameters – Record interval:

The screenshot shows the 'Status Monitor' dialog box with three callout boxes pointing to specific settings:

- Select Interval Type:** Points to the 'Records' radio button in the 'Select Interval Type' section.
- Select the count value:** Points to the 'TotalRecs' text box in the 'Summary Counts' section.
- Select the summary statistics:** Points to the 'SampleAvg' text box in the 'Summary Statistics' section.

The dialog box contains the following sections and controls:

- Select Interval Type:** Radio buttons for 'Records' (selected) and 'Time (Sec)'. A 'Record Interval' spinner box is set to 500.
- Summary Counts:** A checkbox for 'Records Processed'. Below it, 'Total' and 'Per Interval' labels with corresponding text boxes: 'TotalRecs' and 'SampleRecs'.
- Processing Time:** A checkbox for 'Processing Time'. Below it, 'Total' and 'Per Interval' labels with corresponding text boxes: 'TotalTime' and 'SampleTime'.
- Summary Statistics:** Checkboxes for 'Avg Processing Time' and 'Min/Max Processing Time'. Below them are text boxes for 'SampleAvg', 'SampleMin', and 'SampleMax'. A checkbox for 'Exclude First Sample From Summary Statistics' is checked.
- Buttons:** 'OK' and 'Cancel' buttons at the bottom.

Performance optimization

New Features – Status Monitor 3

- Parameters – Time interval:

The screenshot shows the 'Status Monitor' dialog box with three callout boxes pointing to specific settings:

- Select Interval Type:** Points to the 'Time (Sec)' radio button, which is selected. The 'Time Interval (Sec)' spinner is set to 1.
- Select the count value:** Points to the 'Records Processed' section, specifically the 'TotalRecs' and 'SampleRecs' input fields.
- Select the summary statistics:** Points to the 'Summary Statistics' section, specifically the 'SampleAvg', 'SampleMin', and 'SampleMax' input fields.

The dialog box contains the following sections and options:

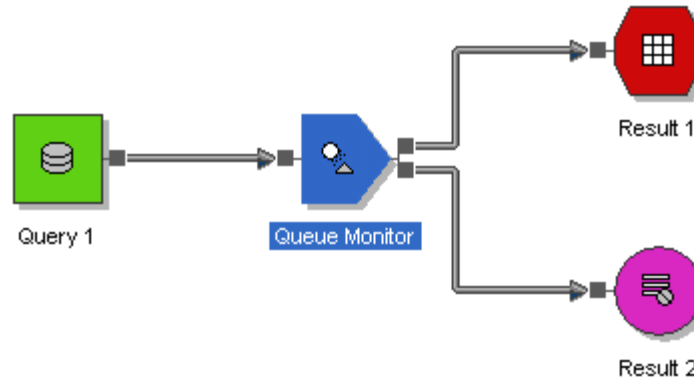
- Select Interval Type:** Records, Time (Sec). Time Interval (Sec): 1.
- Summary Counts:**
 - Records Processed: Total (TotalRecs), Per Interval (SampleRecs).
 - Processing Time: Total (TotalTime), Per Interval (SampleTime).
- Summary Statistics:**
 - Avg Records Processed: SampleAvg.
 - Min/Max Records Processed: Min Records (SampleMin), Max Records (SampleMax).
 - Exclude First Sample From Summary Statistics.

Buttons: OK, Cancel.

Performance optimization

New Features – Queue Monitor

- Purpose:
 - Use Queue Monitor to identify performance bottlenecks in data flows

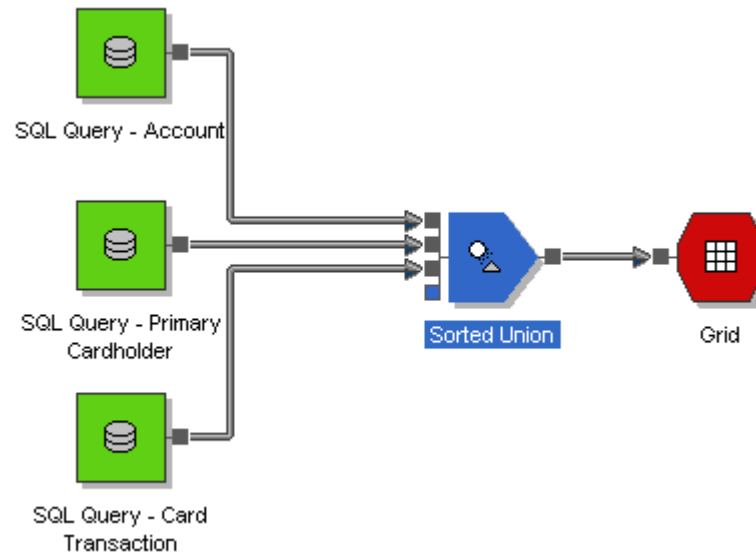


- Parameters:
 - Set the sampling interval in seconds.

Performance optimization

New Features – Sorted Union 1

- Purpose:
 - Use the Sorted Union Transform to produce a single result set by merging rows from two or more data flows.

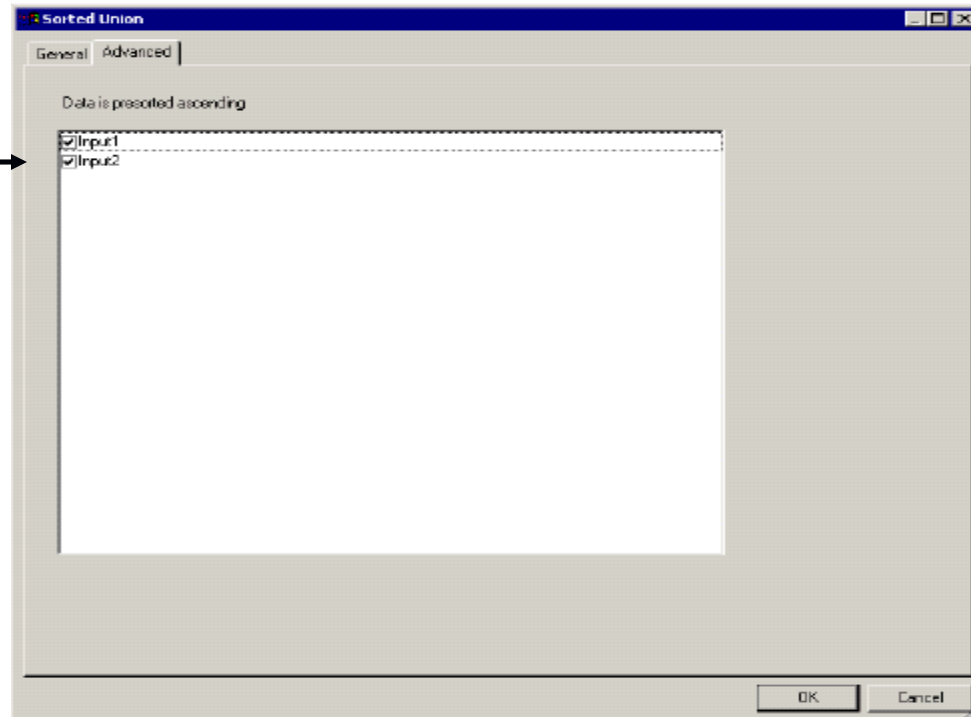


Sorting and Data Flow

New Features – Sorted Union 2

Advanced Tab:

Select the check box for any input source that is already sorted in ascending order.



- If you sort your data with an Order By clause, Disk Sort or Memory Sort before unioning, you must sort in ascending order



Performance optimization

- Introduction
- Plan sources
- Plan sinks
- Plan sorts
- Example
- New Features
- **Summary**
- Exercises



Performance optimization

Summary

- Test! Test! Test!
- Push bottlenecks to target
- Be aware and take advantage of new features designed to improve efficiency



Performance optimization

- Introduction
- Plan sources
- Plan sinks
- Plan sorts
- Example
- New Features
- Summary
- **Exercises**