How to Use Pushdown Optimization in Teradata Sessions
Abstract

This article explains how to use pushdown optimization to move data between PowerCenter and Teradata databases. It also lists issues that you might encounter when loading data to or unloading data from Teradata and the workarounds.

Supported Versions

- PowerCenter 9.x

Table of Contents

Overview .......................................................................................................................................... 2
Pushdown Optimization ................................................................................................................... 3
  Pushdown Optimization Scenarios .............................................................................................. 3
  Pushdown Optimization Types .................................................................................................... 4
Pushdown Compatibility ................................................................................................................ 5
Transformations that can be Pushed Down to Teradata ............................................................. 6
Rules and Guidelines ................................................................................................................... 6
Pushdown Optimization Configuration ............................................................................................ 7
  Full Pushdown Optimization without Affecting the Source System ............................................. 7
  Full Pushdown Optimization with Parallel Lookups ..................................................................... 8
  Pushdown Optimization with Sorted Aggregation ....................................................................... 9
  Pushdown Optimization for an Aggregator Transformation ......................................................... 10
  Pushdown Optimization when a Transformation Contains a Variable Port ............................... 10
  Pushdown Optimization in Mappings with Multiple Targets ...................................................... 10
  Pushdown Optimization for a Session that Contains an SQL Override ....................................... 11
Troubleshooting ............................................................................................................................. 12

Overview

PowerCenter works with Teradata database and tools to provide a data integration solution that allows you to integrate data from virtually any business system into Teradata. You can also leverage Teradata data for use in other business systems. When you run sessions that move data between PowerCenter and Teradata databases, you can improve session performance by using pushdown optimization.

Use pushdown optimization to reduce the amount of data passed between Teradata and PowerCenter or when the Teradata database can process transformation logic faster than PowerCenter.
For more information about pushdown optimization, see the *PowerCenter Advanced Workflow Guide*.

**Pushdown Optimization**

Pushdown optimization improves session performance by “pushing” as much transformation logic as possible to the Teradata source or target database. When you run a session configured for pushdown optimization, the PowerCenter Integration Service translates the transformation logic into SQL queries and sends the queries to the Teradata database. The Teradata database executes the SQL queries to process the transformation logic.

The PowerCenter Integration Service processes any transformation logic that cannot be pushed to the database.

The following image shows how pushdown optimization works with a Teradata database system:

---

*Pushdown Optimization Scenarios*

Use pushdown optimization to improve the performance of sessions that use Teradata relational connections to connect to Teradata. Pushdown optimization can improve session performance in the following scenarios:

- You need to reduce the number of rows passed between Teradata and PowerCenter. For example, pushing a Filter transformation to the Teradata source can reduce the number of rows PowerCenter extracts from the source.
• The database server is more powerful than the machine on which the PowerCenter Integration Service runs. For example, pushing a complex Expression transformation to the source or target improves performance when the database server can perform the expression faster than the machine on which the PowerCenter Integration Service runs.

• The generated query can take advantage of prebuilt indexes. For example, pushing a Joiner transformation to the Teradata source improves performance when the database can join tables using indexes and statistics that PowerCenter cannot access.

Example
The following image shows a mapping in which you can improve performance by using pushdown optimization:

If you configure this mapping for pushdown optimization, the PowerCenter Integration Service generates an SQL query based on the Filter (FIL_ACCT) and Lookup (LKP_ADDR) transformation logic and pushes the query to the source database. This improves session performance because it reduces the number of rows sent to PowerCenter.

The PowerCenter Integration Service processes the Java transformation (JAVA_UPDATEBAL) logic since that cannot be pushed to the database, and then loads data to the target.

Pushdown Optimization Types
To configure a session to use pushdown optimization, choose a Pushdown Optimization type in the session properties. You can select one of the following pushdown optimization types:

<table>
<thead>
<tr>
<th>Pushdown Optimization Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>The PowerCenter Integration Service does not push any transformation logic to the database.</td>
</tr>
<tr>
<td>To Source</td>
<td>The PowerCenter Integration Service analyzes the mapping from the source to the target or until it reaches a downstream transformation that it cannot push to the database. It pushes as much transformation logic as possible to the source database. The PowerCenter Integration Service generates SQL in the following form: SELECT ... FROM source ... WHERE (filter/join condition) ... GROUP BY</td>
</tr>
<tr>
<td>To Target</td>
<td>The PowerCenter Integration Service analyzes the mapping from the target back to the source or until it reaches an upstream transformation that it cannot push to the database. It pushes as much transformation logic as possible to the target database. The PowerCenter Integration Service generates SQL in the following form: INSERT INTO target(...) VALUES (?, UPPER(?))</td>
</tr>
</tbody>
</table>
**Pushdown Optimization Type**

<table>
<thead>
<tr>
<th>Pushdown Optimization Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| Full                       | The PowerCenter Integration Service attempts to push all transformation logic to the target database. If the PowerCenter Integration Service cannot push all transformation logic to the database, it performs both source-side and target-side pushdown optimization. 

The PowerCenter Integration Service generates SQL in the following form: 

```
INSERT INTO target(...) SELECT ... FROM source ...
``` |
| $\$\$Pushdown Config        | Allows you to run the same session with different pushdown optimization configurations at different times. |

**Pushdown Compatibility**

To push a transformation with multiple connections to a database, the connections must be pushdown compatible. Connections are pushdown compatible if they connect to databases on the same database management system and the PowerCenter Integration Service can identify the database tables that the connections access.

The following transformations can have multiple connections:

- The Joiner transformation can join data from multiple source connections.
- The Union transformation can merge data from multiple source connections.
- The connection for the Lookup transformation can differ from the source connection.
- The target connection can differ from the source connection.

Each connection object is pushdown compatible with itself. If you configure a session to use the same connection object for the source and target connections, the PowerCenter Integration Service can push the transformation logic to the source or target database.

Some relational connections are pushdown compatible if they are of the same database type, have the same database user name and password, and have certain identical properties.

To enable pushdown compatibility, the following connection properties must be identical for Teradata:

- Code page
- Data source name
- Connection environment SQL
- Transaction environment SQL
Transformations that can be Pushed Down to Teradata

The PowerCenter Integration Service can push the logic for the following transformations to Teradata:

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Pushdown Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregator</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Expression*</td>
<td>Source-side, Target-side, Full</td>
</tr>
<tr>
<td>Filter</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Joiner</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Lookup, connected</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Lookup, unconnected</td>
<td>Source-side, Target-side, Full</td>
</tr>
<tr>
<td>Router</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Sorter</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Source Qualifier</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Target</td>
<td>Target-side, Full</td>
</tr>
<tr>
<td>Union</td>
<td>Source-side, Full</td>
</tr>
<tr>
<td>Update Strategy</td>
<td>Full</td>
</tr>
</tbody>
</table>

* PowerCenter expressions can be pushed down only if there is an equivalent database function. To work around this issue, you can enter an SQL override in the source qualifier.

Rules and Guidelines

Use the following rules and guidelines when you configure the PowerCenter Integration Service to push transformation logic to Teradata:

- If you use ADD_TO_DATE in transformation logic to change days, hours, minutes, or seconds, you cannot push the function to a Teradata database.
- To ensure that a pushdown optimization session does not fail, verify that the transformation logic does not converts a Decimal or Double data type to a String data type.
- To ensure that a target-side pushdown optimization session does not fail, verify that the transformation logic does not converts a Date data type to a String data type.
- When you push LTRIM, RTRIM, or SOUNDEx functions to Teradata, the database treats the argument (' ') as NULL, but the PowerCenter Integration Service treats the argument (' ') as spaces.
- When you push SYSDATE or SYSTIMESTAMP functions to Teradata, the database server returns the timestamp in the time zone of the database server, not the PowerCenter Integration Service.
The PowerCenter Integration Service converts all dates before pushing transformations to a Teradata database. To ensure that the session does not fail, verify that Teradata supports the date format after the date conversion.

You cannot use the HH24 format in the date format string for Teradata. When the PowerCenter Integration Service generates SQL for a Teradata database, it uses the HH format string instead.

You cannot use blank spaces in the date format string in Teradata. When the PowerCenter Integration Service generates SQL for a Teradata database, it substitutes the space with 'B.'

If you push Source Qualifier transformation logic to Teradata with temporary views, the data dictionary in Teradata can cause the SQL statements to fail. The SQL statements fail due to the dynamic creation and deletion of views in the environment that uses many pushdown optimization sessions. You can disable the creation of temporary views for pushdown optimization to Teradata when the Source Qualifier transformation contains a source filter, user-defined joins, or an SQL override. The PowerCenter Integration Service creates derived tables instead of views.

If you use a Teradata source and the Source Qualifier transformation contains a source filter, user-defined joins, or an SQL override, then you do not need to allow the temporary pushdown view. If you push Source Qualifier transformation logic to Teradata with temporary views, the data dictionary in Teradata can cause the SQL statements to fail. The SQL statements fail due to the dynamic creation and deletion of views in the environment that uses many pushdown optimization sessions.

Pushdown Optimization Configuration

You might need to configure pushdown optimization to perform the following tasks:

- Achieve full pushdown optimization without affecting the source.
- Achieve full pushdown optimization with parallel lookups.
- Achieve pushdown optimization with sorted aggregation.
- Achieve pushdown optimization for an Aggregator transformation with pass-through ports.
- Achieve pushdown optimization when a transformation contains a variable port.
- Improve pushdown performance in mappings with multiple targets.
- Achieve pushdown optimization for a session that contains an SQL Override.

Full Pushdown Optimization without Affecting the Source System

To achieve full pushdown optimization for a session in which the source and target reside in different database management systems, you can stage the source data in the Teradata target database.

For example, the following mapping contains an OLTP source and a Teradata target:
Since the source and target tables reside in different database management systems, you cannot configure the session for full pushdown optimization as it is. You could configure the session for source-side pushdown optimization, which would push the Filter and Lookup transformation logic to the source. However, pushing transformation logic to a transactional source might reduce performance of the source database.

To avoid the performance problems caused by pushing transformation logic to the source, you can reconfigure the mapping to stage the source data in the target database.

To achieve full pushdown optimization, redesign the mapping as follows:

1. Create a simple, pass-through mapping to pass all source data to a staging table in the Teradata target database:

   ![Diagram of OLTP_SRC to SQ, SQ to STAGING](image)

   Configure the session to use Teradata PT or a standalone load utility to load the data to the staging table. Do not configure the session to use pushdown optimization.

2. Configure the original mapping to read from the staging table:

   ![Diagram of STAGING to SQ, SQ to FILTER, FILTER to LOOKUP_1, LOOKUP_2, UPDATE_STRATEGY, TD_TGT](image)

   Configure the session to use full pushdown optimization. The PowerCenter Integration Service pushes all transformation logic to the Teradata database, increasing session performance.

### Full Pushdown Optimization with Parallel Lookups

The PowerCenter PowerCenter Integration Service cannot push down mapping logic that contains parallel Lookup transformations. The PowerCenter Integration Service processes all transformations after a pipeline branch when multiple Lookup transformations are present in different branches of pipeline, and the branches merge downstream.

To achieve full pushdown optimization for a mapping that contains parallel lookups, redesign the mapping to serialize the lookups.
For example, the PowerCenter Integration Service cannot fully push down the following mapping:

![Diagram of the mapping](image)

To achieve full pushdown optimization, redesign the mapping so that the lookups are serialized as follows:

![Updated diagram](image)

When you serialize the Lookup transformations, the PowerCenter Integration Service generates an SQL query in which the lookups become part of a subquery. The PowerCenter Integration Service can then push the entire query to the source database.

**Pushdown Optimization with Sorted Aggregation**

The PowerCenter Integration Service cannot push an Aggregator transformation to Teradata if it is downstream from a Sorter transformation.

To achieve pushdown optimization for a mapping that contains a Sorter transformation before an Aggregator transformation, redesign the mapping to remove the Sorter transformation.

For example, the PowerCenter Integration Service cannot push down the Aggregator transformation in the following mapping:

![Original mapping with Sorter and Aggregator](image)

To redesign this mapping to achieve full or source-side pushdown optimization, configure the Aggregator transformation so that it does not use sorted input, and remove the Sorter transformation.
Pushdown Optimization for an Aggregator Transformation

The PowerCenter Integration Service cannot push an Aggregator transformation to Teradata if the Aggregator transformation contains pass-through ports.

To achieve source-side or full pushdown optimization for a mapping that contains an Aggregator transformation with pass-through ports, redesign the mapping to remove the pass-through ports from the Aggregator transformation.

Pushdown Optimization when a Transformation Contains a Variable Port

The PowerCenter Integration Service cannot push down transformation logic when the transformation contains a variable port.

To achieve pushdown optimization for a mapping that contains a transformation with a variable port, update the transformation expression to eliminate the variable port. For example, a transformation contains a variable and an output port with the following expressions:

- **Variable port expression**: \( \text{NET\_AMOUNT} = \text{AMOUNT} - \text{FEE} \)
- **Output port expression**: \( \text{DOLLAR\_AMT} = \text{NET\_AMOUNT} \times \text{RATE} \)

To achieve pushdown optimization for the mapping, remove the variable port and reconfigure the output port as follows:

- **Output port expression**: \( \text{DOLLAR\_AMT} = (\text{AMOUNT} - \text{FEE}) \times \text{RATE} \)

Pushdown Optimization in Mappings with Multiple Targets

If you configure a mapping that contains complex transformation logic and multiple targets for full pushdown optimization, the PowerCenter Integration Service generates one “INSERT … SELECT …” SQL query for each target. This makes pushdown optimization inefficient because it can cause duplicate processing of complex transformation logic within the database.

To improve session performance when using full pushdown optimization for mappings with multiple targets, redesign the original mapping to stage the target data in the Teradata database. Then create a second mapping that uses the staging table as the source.
For example, the following mapping contains two Teradata sources and two Teradata targets, all in the same RDBMS:

To achieve full pushdown optimization, redesign the mapping as follows:

1. Configure the original mapping to write to a staging table in the Teradata target database:

2. Create a second mapping to pass all target data from the staging table to the Teradata targets:

*Pushdown Optimization for a Session that Contains an SQL Override*

In a mapping, the Source Qualifier transformation provides the SQL Query option to override the default query. You can enter an SQL statement supported by the source database. When you override the default SQL query for a session configured for pushdown optimization, the PowerCenter Integration Service creates a view to represent the SQL override. It then runs an SQL query against this view to push the transformation logic to the database.
To use an SQL override in a session configured for pushdown optimization, enable the Allow Temporary View for Pushdown option in the session properties. This option allows the PowerCenter Integration Service to create temporary view objects in the database when it pushes the session to the database. The PowerCenter Integration Service uses a prefix of PM_V for the view objects it creates.

When the session completes, the PowerCenter Integration Service drops the view from the database. If the session does not complete successfully, the PowerCenter Integration Service might not drop the view. You can remove the views manually.

To search for views generated by the PowerCenter Integration Service, run the following query against the Teradata source database:

```sql
SELECT TableName FROM DBC.Tables WHERE CreatorName = USER
 AND TableKind = 'V'
 AND TableName LIKE 'PM\_V%' ESCAPE '\'
```

**Note:** If required, you can disable creation of temporary views for pushdown optimization to Teradata when the Source Qualifier transformation contains source filter, user defined joins, or SQL override.

**Rules and Guidelines**

To avoid problems when you run a pushdown session that contains an SQL override, use the following rules and guidelines:

- Ensure that the SQL override syntax is compatible with the Teradata source database. PowerCenter does not validate the syntax, so test the query before you push it to the database.
- Do not use an order by clause in the SQL override.
- Use ANSI outer join syntax in the SQL override. If the Source Qualifier transformation contains Informatica outer join syntax in the SQL override, the PowerCenter Integration Service processes the Source Qualifier transformation logic.
- If the Source Qualifier transformation is configured for a distinct sort and contains an SQL override, the PowerCenter Integration Service ignores the distinct sort configuration.
- If the Source Qualifier contains multiple partitions, specify the SQL override for all partitions.
- Do not use a Sequence Generator transformation in the mapping. Teradata does not have a sequence generator function or operator.

**Troubleshooting**

This section describes issues that you might encounter when you use pushdown optimization to improve session performance:

- When you run a session configured for full pushdown optimization and the session processes a large amount of data, the database must run a long transaction. The following database performance issues can result from long transactions:
  - A long transaction uses more database resources.
  - A long transaction locks the database for longer periods of time, which reduces database concurrency and increases the likelihood of deadlock.
- A long transaction can increase the likelihood that an unexpected event may occur.

- When the database server performs the transactions, the session time information in the session logs is based on the database server time. Time information such as session start time, time zone, and timestamp generated when a transaction runs on the database server may be different from the time information generated when the transaction runs on the server that hosts PowerCenter.

- Transactions performed by the database can produce different results than transactions performed by the PowerCenter Integration Service. Results can differ in the following areas:
  - Float arithmetic
  - Null values
  - Sort order
  - Case sensitivity
  - Numeric to character conversion
  - Precision

For example, the default numeric precision for Teradata is different from the default numeric precision for PowerCenter. A PowerCenter transformation Decimal datatype has a precision of 1-28. The corresponding Teradata Decimal datatype has a precision of 1-18. The results can vary if the database uses a different precision than the PowerCenter Integration Service.

- The PowerCenter Integration Service cannot track errors that occur in the database. When the PowerCenter Integration Service runs a session configured for full pushdown optimization and an error occurs, the database handles the errors. When the database handles errors, the PowerCenter Integration Service does not write reject rows to the reject file.

- The PowerCenter Integration Service cannot trace all events that occur in the database server. The PowerCenter Integration Service does not write the details of transformations processed by the database to the session log. The statistics that the PowerCenter Integration Service can trace depend on the type of pushdown optimization and determine the contents of the session log.

When the session is configured for full pushdown optimization, the session log does not contain the thread busy percentage and does not contain the number of rows read from the source.

When the session is configured for source-side pushdown optimization, the session log contains the number of rows read after sources are optimized.

Author

Jyothi Jandhyala