informatica Marketplace

Partition Types

Overview

The PowerCenter Integration Services creates a default partition type at each partition point. If you have the Partitioning option, you can change the partition type. The partition type controls how the PowerCenter Integration Service distributes data among partitions at partition points. When you configure the partitioning information for a pipeline, you must define a partition type at each partition point in the pipeline. The partition type determines how the PowerCenter Integration Service redistributes data across partition points.

You can define the following partition types in the Workflow Manager:

- **Database partitioning.** The PowerCenter Integration Service queries the IBM DB2 or Oracle system for table partition information. It reads partitioned data from the corresponding nodes in the database. Use database partitioning with Oracle or IBM DB2 source instances on a multi-node table space. Use database partitioning with DB2 targets.
- **Hash partitioning.** Use hash partitioning when you want the PowerCenter Integration Service to distribute rows to the partitions by group. For example, you need to sort items by item ID, but you do not know how many items have a particular ID number.

  You can use the following types of hash partitioning:

  - **Hash auto-keys.** The PowerCenter Integration Service uses all grouped or sorted ports as a compound partition key. You may need to use hash auto-keys partitioning at Rank,Sorter, and unsorted Aggregator transformations.

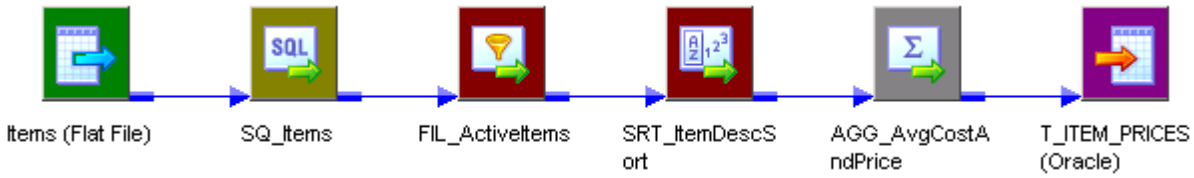
  - **Hash user keys.** The PowerCenter Integration Service uses a hash function to group rows of data among partitions. You define the number of ports to generate the partition key.

- **Key range.** You specify one or more ports to form a compound partition key. The PowerCenter Integration Service passes data to each partition depending on the ranges you specify for each port. Use key range partitioning where the sources or targets in the pipeline are partitioned by key range.
- **Pass-through.** The PowerCenter Integration Service passes all rows at one partition point to the next partition point without redistributing them. Choose pass-through partitioning where you want to create an additional pipeline stage to improve performance, but do not want to change the distribution of data across partitions.
- **Round-robin.** The PowerCenter Integration Service distributes blocks of data to one or more partitions. Use round-robin partitioning so that each partition processes rows based on the number and size of the blocks.

informatica Marketplace

Setting Partition Types in the Pipeline

You can create different partition types at different points in the pipeline. The following figure shows a mapping where you can create partition types to increase session performance:



This mapping reads data about items and calculates average wholesale costs and prices. The mapping must read item information from three flat files of various sizes, and then filter out discontinued items. It sorts the active items by description, calculates the average prices and wholesale costs, and writes the results to a relational database in which the target tables are partitioned by key range.

You can delete the default partition point at the Aggregator transformation because hash auto-keys partitioning at the Sorter transformation sends all rows that contain items with the same description to the same partition.

Therefore, the Aggregator transformation receives data for all items with the same description in one partition and can calculate the average costs and prices for this item correctly.

When you use this mapping in a session, you can increase session performance by defining different partition types at the following partition points in the pipeline:

Source qualifier. To read data from the three flat files concurrently, you must specify three partitions at the source qualifier. Accept the default partition type, pass-through.

Filter transformation. Since the source files vary in size, each partition processes a different amount of data. Set a partition point at the Filter transformation, and choose round-robin partitioning to balance the load going into the Filter transformation.

Sorter transformation. To eliminate overlapping groups in the Sorter and Aggregator transformations, use hash auto-keys partitioning at the Sorter transformation. This causes the Integrated Service to group all items with the same description into the same partition before the Sorter and Aggregator transformations process the rows. You can delete the default partition point at the Aggregator transformation.

Target. Since the target tables are partitioned by key range, specify key range partitioning at the target to optimize writing data to the target.

Setting Partition Types

The Workflow Manager sets a default partition type for each partition point in the pipeline. The Workflow Manager specifies pass-through as the default partition type for all partition points unless the transformation scope for a transformation is All Input. You can change the default type.

For example, at the source qualifier and target instance, the Workflow Manager specifies pass-through partitioning. For Rank and unsorted Aggregator transformations, the Workflow Manager specifies hash auto-keys partitioning when the transformation scope is All Input.

You must specify pass-through partitioning for all transformations that are downstream from a transaction generator or an active source that generates commits and upstream from a target or a transformation with Transaction transformation scope. Also, if you configure the session to use constraint-based loading, you must specify pass-through partitioning for all transformations that are downstream from the last active source.

If workflow recovery is enabled, the Workflow Manager sets the partition type to pass-through unless the partition point is either an Aggregator transformation or a Rank transformation.

You cannot create partition points for the following transformations:

- Source definition
- Sequence Generator
- XML Parser
- XML target
- Unconnected transformations

The following table lists valid partition types and the default partition type for different partition points in the pipeline:

Table 1. Valid Partition Types for Partition Points

Transformation (Partition Point)	Round-Robin	Hash Auto-Keys	Hash User Keys	Key Range	Pass-Through	Database Partitioning
Source Qualifier (relational sources)	no	no	no	yes	yes	yes (Oracle, DB2)
Source Qualifier (flat file sources)	no	no	no	no	yes	no
Web Service Source Qualifier	no	no	no	no	yes	no
XML Source Qualifier	no	no	no	no	yes	no
Normalizer (COBOL sources)	no	no	no	no	yes	no
Normalizer (relational)	yes	no	yes	yes	yes	no
Aggregator (sorted)	no	no	no	no	yes	no
Aggregator (unsorted)	no	yes	no	no	yes	no
Custom	yes	no	yes	yes	yes	no
Data Masking	yes	no	yes	yes	yes	no
Expression	yes	no	yes	yes	yes	no
External Procedure	yes	no	yes	yes	yes	no

Transformation (Partition Point)	Round-Robin	Hash Auto-Keys	Hash User Keys	Key Range	Pass-Through	Database Partitioning
Filter	yes	no	yes	yes	yes	no
HTTP	no	no	no	no	yes	no
Java	yes	no	yes	yes	yes	no
Joiner	no	yes	no	no	yes	no
Lookup	yes	yes	yes	yes	yes	no
Rank	no	yes	no	no	yes	no
Router	yes	no	yes	yes	yes	no
Sorter	no	yes	no	yes	yes	no
Stored Procedure	yes	no	yes	yes	yes	no
Transaction Control	yes	no	yes	yes	yes	no
Union	yes	no	yes	yes	yes	no
Unstructured Data	yes	no	yes	yes	yes	no
Update Strategy	yes	no	yes	yes	yes	no
Web Service Consumer	no	no	no	no	yes	no
XML Generator	no	no	no	no	yes	no
XML Parser	no	no	no	no	yes	no
Relational target definition	yes	no	yes	yes	yes	yes (DB2)
Flat file target definition	yes	no	yes	yes	yes	no
Web Service target	no	no	no	no	yes	no

For the following transformations, the default partition type is pass-through when the transformation scope is Transaction, and the default partition type is hash auto-keys when the transformation scope is All Input:

- Aggregator (unsorted)
- Joiner
- Rank
- Sorter

informatica Marketplace

Database Partitioning Partition Type

You can optimize session performance by using the database partitioning partition type for source and target databases. When you use source database partitioning, the Integration Service queries the database system for table partition information and fetches data into the session partitions. When you use target database partitioning, the Integration Service loads data into corresponding database partition nodes.

Use database partitioning for Oracle and IBM DB2 sources and IBM DB2 targets. Use any number of pipeline partitions and any number of database partitions. However, you can improve performance when the number of pipeline partitions equals the number of database partitions.

Database partitioning can improve performance for IBM DB2 sources and targets that use range partitioning. For Oracle sources that use composite partitioning, you can improve performance when the number of pipeline partitions equals the number of database sub partitions. For example, if an Oracle source contains three partitions and two sub partitions for each partition, set the number of pipeline partitions at the source to six.

Partitioning Database Sources

When you use source database partitioning, the Integration Service queries the database system catalog for partition information. It distributes the data from the database partitions among the session partitions. If he session has more partitions than the database, the Integration Service generates SQL for each database partition and redistributes the data to the session partitions at the next partition point.

Database Partitioning with One Source

When you use database partitioning with a source qualifier with one source, the Integration Service generates SQL queries for each database partition and distributes the data from the database partitions among the session partitions equally.

For example, when a session has three partitions, and the database has five partitions, the Integration Service executes SQL queries in the session partitions against the database partitions. The first and second session partitions receive data from two database partitions. The third session partition receives data from one database partition.

When you use an Oracle database, the Integration Service generates SQL statements similar to the following statements for partition 1:

SELECT <column list> FROM <table name> PARTITION <database_partition1 name> UNION ALL

SELECT <column list> FROM <table name> PARTITION <database_partition4 name> UNION ALL

When you use an IBM DB2 database, the Integration Service creates SQL statements similar to the following for partition 1:

SELECT <column list> FROM <table name>

WHERE (nodenumber(<column 1>)=0 OR nodenumber(<column 1>) = 3)

If an Oracle source has five partitions, 1–5, and two subpartitions, a and b, in each partition, and a session has three partitions, the Integration Service executes SQL queries in the session partitions against the database subpartitions. The first and second session partitions receive data from four database subpartitions. The third session partition receives data from two database subpartitions.

The Integration Service generates SQL statements similar to the following statements for partition 1:

SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition1_a name> UNION ALL

SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition1_b name> UNION ALL

SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition4_a name> UNION ALL

SELECT <column list> FROM <table name> SUBPARTITION <database_subpartition4_b name> UNION ALL