

## Configuring pushdown optimization for Snowflake using the ODBC Connector

## Abstract

This article explains how to configure pushdown optimization for a mapping that uses the ODBC connection type to read from or write to Snowflake. This article also explains the pushdown optimization functions, transformations, operators, and data types that you can push down to Snowflake using the Snowflake ODBC connection type.

## Supported Versions

- Informatica Cloud® Data Integration

## Table of Contents

Overview. . . . .	2
Prepare for pushdown optimization using a Snowflake ODBC connection. . . . .	3
Configure the Snowflake ODBC driver on Windows. . . . .	3
Configure the Snowflake ODBC driver on Linux. . . . .	6
Create a Snowflake ODBC connection. . . . .	7
Configuring pushdown optimization. . . . .	8
Pushdown compatibility for Snowflake ODBC connection. . . . .	8
Functions and operators with Snowflake ODBC connection. . . . .	9
Transformations with Snowflake ODBC connection. . . . .	10

## Overview

When you enable pushdown optimization for a task, the task pushes down the entire transformation logic as SQL queries to the underlying database for processing and the database runs the query. Since the entire transformation logic is encapsulated in an SQL query and there is no back-and-forth data movement between the database and the Informatica engine, the task is processed faster.

It is important to know what your primary use case is before you configure pushdown optimization. If your use case is to process data in the cloud for a mapping that reads from a Cloud Data Warehouse or Cloud Data Lake and writes to a Cloud Data Warehouse target such as Snowflake, you can use pushdown optimization to push the processing logic to the warehouse or data lake using the applicable native connector as the source and Snowflake as the target. The native connectors offer a wide range of capabilities that you can explore. For more information about configuring pushdown optimization and the supported functionalities using Snowflake Data Cloud Connector, see the help for Snowflake Data Cloud Connector.

However, if you are using your on-premises data warehouse or any ODBC-based targets, ODBC based pushdown using the ODBC Connector is the better choice. You do not need any separate license to use ODBC pushdown.

In this article, we'll configure pushdown optimization for a task that uses ODBC Connector to push transformation logic to process in the Snowflake source and target databases. The ODBC connection must use the Snowflake subtype in the connection. We'll show you how to optimize a Snowflake ODBC task to read from or write to Snowflake. You can configure source or full pushdown for the task.

Perform the following steps to optimize a Snowflake ODBC task:

1. Configure the Snowflake ODBC driver.
2. Create an ODBC connection with the ODBC subtype as Snowflake and configure the Snowflake database properties to which you want to connect.
3. Create a mapping that uses the configured ODBC connection and enable the task for pushdown optimization.

When you run the task enabled with pushdown optimization, the task converts the transformation logic to an SQL query. The task sends the query to the database, and the database uses the database resources to run the query.

You can also create a temporary view, temporary sequence, and push logic across databases or schemas. For information about the advanced session properties that you can use with pushdown optimization using a Snowflake ODBC connection, see the topic "Advanced Session Properties" in Tasks in the Data Integration help.

## Prepare for pushdown optimization using a Snowflake ODBC connection

Before you can configure pushdown optimization using a Snowflake ODBC connection, you need to configure the Snowflake ODBC driver based on your operating system.

Snowflake supports Snowflake ODBC drivers on Windows and Linux systems. Install the Snowflake ODBC 64-bit driver based on your system requirement.

### *Configure the Snowflake ODBC driver on Windows*

Before you establish an ODBC connection to connect to Snowflake on Windows, configure the ODBC driver.

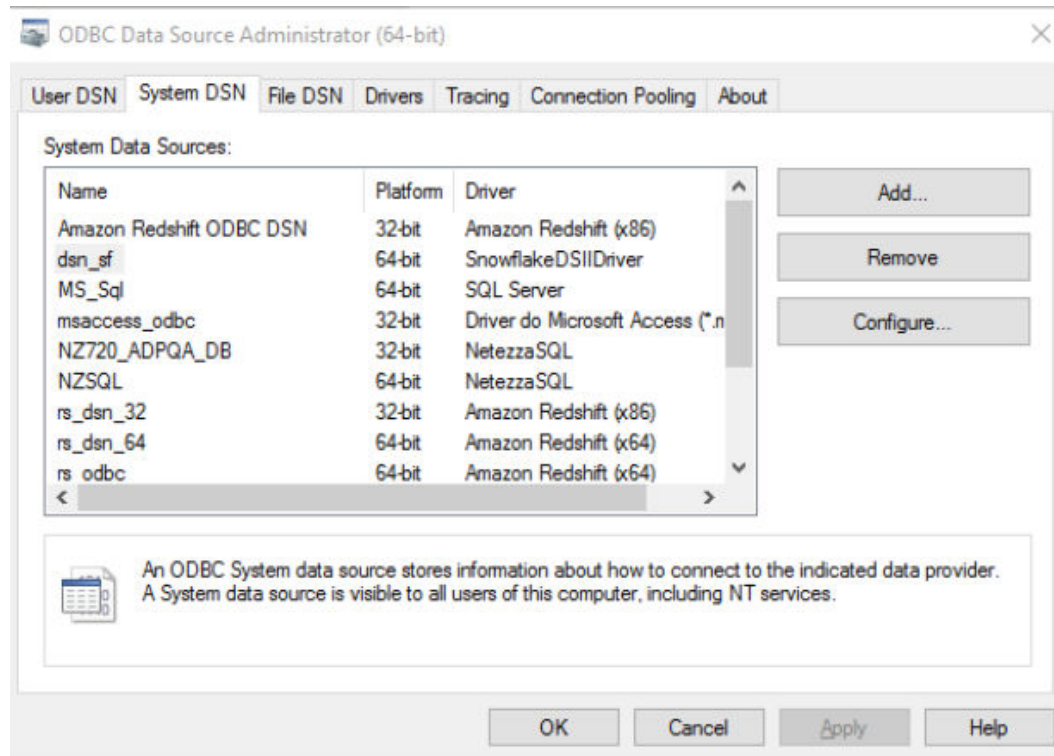
1. Download the Snowflake ODBC 64-bit driver from your Snowflake account.
2. Install the Snowflake ODBC driver on the machine where the Secure Agent is installed.
3. Open the folder in which ODBC data source file is installed.
4. Run the `odbcad32.exe` file.

The **ODBC Data Source Administrator** dialog box appears.

5. Click **System DSN**.

The **System DSN** tab appears.

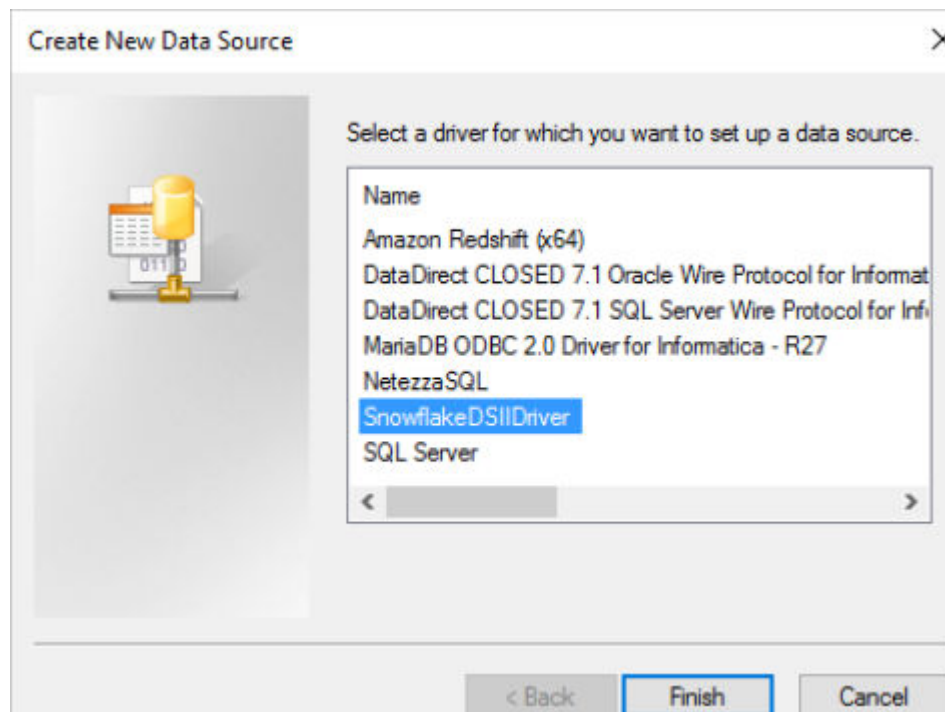
The following image shows the System DSN tab on the ODBC Data Source Administrator (64-bit) dialog box:



6. Click **Add**.

The **Create New Data Source** dialog appears.

The following image shows the Create New Data Source dialog where you can select the Snowflake data source:



7. Select the **SnowflakeDSIIDriver** and click **Finish**.
8. Click **Configure**.

The Snowflake Configuration Dialog appears.

The following image shows the Snowflake Configuration Dialog where you can configure the connection properties:

9. Specify the following connection properties:

Property	Description
Data Source	Name of the data source.
User	User name to access Snowflake.
Password	Password to access Snowflake.
Server	Domain name of your Snowflake account.
Database	Name of the Snowflake database.
Schema	Name of the Snowflake schema.
Warehouse	Name of the Snowflake warehouse.

Property	Description
Role	The Snowflake role assigned to user.
Tracing (0-6)	Determines the amount of detail that appears in the log file. You can specify the following values: <ul style="list-style-type: none"> <li>- 0. Disable tracing.</li> <li>- 1. Fatal error tracing.</li> <li>- 2. Error tracing.</li> <li>- 3. Warning tracing.</li> <li>- 4. Info tracing.</li> <li>- 5. Debug tracing.</li> <li>- 6. Detailed tracing.</li> </ul>

**Note:** It is important to specify the database, schema, and warehouse properties in the ODBC data source name.

10. Click **OK**.

### Configure the Snowflake ODBC driver on Linux

Before you establish an ODBC connection to connect to Snowflake on Linux, configure the ODBC driver. You cannot use the Snowflake ODBC driver with SUSE Linux.

1. Download the Snowflake ODBC 64-bit driver from your Snowflake account.
2. Install the Snowflake ODBC driver on the machine where the Secure Agent is installed.
3. Configure the `odbc.ini` file properties in the following format:

```
[ODBC Data Sources]
driver_name=dsn_name

[dsn_name]
Driver=path/driver_file

Description=
Database=
Schema=
Warehouse=
Server=domain_name
role=role
```

4. Specify the following properties in the `odbc.ini` file:

Property	Description
ODBC Data Sources	Name of the data source.
Driver	Location of the Snowflake ODBC driver file.
Description	Description of the data source.
Database	Name of the Snowflake database.
Schema	Name of the Snowflake schema.
Warehouse	Name of the Snowflake warehouse.

Property	Description
Server	Domain name of your Snowflake account.
Role	The Snowflake role assigned to user.

**Note:** It is important to specify the database, schema, and warehouse properties in the ODBC data source name.

- Run the following command to export the `odbc.ini` file:  

```
Export ODBCINI=/<odbc.ini file path>/odbc.ini
```
- Restart the Secure Agent.

## Create a Snowflake ODBC connection

Create an ODBC connection to connect to Snowflake after you configure the Snowflake ODBC drivers.

Perform the following steps to create a Snowflake ODBC connection on the **Connections** page:

- In Administrator, click **Connections**.  
The Connections page appears.
- Click **New Connection**.

The **New Connection** page appears. The following image shows the New Connection page:

### New Connection

#### Connection Details

Connection Name: \*

Description:

Type: \* ?

#### ODBC Connection Properties

Runtime Environment: \* ?

User Name: \*

Password: \*

Data Source Name: \*

Schema:

Code Page: \*

ODBC Subtype:

Driver Manager for Linux:

- Configure the following connection details in the **Connection Details** section:

Property	Description
Connection Name	Name of the ODBC connection. For example, sf_odbc.
Description	Description of the connection.
Type	Type of the connection. Select the type of the connection as <b>ODBC</b> .

- Configure the following connection details in the **ODBC Connection Properties** section:

Property	Description
Runtime Environment	Runtime environment that contains the Secure Agent you can use to access the system.
User Name	User name to log in to Snowflake.
Password	Password to log in to Snowflake.
Data Source Name	Enter the name of the ODBC data source name that you created for Snowflake.
Schema	Name of the Snowflake schema.
Code Page	The code page of the database server or flat file defined in the connection.
ODBC Subtype	Enter the value of the <b>ODBC Subtype</b> field as <b>Snowflake</b> .
Driver Manager for Linux	The driver that the Snowflake ODBC driver manager sends database calls to.

You can use the configured connection in a mapping. You can then enable pushdown optimization on the **Schedule** tab in the mapping task.

## Configuring pushdown optimization

Create a mapping that uses the configured ODBC connection. To optimize a mapping, add the mapping to a task, and then configure pushdown optimization in the mapping task.

- Create a mapping task.
- In the **Pushdown Optimization** section on the **Schedule** tab, set the pushdown optimization value to **Full** or **To Source**.

When you run the mapping task, the transformation logic is pushed to the configured database.

To verify that the mapping was optimized, you can check the session log for the job. In Monitor, view the log for jobs.

## Pushdown compatibility for Snowflake ODBC connection

You can configure the task to push transformations, variables, functions, and operators to the database.

When you use pushdown optimization, Data Integration converts the expression in the transformation by determining equivalent operators, variables, and functions in the database. If there is no equivalent operator, variable, and function, Data Integration processes the transformation logic.



## Functions and operators with Snowflake ODBC connection

The following table displays the functions that you can push to the Snowflake database by using source-side or full pushdown optimization:

Function	Function	Function
ABS()	LENGTH()	SINH()
ADD_TO_DATE()	LN()	SQRT()
ASCII()	LOG()	STDDEV()
AVG()	LOWER()	SUBSTR()
CEIL()	LPAD()	SUM()
CHR()	LTRIM()	TAN()
CONCAT()	MAX()	TANH()
COS()	MEDIAN()	TO_BIGINT
COSH()	MIN()	TO_CHAR(NUMBER)
COUNT()	MEDIAN()	TO_DATE()
DATE_COMPARE()	MIN()	TO_DECIMAL()
DECODE()	MD5()	TO_FLOAT()
EXP()	MOD()	TO_INTEGER()
FLOOR()	POWER()	TO_NUMBER()
GET_DATE_PART()	REPLACESTR()	TRUNC(DATE)
IIF()	ROUND(NUMBER)	TRUNC(NUMBER)
INITCAP()	RPAD()	UPPER()
INSTR()	RTRIM()	VARIANCE()
ISNULL()	SIGN()	
LAST_DAY()	SIN()	

The following table summarizes the supported operators in a Snowflake database:

+ - \* / % || > = >= <= != AND OR NOT ^=

## Restrictions

When you push functions to Snowflake, adhere to the following guidelines:

- The Snowflake aggregate functions accept only one argument, which is a field set for the aggregate function. The agent ignores any filter condition defined in the argument. In addition, ensure that all fields mapped to the target are listed in the GROUP BY clause.
- When you push the SYSTIMESTAMP() function to the Snowflake database, do not specify any format. The Snowflake database returns the complete time stamp.
- You cannot push the TO\_BIGINT() or TO\_INTEGER() function with more than one argument to the Snowflake database.
- When you push the REPLACESTR() function to the Snowflake database, the agent ignores the caseFlag argument. The REPLACESTR() function must include four parameters for pushdown to work.
- The MD5 function returns NULL if any input is NULL.
- You cannot use millisecond and microsecond values when you push functions to the Snowflake database.
- You must use only the following supported date and time formats:
  - Y
  - YY
  - YYYY
  - YYYY
  - MM
  - MON
  - MONTH
  - D
  - DD
  - DDD
  - DY
  - DAY
  - HH
  - MI
  - SS
  - NS

For information on date and time related functions, see

<https://docs.snowflake.net/manuals/sql-reference/functions-date-time.html#label-supported-date-time-parts>

## Transformations with Snowflake ODBC connection

The following table lists the transformations that you can push using source of full pushdown:

Transformations	Pushdown
Aggregator	Source, Full
Expression	Source, Full

Transformations	Pushdown
Filter	Source, Full
Joiner	Source, Full
Lookup (connected and unconnected)	Source, Full
Sorter	Source, Full
Union	Source, Full
Router	Full
Update Strategy	-
Sequence Generator	Source, Full

## Lookup transformation

You can configure full pushdown optimization to push a Lookup transformation to process in Snowflake. You can push both a connected and an unconnected lookup.

See the following restrictions for multiple matches when you configure a connected and unconnected Lookup transformation:

### Connected lookup

If you use a connected Lookup transformation, you must select the **Return All Rows** multiple matches option in the lookup object properties. If you select any other option other than **Return All Rows**, the pushdown query is not generated.

### Unconnected lookup

For pushdown optimization to work with an unconnected lookup, select the **Report error** multiple matches option. Additionally, you must enable the **Create Temporary View** property in the session properties of the mapping task. If there are multiple matches in the data, the Secure Agent processes the records, but does not log an error when it finds multiple matches.

If there are multiple matches in the data, the Secure Agent processes the records, but does not log an error when it finds multiple matches.

## Source transformation

You must enable the **Create Temporary View** property in the session properties of the mapping task when the Source transformation has the following configurations:

- Filter or joiner in the query options of the source.
- Custom SQL query

Mappings enabled for pushdown optimization has the following restrictions for sources:

- Mappings fail to read data from two Snowflake sources that have fields with the same name and you define a filter condition for one of the common fields.
- To run a SQL query to read from tables where the names contain unicode characters, add a custom flag `OdbcUseUnicodeAPI` and set the value to 1 for the service of type Data Integration Server in the Secure Agent properties.

## Sequence Generator transformation

To use a Sequence Generator transformation, set the **Create Temporary Sequence** advanced session property to **Yes**.

In the **Output** fields of the Sequence Generator transformation, you must not map the CURRVAL field to an input field in a Target transformation or any other downstream transformation.

You can push a Sequence Generator transformation with the following restrictions:

- If the target operation is upsert, update, or delete. Data Integration does not generate the pushdown query for these operations. You must select the target operation as Insert to push the Sequence Generator transformation to Snowflake.
- When the mapping is enabled for cross-database pushdown optimization.

## Target transformation

You can use an existing Snowflake target or create a Snowflake target to write data. The update override target property is not applicable in a target operation.

When you configure a mapping to write data, you might need to set the following configurations based on your use case:

### Not Null constraint

When you define the primary key for the target table, you must explicitly define the Not Null constraint. Mappings that upsert, delete, or update data fail when you do not add the Not Null constraint for the target table. You can add the Not Null constraint and rerun the mapping.

### Create views

You must enable the **Create Temporary View** property in the session properties of the mapping task when the Target transformation is configured to upsert, update, or delete data in Snowflake.

When you configure a mapping to create a new target at runtime, you might need to set the following configurations based on your use case:

### SQL identifiers

To handle quotes, you must explicitly enable the AddQuotesAlwaysPDO flag and set the value to Yes in the custom properties in the advanced session properties of the mapping task .

### Truncate a target

If you specify the truncate target option in the target mapping, add the **AddQuotesAlways** DTM property and set the value to **Yes** in the Secure Agent custom configuration details. However, mappings fail when you attempt to truncate a Snowflake target that contains special characters.

## Restrictions

A mapping that creates a new target at runtime for pushdown optimization has the following restrictions:

- The TIMESTAMP\_LTZ, TIMETAMP\_TZ, Boolean, and Time data types are not applicable.
- If you drop the target created on Snowflake using the drop table tablename command, and rerun the same mapping, Data Integration does not create the target. It does not trigger the create query and results in an error.
- If the name you specify for the target already exists in Snowflake, Data Integration inserts the data to the existing target table. If you delete the created target table and re-run the mapping task, the Secure Agent fails to create the target table.

## **Author**

**Dimple Rai**