Aggreagator transformation:

Sorted Input Conditions

Do not use sorted input if either of the following conditions are true:

•The aggregate expression uses nested aggregate functions.

•The session uses incremental aggregation.

Source data is data driven.

what is incremental aggregation in aggregate transformation?

Data must be sorted in the following ways:

By the Aggregator group by ports, in the order they appear in the Aggregator transformation.•

Using the same sort order configured for the session.

If data is not in strict ascending or descending order based on the session sort order, the Integration Service fails the session.

For example, if you configure a session to use a French sort order, data passing into the Aggregator transformation must be sorted using the French sort order.

If the session uses relational sources, you can also use the Number of Sorted Ports

option in the Source Qualifier transformation to sort group by columns in the source database.

Tips:

Use sorted input to decrease the use of aggregate caches.

The Aggregator transformation might not provide sorted output.

---To sort output from an Aggregator transformation, use a Sorter transformation.

Limit connected input/output or output ports.

Filter the data before aggregating it.

Scenarios to be looked into---

Expression Transformation:

Need to look some scenarios--

Filter transformation:

Active transformation.

You cannot concatenate ports from more than one transformation into the Filter transformation.

The input ports for the filter must come from a single transformation.

Tip: Place the Filter transformation as close to the sources in the mapping as possible to maximize session performance.

Rather than passing rows you plan to discard through the mapping,

you can filter out unwanted data early in the flow of data from sources to targets.

Tips for Filter Transformations:

Use the Filter transformation early in the mapping.

Use the Source Qualifier transformation to filter.

Filter transformation for sources other than relational.

Joiner Transformation:

Active transformation.

Use the Joiner transformation to join source data from two related heterogeneous sources residing in different locations or file systems.

The two input pipelines include a master pipeline and a detail pipeline or a master and a detail branch.

master pipeline ends at the Joiner transformation, while the detail pipeline continues to the target.

Limitations:

You cannot use a Joiner transformation when either input pipeline contains an Update Strategy transformation.

You cannot use a Joiner transformation if you connect a Sequence Generator transformation directly before the Joiner transformation.

If you use multiple ports in the join condition, the Integration Service compares the ports in the order you specify.

Both ports in a condition must have the same datatype

Note: The Joiner transformation does not match null values.

For example, if both EMP\_ID1 and EMP\_ID2 contain a row with a null value,

the Integration Service does not consider them a match and does not join the two rows.

To join rows with null values, replace null input with default values, and then join on the default values.

Normal

Master Outer

Detail Outer

Full OuterNote

With a normal join, the Integration Service discards all rows of data from the master and detail source that do not match, based on the condition.

A master outer join keeps all rows of data from the detail source and the matching rows from the master source. It discards the unmatched rows from the master source.

A detail outer join keeps all rows of data from the master source and the matching rows from the detail source. It discards the unmatched rows from the detail source.

A full outer join keeps all rows of data from both the master and detail sources.

You can improve session performance by configuring the Joiner transformation to use sorted input.

To improve performance for an unsorted Joiner transformation, use the source with fewer rows as the master source.

To improve performance for a sorted Joiner transformation, use the source with fewer duplicate key values as the master.

Tips for Joiner Transformations

Perform joins in a database when possible

Join sorted data when possible.

For an unsorted Joiner transformation, designate the source with fewer rows as the master source.

For a sorted Joiner transformation, designate the source with fewer duplicate key values as the master source.

Lookup Transformation

The Lookup transformation can be an active or passive transformation.

You can configure the Lookup transformation to return a single row or multiple rows.

You can configure a connected or unconnected Lookup transformation.

Connected or unconnected lookup.

A connected Lookup transformation receives source data, performs a lookup, and returns data to the pipeline.

An unconnected Lookup transformation is not connected to a source or target.

A transformation in the pipeline calls the Lookup transformation with a :LKP expression.

The unconnected Lookup transformation returns one column to the calling transformation.

Use the following options with flat file lookups:

Use indirect files as lookup sources by configuring a file list as the lookup file name.

Use sorted input for the lookup.

Sort null data high or low.

Use case-sensitive string comparison with flat file lookups.

Using Sorted Input.

When you configure a flat file Lookup transformation for sorted input, the condition columns must be grouped.

If the condition columns are not grouped, the Lookup transformation returns incorrect results.

For optimal caching performance, sort the condition columns.