



Informatica® Cloud Data Integration

# Snowflake Cloud Data Warehouse V2 Connector

© Copyright Informatica LLC 2019, 2020

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, Informatica Cloud, Informatica Intelligent Cloud Services, PowerCenter, PowerExchange, and the Informatica logo are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2020-05-04

# Table of Contents

<b>Preface .....</b>	<b>5</b>
Informatica Resources. ....	5
Informatica Documentation. ....	5
Informatica Intelligent Cloud Services web site. ....	5
Informatica Intelligent Cloud Services Communities. ....	5
Informatica Intelligent Cloud Services Marketplace. ....	6
Data Integration connector documentation. ....	6
Informatica Knowledge Base. ....	6
Informatica Intelligent Cloud Services Trust Center. ....	6
Informatica Global Customer Support. ....	6
 <b>Chapter 1: Introduction to Snowflake Cloud Data Warehouse V2 Connector....</b>	 <b>7</b>
Snowflake Cloud Data Warehouse V2 Connector overview. ....	7
Snowflake Cloud Data Warehouse V2 Connector task and object types. ....	7
Administration of Snowflake Cloud Data Warehouse V2 Connector. ....	8
 <b>Chapter 2: Snowflake Cloud Data Warehouse V2 connections.....</b>	 <b>9</b>
Snowflake Cloud Data Warehouse V2 connections overview. ....	9
Snowflake Cloud Data Warehouse V2 connection properties. ....	10
 <b>Chapter 3: Mappings and mapping configuration tasks with Snowflake Cloud Data Warehouse V2 Connector.....</b>	 <b>11</b>
Snowflake Cloud Data Warehouse V2 objects in mappings. ....	11
Snowflake Cloud Data Warehouse V2 sources in mappings. ....	11
Key range partitioning. ....	12
Snowflake Cloud Data Warehouse V2 targets in mappings. ....	13
Snowflake Cloud Data Warehouse V2 lookups in mappings. ....	17
Import Snowflake Objects . ....	21
Snowflake Cloud Data Warehouse V2 mapping example. ....	22
Rules and guidelines for Cloud Data Warehouse V2 objects. ....	24
Target statistics of processed rows in Snowflake Cloud Data Warehouse V2 write operations. ....	26
Configuring custom temp directory for local staging. ....	27
Data Encryption in Snowflake Cloud Data Warehouse sources and targets. ....	27
 <b>Chapter 4: Mass ingestion tasks with Snowflake Cloud Data Warehouse V2 Connector.....</b>	 <b>28</b>
Mass ingestion task overview. ....	28
Before you begin. ....	28
Snowflake Cloud Data Warehouse V2 targets in mass ingestion tasks. ....	29
File format and copy options. ....	30

External stage. . . . .	32
Creating a mass ingestion task. . . . .	35
Viewing mass ingestion task details. . . . .	37
Running a mass ingestion task. . . . .	37
Rules and Guidelines for Snowflake Cloud Data Warehouse mass ingestion tasks. . . . .	37
<b>Chapter 5: Snowflake pushdown optimization. . . . .</b>	<b>38</b>
Pushdown optimization. . . . .	38
Pushdown optimization functions. . . . .	39
Pushdown Optimization Transformations. . . . .	41
Configuring a Snowflake ODBC connection. . . . .	41
Configuring a Snowflake ODBC connection on Windows. . . . .	42
Configuring a Snowflake ODBC connection on Linux. . . . .	45
Create an ODBC connection. . . . .	46
Cross-schema pushdown optimization. . . . .	48
Configuring cross-schema optimization for Snowflake Cloud Data Warehouse mapping. . . . .	48
Cross-database pushdown optimization . . . . .	49
Configuring cross-database pushdown optimization. . . . .	49
Rules and guidelines for pushdown optimization. . . . .	50
Troubleshooting. . . . .	51
<b>Chapter 6: Data type reference. . . . .</b>	<b>52</b>
Data type reference overview. . . . .	52
Snowflake Cloud Data Warehouse V2 and transformation data types. . . . .	52
<b>Index. . . . .</b>	<b>54</b>

# Preface

Use *Snowflake Cloud Data Warehouse V2 Connector* to learn how to read from or write to Snowflake Cloud Data Warehouse. Learn to create a connection, develop mappings, and run mapping and mass ingestion tasks in Cloud Data Integration. Learn how to configure pushdown, cross-schema pushdown optimization to a Snowflake database using a Snowflake ODBC connection.

## Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

### Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

### Informatica Intelligent Cloud Services web site

You can access the Informatica Intelligent Cloud Services web site at <http://www.informatica.com/cloud>. This site contains information about Informatica Cloud integration services.

### Informatica Intelligent Cloud Services Communities

Use the Informatica Intelligent Cloud Services Community to discuss and resolve technical issues. You can also find technical tips, documentation updates, and answers to frequently asked questions.

Access the Informatica Intelligent Cloud Services Community at:

<https://network.informatica.com/community/informatica-network/products/cloud-integration>

Developers can learn more and share tips at the Cloud Developer community:

<https://network.informatica.com/community/informatica-network/products/cloud-integration/cloud-developers>

## Informatica Intelligent Cloud Services Marketplace

Visit the Informatica Marketplace to try and buy Data Integration Connectors, templates, and mapplets:

<https://marketplace.informatica.com/>

## Data Integration connector documentation

You can access documentation for Data Integration Connectors at the Documentation Portal. To explore the Documentation Portal, visit <https://docs.informatica.com>.

## Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at [KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com).

## Informatica Intelligent Cloud Services Trust Center

The Informatica Intelligent Cloud Services Trust Center provides information about Informatica security policies and real-time system availability.

You can access the trust center at <https://www.informatica.com/trust-center.html>.

Subscribe to the Informatica Intelligent Cloud Services Trust Center to receive upgrade, maintenance, and incident notifications. The [Informatica Intelligent Cloud Services Status](#) page displays the production status of all the Informatica cloud products. All maintenance updates are posted to this page, and during an outage, it will have the most current information. To ensure you are notified of updates and outages, you can subscribe to receive updates for a single component or all Informatica Intelligent Cloud Services components. Subscribing to all components is the best way to be certain you never miss an update.

To subscribe, go to <https://status.informatica.com/> and click **SUBSCRIBE TO UPDATES**. You can then choose to receive notifications sent as emails, SMS text messages, webhooks, RSS feeds, or any combination of the four.

## Informatica Global Customer Support

You can contact a Customer Support Center by telephone or online.

For online support, click **Submit Support Request** in Informatica Intelligent Cloud Services. You can also use Online Support to log a case. Online Support requires a login. You can request a login at <https://network.informatica.com/welcome>.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at <https://www.informatica.com/services-and-training/support-services/contact-us.html>.

# CHAPTER 1

## Introduction to Snowflake Cloud Data Warehouse V2 Connector

This chapter includes the following topics:

- [Snowflake Cloud Data Warehouse V2 Connector overview, 7](#)
- [Snowflake Cloud Data Warehouse V2 Connector task and object types, 7](#)
- [Administration of Snowflake Cloud Data Warehouse V2 Connector, 8](#)

## Snowflake Cloud Data Warehouse V2 Connector overview

You can use Snowflake Cloud Data Warehouse V2 Connector to securely read data from or write data to Snowflake. You can also use Snowflake Connector to read data from and write data to Snowflake that is enabled for staging data in Azure or Amazon. You can read data from other applications, databases, and flat files and write data to Snowflake.

You can create a Snowflake Cloud Data Warehouse V2 connection and use the connection in mass ingestion tasks, mappings, and mapping tasks. When you run a Snowflake Cloud Data Warehouse V2 mapping or mapping task, the Secure Agent writes data to Snowflake based on the workflow and Snowflake Cloud Data Warehouse V2 connection configuration. When you run a mass ingestion task, the Secure Agent transfers files from any source that the mass ingestion task supports to a Snowflake Cloud Data Warehouse target.

Informatica recommends that you use Snowflake Connector if you want to create a mapping task to read data from and write data to Snowflake. For more information about using Snowflake Connector, see the *Snowflake Connector Guide*.

## Snowflake Cloud Data Warehouse V2 Connector task and object types

When you create a Snowflake Cloud Data Warehouse V2 connection to perform a task, you can select objects supported by Snowflake Cloud Data Warehouse V2 Connector for the task.

The following table provides the list of tasks and object types supported by Snowflake Cloud Data Warehouse V2 Connector:

Task Type	Source	Target	Lookup
Mapping task	Yes	Yes	Yes
Mass ingestion task	No	Yes	No

## Administration of Snowflake Cloud Data Warehouse V2 Connector

Before you use a Snowflake Cloud Data Warehouse V2 connection, verify if the `Snowflake Cloud Data Warehouse` license is enabled under **Connector Licences** and the `SDKPatch` package is assigned under **Packages**.

## CHAPTER 2

# Snowflake Cloud Data Warehouse V2 connections

This chapter includes the following topics:

- [Snowflake Cloud Data Warehouse V2 connections overview, 9](#)
- [Snowflake Cloud Data Warehouse V2 connection properties, 10](#)

## Snowflake Cloud Data Warehouse V2 connections overview

Create a Snowflake Cloud Data Warehouse V2 connection to securely read data from or write data to Snowflake. You can use a Snowflake Cloud Data Warehouse V2 connection to specify sources and targets in mappings and mapping tasks. You can also specify a Snowflake Cloud Data Warehouse V2 connection in a mass ingestion task to load files from any source that supports mass ingestion to Snowflake Cloud Data Warehouse.

You create a Snowflake Cloud Data Warehouse V2 connection on the Connections page. You can then use the connection in the Mapping Designer when you create a mapping or in the MAass Ingestion Designer when you create a mass ingestion task.

# Snowflake Cloud Data Warehouse V2 connection properties

When you set up a Snowflake Cloud Data Warehouse V2 connection, you must configure the connection properties.

The following table describes the Snowflake Cloud Data Warehouse V2 connection properties:

Connection property	Description
Runtime Environment	The name of the runtime environment where you want to run the tasks. Snowflake Cloud Data Warehouse V2 also supports the Hosted Agent.
Username	The user name to connect to the Snowflake Cloud Data Warehouse account.
Password	The password to connect to the Snowflake Cloud Data Warehouse account.
Account	The name of the Snowflake account. In the Snowflake URL, your account name is the first segment in the domain. For example, 123abc is your account name in <code>https://123abc.snowflakecomputing.com</code> .
Warehouse	The Snowflake warehouse name. You must specify the warehouse name.
Role	The Snowflake role assigned to the user.
Additional JDBC URL Parameters	<p>Optional. The additional JDBC connection parameters.</p> <p>Enter one or more JDBC connection parameters in the following format:</p> <pre>&lt;param1&gt;=&lt;value&gt;&amp;&lt;param2&gt;=&lt;value&gt;&amp;&lt;param3&gt;=&lt;value&gt;....</pre> <p>For example:</p> <pre>user=jon&amp;warehouse=mywh&amp;db=mydb&amp;schema=public</pre> <p>To override the database and schema name used to create temporary tables in Snowflake, enter the database and schema name in the following format:</p> <pre>ProcessConnDB=&lt;DB name&gt;&amp;ProcessConnSchema=&lt;schema_name&gt;</pre> <p>To access Snowflake through Okta SSO authentication, enter the web-based IdP implementing SAML 2.0 protocol in the following format:</p> <pre>authenticator=https://&lt;Your_Okta_Account_Name&gt;.okta.com</pre> <p><b>Note:</b> Microsoft ADFS is not supported.</p> <p>For more information about configuring Okta authentication, see the following website: <a href="https://docs.snowflake.net/manuals/user-guide/admin-security-fed-auth-configure-snowflake.html#configuring-snowflake-to-use-federated-authentication">https://docs.snowflake.net/manuals/user-guide/admin-security-fed-auth-configure-snowflake.html#configuring-snowflake-to-use-federated-authentication</a></p>

## CHAPTER 3

# Mappings and mapping configuration tasks with Snowflake Cloud Data Warehouse V2 Connector

This chapter includes the following topics:

- [Snowflake Cloud Data Warehouse V2 objects in mappings, 11](#)
- [Import Snowflake Objects , 21](#)
- [Snowflake Cloud Data Warehouse V2 mapping example, 22](#)
- [Rules and guidelines for Cloud Data Warehouse V2 objects, 24](#)
- [Target statistics of processed rows in Snowflake Cloud Data Warehouse V2 write operations, 26](#)
- [Configuring custom temp directory for local staging, 27](#)
- [Data Encryption in Snowflake Cloud Data Warehouse sources and targets, 27](#)

## Snowflake Cloud Data Warehouse V2 objects in mappings

When you create a mapping, you can configure a Source or Target transformation to represent a Snowflake Cloud Data Warehouse V2 object. You can use tables or views as Snowflake Cloud Data Warehouse V2 objects.

### Snowflake Cloud Data Warehouse V2 sources in mappings

In a mapping, you can configure a Source transformation to represent a Snowflake Cloud Data Warehouse V2 source.

You can configure partitioning to optimize the mapping performance at run time when you read data from Snowflake Cloud Data Warehouse. The partition type controls how the agent distributes data among partitions at partition points. You can define the partition type as key range partitioning. With partitioning, the agent distributes rows of source data based on the number of threads that you define as partition.

The following table describes the Snowflake Cloud Data Warehouse V2 source properties that you can configure in a Source transformation:

Property	Description
Connection	Name of the source connection.
Source Type	Type of the source object. Select Single Object, Multiple Objects, Query, or Parameter. <b>Note:</b> When you use a custom SQL query to import Snowflake Cloud Data Warehouse tables, the Secure Agent fetches the metadata using separate metadata calls.
Object	The source object for the task. Select the source object for a single source. When you select the multiple source option, you can add source objects and configure relationship between them.
Filter	Filters records based on the filter condition. Configure a simple filter.
Sort	Sorts records based on the conditions you specify. You can specify the following sort conditions: <ul style="list-style-type: none"> <li>- Not parameterized. Select the fields and type of sorting to use.</li> <li>- Parameterized. Use a parameter to specify the sort option.</li> </ul>

The following table describes the advanced properties that you can configure in a Source transformation:

Advanced Property	Description
Database	Overrides the database specified in the connection.
Schema	Overrides the schema specified in the connection.
Warehouse	Overrides the Snowflake Cloud Data Warehouse name specified in the connection.
Role	Overrides the Snowflake Cloud Data Warehouse role assigned to user, specified in the connection.
Pre SQL	The pre-SQL command to run on the Snowflake Cloud Data Warehouse source table before the Secure Agent reads the data. For example, if you want to update records in the database before you read the records from the table, specify a pre-SQL statement.
Post SQL	The post-SQL command to run on the Snowflake Cloud Data Warehouse table after the Secure Agent completes the read operation. For example, if you want to delete some records after the latest records are loaded, specify a post-SQL statement.
Table Name	Overrides the table name of the imported Snowflake Cloud Data Warehouse source table.
SQL Override	The SQL statement to override the default query used to read data from the Snowflake Cloud Data Warehouse source.
Tracing Level	Determines the amount of detail that appears in the log file. You can select Terse, Normal, Verbose Initialization, or Verbose Data. Default value is Normal.

## Key range partitioning

You can configure key range partitioning when you use a mapping task to read data from Snowflake Cloud Data Warehouse sources. With key range partitioning, the agent distributes rows of source data based on the

field that you define as partition keys. The agent compares the field value to the range values for each partition and sends rows to the appropriate partitions.

Use key range partitioning for columns that have an even distribution of data values. Otherwise, the partitions might have unequal size. For example, a column might have 10 rows between key values 1 and 1000 and the column might have 999 rows between key values 1001 and 2000. If the mapping includes multiple sources, use the same number of key ranges for each source.

When you define key range partitioning for a column, the agent reads the rows that are within the specified partition range. For example, if you configure two partitions for a column with the ranges as 10 through 20 and 30 through 40, the agent does not read the rows 20 through 30 because these rows are not within the specified partition range.

You can configure a partition key for fields of the following data types:

- Integer
- String
- Any type of number data type. However, you cannot use decimals in key range values.
- Datetime. Use the following format to specify the date and time: YYYY-MM-DD HH24:MI:SS. For example, 1971-01-01 12:30:30

**Note:** If you specify the date and time in any other format, the task fails.

## Configuring key range partitioning

Perform the following steps to configure key range partitioning for Snowflake Cloud Data Warehouse V2 sources:

1. In the **Source Properties** page, click the **Partitions** tab.
2. Select the required partition key from the list.
3. Click **Add New Key Range** to define the number of partitions and the key ranges based on which the agent must partition data.

Use a blank value for the start range to indicate the minimum value. Use a blank value for the end range to indicate the maximum value.

The following image displays the details of **Partitions** tab:

The screenshot shows the 'NewSource Properties' dialog with the 'Partitions' tab selected. The 'Partition key' is set to 'ROW\_ID'. Below it is a table titled 'Key Ranges' with columns 'Partition', 'Start range', and 'End range'.

Partition	Start range	End range
#1		100000
#2	100000	200000
#3	200000	300000
#4	300000	400000
#5	400000	500001

## Snowflake Cloud Data Warehouse V2 targets in mappings

In a mapping, you can configure a Target transformation to represent a Snowflake Cloud Data Warehouse V2 target.

You can write data to an existing table or create a table in the target by using create target option.

You can configure partitioning to optimize the mapping performance at run time when you write data to Snowflake Cloud Data Warehouse V2 targets. The partition type controls how the agent distributes data among partitions at partition points. You can define the partition type as passthrough partitioning. With

partitioning, the agent distributes rows of target data based on the number of threads that you define as partition.

The following table describes the Snowflake Cloud Data Warehouse V2 target properties that you can configure in a Target transformation:

Property	Description
Connection	Name of the target connection.
Target Type	Type of target object.
Object	The target object for the task. Select the target object. You can either select an existing table or create a new table.
Create New at Runtime	Creates a Snowflake Cloud Data Warehouse target table at runtime based on the table type and the path you specify. To create a target table at runtime, provide the following parameters: <ul style="list-style-type: none"> <li>- Optional. Specify the table type as <code>table</code>.</li> <li>- In the <b>Path</b> field, specify the Snowflake Cloud Data Warehouse database name and schema in the following format: <code>&lt;database_name&gt;/&lt;schema&gt;</code></li> </ul> The Secure Agent creates the target table based on the object name and the path you specify. <b>Note:</b> You can edit the metadata of the source fields before creating the target.
Operation	The target operation. Select Insert, Update, Upsert, Delete, or Data Driven.
Update columns	The temporary key column to update data to or delete data from a Snowflake Cloud Data Warehouse target. If you perform an update, update else insert, or delete operation and the Snowflake Cloud Data Warehouse V2 target does not include a primary key column, click Add to add a temporary key. You can select multiple columns.

The following table describes the advanced properties that you can configure in a Target transformation:

Advanced Property	Description
Database	Overrides the database that you used to import the object.
Schema	Overrides the schema that you used to import the object.
Warehouse	Overrides the Snowflake Cloud Data Warehouse name specified in the connection.
Role	Overrides the Snowflake Cloud Data Warehouse role assigned to user specified in the connection.
Pre SQL	The pre-SQL command to run before the Secure Agent writes to Snowflake Cloud Data Warehouse. For example, if you want to assign sequence object to a primary key field of the target table before you write data to the table, specify a pre-SQL statement.
Post SQL	The pre-SQL command to run after the Secure Agent completes the write operation. For example, if you want to alter the table created by using create target option and assign constraints to the table before you write data to the table, specify a post-SQL statement.
Batch Row Size	The number of rows written to a single file in the Secure Agent location. When the number of rows written to the file reaches the value specified, the Secure Agent flushes the data queue and starts processing the write commands.

Advanced Property	Description
Number of local staging files	The number of files that represents a single batch of data. The default number of files is 64. After the Secure Agent uploads the specified number of local staging files to the Snowflake user stage, Snowflake unloads the data to the target table.
Truncate Target Table	Truncates the database target table before inserting new rows. Select one of the following options: <ul style="list-style-type: none"> <li>- True. Truncates the target table before inserting all rows.</li> <li>- False. Inserts new rows without truncating the target table</li> </ul> Default is false.
Additional Write Runtime Parameters	Specify additional runtime parameters. For example, if you want to specify user-defined staging objects in the Snowflake database to create temporary tables, specify the staging location in the following format:  <code>remoteStage=CQA.CQA_SCHEMA.CQA_STAGE</code>  If you want to optimize the write performance, you can choose to compress files before writing to Snowflake tables. You can set the compression parameter to On or Off, for example:  <code>Compression=On</code>  By default, compression is on. Separate multiple runtime parameters with &.
Table Name	Overrides the table name of the Snowflake Cloud Data Warehouse target table.
Rejected File Path	The filename and path of the file on the Secure Agent machine where the Secure Agent writes rejected records. For example, <code>\rejectedfiles\reject7</code>
Success File Directory	Not supported.
Error File Directory	Not supported.
Forward Rejected Rows	Determines whether the transformation passes rejected rows to the next transformation or drops rejected rows. By default, the agent forwards rejected rows to the next transformation.

## Configuring the .csv file size for a Snowflake Cloud Data Warehouse target

When you create a mapping to write to Snowflake, you can specify the size of the local staging .csv file in bytes. Specify the local staging file size property, `csvFileSize`, in the **Additional Write Runtime Parameters** field in the advanced Snowflake target properties. The default file size is 50 MB.

If the intended data size is 50 MB, calculate the `csvFileSize` value in bytes, for example  $50 \times 1024 \times 1024$  and then specify 52428800 as the `csvFileSize`. It is recommended that you configure the right combination of the number of local staging files and `csvFileSize` while writing data to Snowflake for better performance.

## Configure loader properties as additional runtime parameters

You can configure write properties to load data to Snowflake Cloud Data Warehouse in the **Additional Write Runtime Parameters** field in the Snowflake Cloud Data Warehouse advanced target properties of the Target transformation.

The following table lists some of the additional runtime parameters that you can specify to load data to Snowflake Cloud Data Warehouse:

Property	Supported Type	Description	Default Value
oneBatch	Boolean	Process all data in a single batch.	false
remoteStage	String	Specifies whether to use internal or external stage.	"~"(user stage)
onError	String	Specifies the action to perform when an error is encountered while loading data from a file. For example, <code>on_error option ABORT_STATEMENT CONTINUE SKIP_FILE</code>	CONTINUE
compressFileByPut	Boolean	Compress file by PUT.	false
compressDataBeforePut	Boolean	Compress data before PUT. The loader compresses the data to a gzip format before uploading the data.	true
copyEmptyFieldAsEmpty	Boolean	The COPY command option to set incoming empty fields as null.	-

When you set the values in the additional runtime parameters field, every configured partition initializes a new loader instance and the configured values apply similarly across all the partitions.

### Example 1

You want to compress files by using the Put command before loading data to Snowflake Cloud Data Warehouse.

Specify the following compression option: `compressDataBeforePut=false&compressFileByPut=true`

If you specify both the options as true, Snowflake Cloud Data Warehouse considers the `compressDataBeforePut` option.

### Example 2

You want to replace the incoming fields with empty values as NULL while loading the data to Snowflake Cloud Data Warehouse.

Specify the `copyEmptyFieldAsEmpty` Boolean option and set the value to true or false based on your requirement.

Consider the following scenarios before you configure the `copyEmptyFieldAsEmpty` Boolean parameter:

- If you do not configure this parameter, Null values are received as NULL, and empty values are received as Empty. This is the default behavior.
- If you set the parameter `copyEmptyFieldAsEmpty=false`, Null values as received as Null and empty values are received as Null.
- If you set the parameter `copyEmptyFieldAsEmpty=true`, Null values are received as empty, while empty values are received as empty.

## Specifying a target

You can use an existing target or create a target to hold the results of a mapping. If you choose to create the target, the agent creates the target when you run the task.

To specify the target properties, perform the following tasks:

1. Select the Target transformation in the mapping.
2. On the **Incoming Fields** tab, configure field rules to specify the fields to include in the target.
3. To specify the target, click the **Target** tab.
4. Select the target connection.
5. For the target type, choose **Single Object** or **Parameter**.
6. Specify the target object or parameter.
7. To specify a target object, perform the following tasks:
  - a. Click **Select** and choose a target object. You can select an existing target object or create a new target object at runtime and specify the object name.

**Target Object**

Select an existing target object or create a new one. Any new target objects will be created when the mapping task is executed.

Target Object: ☐ Existing ☒ Create New at Runtime

Object Name:

TableType:

Path:

- b. To create a target object at runtime, select **Create New at Runtime**.
  - c. Enter the name for the target object in the **Object Name** field.
  - d. Specify the TableType as `table`.

The TableType property is optional.
  - e. In the **Path** field, specify the Snowflake Cloud Data Warehouse database name and schema in the following format: `<database_name>/<schema>`
  - f. Click **OK**.
8. Specify the advanced properties for the target, if required.

## Snowflake Cloud Data Warehouse V2 lookups in mappings

You can create lookups for objects in a Snowflake Cloud Data Warehouse V2 connection. You can retrieve data from a Snowflake Cloud Data Warehouse V2 lookup object based on the specified lookup condition.

When you configure a lookup in Snowflake Cloud Data Warehouse V2, you select the lookup connection and lookup object. You also define the behavior when a lookup condition returns more than one match.

The following table describes the Snowflake Cloud Data Warehouse V2 lookup object properties that you can configure in a Lookup transformation:

Property	Description
Connection	Name of the lookup connection.
Source Type	Type of the source object. Select Single Object or Parameter.
Lookup Object	Name of the lookup object for the mapping.
Multiple Matches	Behavior when the lookup condition returns multiple matches. Select Return any row, Return all rows, or Report error.
Filter	Not supported.
Sort	Not supported.

The following table describes the Snowflake Cloud Data Warehouse V2 lookup object advanced properties that you can configure in a Lookup transformation:

Advanced Property	Description
Database	Overrides the database specified in the connection.
Schema	Overrides the schema specified in the connection.
Warehouse	Overrides the Snowflake warehouse name specified in the connection.
Role	Overrides the Snowflake role assigned to user specified in the connection.
Pre SQL	Not supported.
Post SQL	Not supported.

## Unconnected Lookup transformation

You can configure an unconnected Lookup transformation for the Snowflake Cloud Data Warehouse source in a mapping. Use the Lookup transformation to retrieve data from Snowflake Cloud Data Warehouse based on a specified lookup condition.

An unconnected Lookup transformation is a Lookup transformation that is not connected to any source, target, or transformation in the pipeline.

An unconnected Lookup transformation receives input values from the result of a :LKP expression in another transformation. The Integration Service queries the lookup source based on the lookup ports and condition in the Lookup transformation and passes the returned value to the port that contains the :LKP expression. The :LKP expression can pass lookup results to an expression in another transformation.

For more information about the Lookup transformation, see *Transformations*.

## Configuring an unconnected Lookup transformation

To configure an unconnected Lookup transformation, select the **Unconnected Lookup** option, add incoming fields, configure the lookup condition, and designate a return value. Then configure a lookup expression in a different transformation.

1. Add a Lookup transformation in a mapping.
2. On the **General** tab of the Lookup transformation, enable the **Unconnected Lookup** option.

The diagram illustrates a mapping flow where data from a **Source** transformation is processed by an **Expression** transformation (containing  $f(x)$ ) before being sent to a **Target** transformation. A separate **lkp** (Lookup) transformation is also shown. Below the diagram, the **Properties** panel for the **lkp** transformation is displayed, showing the **General** tab with the **Unconnected Lookup** option checked.

**Properties** | **lkp**

**General**

**Incoming Fields**

**Lookup Object**

**Lookup Condition**

**Return Fields**

**Advanced**

Name: \*

Description:

☒ Unconnected Lookup

3. On the **Incoming Fields** tab of the Lookup transformation, create an incoming field for each argument in the :LKP expression.

For each lookup condition you create, add an incoming field to the Lookup transformation.

Properties

lkp

General

Incoming Fields

Lookup Object

Lookup Condition

Return Fields

Advanced

Incoming Field: 

Not Parameterized

Incoming Fields

Field Name
LOCATIONADDRESS

- In the **Lookup Object** tab, import the lookup object.

Properties

Preview

lkp

General

Incoming Fields

Lookup Object

Lookup Condition

Return Fields

Advanced

Lookup Object Details

Connection: 

conn\_v2 (Snowflake Cloud Data Wareho...

View...

New Connection...

New Parameter...

Source Type: 

Single Object

Lookup Object: 

DB\_PC\_AUTO/SCHEMA\_PC\_AUTO/EMPTA

Select...

Preview Data...

Multiple Matches: 

Return any row

Advanced

The **Multiple Matches** property value **Return all rows** in an unconnected lookup is not supported.

- Designate a return value.

You can pass multiple input values into a Lookup transformation and return one column of data. Data Integration can return one value from the lookup query. Use the return field to specify the return value.

Properties

lkp

General

Incoming Fields

Lookup Object

Lookup Condition

Return Fields

Advanced

Lookup Condition: 

Simple

Lookup Conditions

Lookup Field	Operator	Incoming Field
LOCADDR	=	LOCATIONADDRESS

- Configure a lookup expression in another transformation.

20

Chapter 3: Mappings and mapping configuration tasks with Snowflake Cloud Data Warehouse V2 Connector

Supply input values for an unconnected Lookup transformation from a :LKP expression in a transformation that uses an Expression transformation. The arguments are local input fields that match the Lookup transformation input fields used in the lookup condition.

Field Expression: NewField( decimal, 10, 0 ) ✕

Configure expression by adding fields and functions.

Expression: Not Parameterized ▾

Fields ▾ Expression Validate

LOCATIONID LOCATIONNAME LOCATIONPHONE LOCATIONADDRESS	:LKP.lkp(LOCATIONADDRESS)
--	---------------------------

? OK Cancel

7. Map the fields with the target.

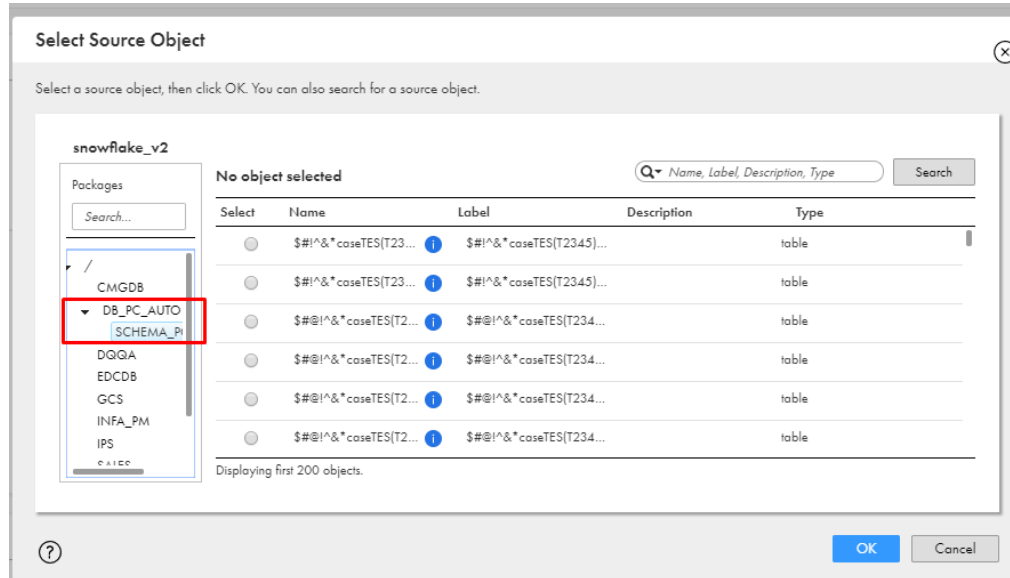
## Import Snowflake Objects

When you create a Source or Target transformation, you can select the Snowflake database and schema from which you can import the tables.

1. To select the Snowflake object in the Source or Target transformation, click **Select** in the **Object** field.

2. In the **Select Source Object** dialog box, select the database and schema and then select the tables from which you want to read from or write to.

The following image shows the employee details tables from a Snowflake database and schema:



You can select the required table in the Source or Target transformation.

## Snowflake Cloud Data Warehouse V2 mapping example

An enterprise application uses the Oracle database to store the product transaction details. You use Snowflake data warehouse to analyze the completed transactions, pending transactions and availability of stocks. You read the product transaction details from an Oracle source and apply lookup condition on the PRODUCTDET table in Snowflake Cloud Data Warehouse which stores details of product and its availability. Based on availability and requirement, you write the transactions to the PENDINGTRANSACTION and COMPLETEDTRANSACTION tables in Snowflake Cloud Data Warehouse and update the INSTOCK field in PRODUCTDET table based on the completed transactions. You use the following objects in the Snowflake Cloud Data Warehouse V2 mapping:

### Source Object

The source object for the mapping task is OracleSrc table in Oracle. Use an Oracle connection to connect to Oracle and read data from the OracleSrc object.

The following image shows the transaction details stored in the OracleSrc table:

transactionID	CustomerID	productID	quantity	OrderPlacedOn
Tran511	CUST21	P45	100	2016-04-05
Tran512	CUST22	P46	200	2016-07-05
Tran513	CUST23	P47	20	2016-07-25
Tran514	CUST24	P47	100	2016-10-25
Tran515	CUST25	P45	1000	2016-12-02
Tran517	CUST27	P46	5000	2017-01-02
Tran516	CUST26	P48	60	2017-01-02
Tran518	CUST28	P49	60	2017-01-03
Tran519	CUST29	P50	700	2017-03-13
Tran520	CUST30	P47	750	2017-03-14

### Lookup Object

The lookup object for the mapping task is PRODUCTDET table in Snowflake Cloud Data Warehouse, which has details of product and its availability.

The following image shows the data stored in the PRODUCTDET table:

Data Preview			
Connection: snowflake_CQA		Object: PRODUCTDET	
PRODUCTID	INSTOCK	PRODUCTDET	PRICE
p45	900	2.5" 80GB IDE Laptop Har...	1968
p46	10000	Laptop Internal CD/DVD R...	1229
p47	5000	New HP ProBook 430 G3 ...	5289
p48	50	New HP ProBook 430 G3 ...	9594
p49	20	Dell Inspiron 15R N5110 B...	1699
p50	800	HP 15-be016TU 15.6-inch...	27490

### Target Object

The mapping task has the following target objects:

#### COMPLETEDTRANSACTION

The COMPLETEDTRANSACTION table includes the TRANSACTIONID, PRODUCTID, QUANTITY, ORDERPLACEDON, and ORDERCOMPLETEDON fields.

The following image shows the data stored in the COMPLETEDTRANSACTION table:

Connection: snowflake_CQA		Object: COMPLETEDTRANSACTION		
TRANSACTIONID	PRODUCTID	QUANTITY	ORDERPLACEDON	ORDERCOMPLETEDON
Tran511	P45	100	2016-04-05 00:00:00.0	2016-04-05 00:00:00.0
Tran512	P46	200	2016-07-05 00:00:00.0	2016-07-05 00:00:00.0
Tran513	P47	20	2016-07-25 00:00:00.0	2016-07-25 00:00:00.0
Tran514	P47	100	2016-10-25 00:00:00.0	2016-10-25 00:00:00.0
Tran517	P46	5000	2017-01-02 00:00:00.0	2017-01-02 00:00:00.0
Tran519	P50	700	2017-03-13 00:00:00.0	2017-03-13 00:00:00.0
Tran520	P47	750	2017-03-14 00:00:00.0	2017-03-14 00:00:00.0

#### PENDINGTRANSACTION

The PENDINGTRANSACTION table includes the PRODUCTID, TRANSACTIONID, REQUIREDQUANTITY, and ORDERPLACEDON fields.

The following image shows the data stored in the PENDINGTRANSACTION table:

Connection: snowflake_CQA		Object: PENDINGTRANSACTION	
PRODUCTID	TRANSACTIONID	REQUIREDQUANTITY	ORDERPLACEDON
P45	Tran515	1000	2016-12-02 00:00:00.0
P48	Tran516	60	2017-01-02 00:00:00.0
P49	Tran518	60	2017-01-03 00:00:00.0

## PRODUCTDET

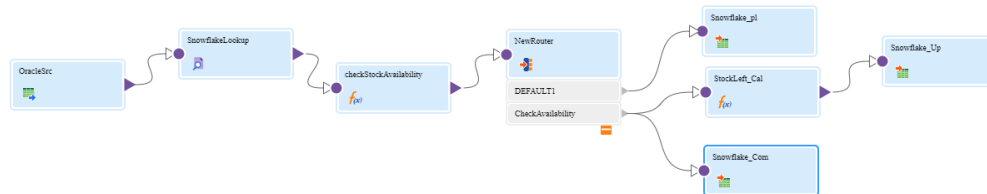
The PRODUCTDET table includes the PRODUCTID, INSTOCK, PRODUCTDET, and PRICE fields. Based on the completed transactions, the INSTOCK field is updated.

The following image shows the data stored in the PRODUCTDET table:

Connection: snowflake_CQA		Object: PRODUCTDET	
PRODUCTID	INSTOCK	PRODUCTDET	PRICE
P48	50	New HP ProBook 430 G3 ...	9594
P49	20	Dell Inspiron 15R N5110 B...	1699
P45	800	2.5" 80GB IDE Laptop Har...	1968
P46	4800	Laptop Internal CD/DVD R...	1229
P47	4130	New HP ProBook 430 G3 ...	5289
P50	100	HP 15-be016TU 15.6-inch...	27490

## Mapping

The following image shows the Snowflake Cloud Data Warehouse V2 mapping:



When you run the mapping, the agent reads the transaction details from source, fetches fields from the lookup, and based on the conditions applied write the available quantity and transaction details to the target tables.

# Rules and guidelines for Cloud Data Warehouse V2 objects

Consider the following rules and guidelines for Snowflake Cloud Data Warehouse V2 objects used as sources, targets, and lookups in mappings:

- You can read or write data of Binary data type, which is in Hexadecimal format.
- You cannot write semi-structured data to the target. For example, XML, JSON, AVRO, or PARQUET data.
- You cannot specify more than one Pre-SQL or Post-SQL query in the source or target transformation.
- The agent reads or writes the maximum float value, which is 1.7976931348623158e+308, as infinity.

- If a Snowflake Cloud Data Warehouse V2 lookup object contains fields with String data type of maximum or default precision and the row size exceeds the maximum row size, the task fails.
- You can use the following formats to specify filter values of Datetime data type:
  - YYYY-MM-DD HH24:MI:SS
  - YYYY/MM/DD HH24:MI:SS
  - MM/DD/YYYY HH24:MI:SS
- When you provide a warehouse name in the connection properties and the mapping properties, the warehouse name in the mapping overrides the warehouse name you specify in the connection. Even though you provide an incorrect warehouse name in the connection properties, the connection is successful. However, before you run the mapping, ensure that you specify the correct warehouse name in the mapping properties.
- When you use a SQL override query to override the custom query used for importing the metadata from Snowflake Cloud Data Warehouse tables, you must specify a fully qualified table name.
- You can read or write to Snowflake Cloud Data Warehouse tables whose table name or field name can contain uppercase, lowercase, and mixed case alphabets, including numbers and special characters. You cannot write data when the Snowflake Cloud Data Warehouse table contains field names with the # and @ characters. If the Secure Agent is installed on Windows, you cannot write data to the Snowflake Cloud Data Warehouse target table when the table names contain the following special characters: /\:\*?"<>|
- When you use the Create Target option to create a table in Snowflake Cloud Data Warehouse, special characters in the column name are replaced by the \_ character.

### Rules and guidelines for query source type

When you configure a SQL query, consider the following points:

- Ensure that the table name that you specify in the query to read from Snowflake Cloud Data Warehouse contains a fully qualified table name.
- Snowflake Cloud Data Warehouse V2 Connector does not support the following features when you use the Query source type option:
  - Filter and sort options.
  - Source partitioning.
  - Advanced properties, except for pre-SQL and post-SQL statements.
- When you configure a pre-SQL or post-SQL query, the query must include a fully qualified table name.

### Snowflake Cloud Data Warehouse V2 targets that receive duplicate values for primary keys from source records

When you perform an update or delete operation in Snowflake Cloud Data Warehouse, and if the records from the source tables contain duplicate primary keys, you must perform one of the following tasks:

- Before you import the target table, define multiple primary keys in the target table.
- Define more than one custom key for the target object using the **Update Columns** option in the advanced target properties.

# Target statistics of processed rows in Snowflake Cloud Data Warehouse V2 write operations

When you run an insert, update, or delete operation, and the Secure Agent successfully applies all rows to the target and does not reject any rows, the state of the task in the **My Jobs** page reflects as Success. If the Secure Agent rejects even one row, the status reflects as Warning. In this case, the **Rows Processed** field in the **My Jobs** page reflects the total number of rows that the Secure Agent processed.

The following image shows the **My Jobs** page that shows the details of the state and the number of processed rows of a Snowflake Cloud Data Warehouse V2 job:

Instance Name	Location	Subtasks	Start Time	End Time	Rows Processed	State
m_rowdata_reject-1	snowflake		Jan 28, 2019, 7:07 AM	Jan 28, 2019, 7:09 AM	40	Success
m_rowdata_rejection-1	snowflake		Jan 28, 2019, 6:49 AM	Jan 28, 2019, 6:50 AM	40	Failed
m_rowdata_insert_reject-2	snowflake		Jan 28, 2019, 6:46 AM	Jan 28, 2019, 6:48 AM	40	Warning
m_rowdata_insert_reject-1	snowflake		Jan 28, 2019, 6:39 AM	Jan 28, 2019, 6:40 AM	40	Warning

To view how many among the processed rows were a success and how many resulted in an error, select the specific instance name and view the **Results** section. You can view the number of success rows and error rows.

The following image shows the details of the Snowflake Cloud Data Warehouse V2 task:

Job Properties

Task Name:

m\_rowdata\_insert\_reject

Instance ID:

1

Task Type:

Mapping

Started By:

snowflake\_mrg through UI

Start Time:

Jan 28, 2019 6:39:05 AM

End Time:

Jan 28, 2019 6:40:20 AM

Duration:

1 minute, 15 seconds

Runtime Environment:

ADAPGAHER001

Secure Agent:

ADAPGAHER001

Results

State:

Warning

Success Rows:

39

Error Rows:

1

Session Log:

Download Session Log

Individual Source/Target Results

Name	Success Rows	Error Rows	Error Message	Actions
<div>Source</div>	40	0		
<div>TSTATS_BATCHROW_TOT1</div>	39	1		

You can also download the session log to get details of the number of output rows, affected rows, applied rows, and rejected rows.

You might also encounter the following scenarios of target statistics for Snowflake Cloud Data Warehouse V2 write operations:

- In insert, update, or delete operation scenarios where the Secure Agent rejects rows due to a constraint violation, a warning appears in the **Job Properties** page. Download the session log to view the target statistics.
- In update or delete operation scenarios where the Secure Agent does not find a match for some records, that number does not reflect in the **My Jobs** page and the session log. For example, if there are 5 input rows and the Secure Agent updates only 4 target rows, the status of the number of processed rows stills reflects as 5. This issue occurs when Snowflake Cloud Data Warehouse V2 does not return an error message for rejected rows.
- In update or delete operation scenarios where the Secure Agent updates or deletes more rows because of a non-unique match, that actual number of updated or deleted records does not reflect both in the **My Jobs** page and in the session log. For example, if there were 5 input records and the Secure Agent updated 10 target rows, the **My Jobs** page reflects only 5 processed rows.
- The number of success rows for the target object in the **Job Properties** page is not updated and remains the same as the number of incoming rows. For example, while writing 5 records to the target, if two records are rejected, the number of success rows still reflects as 5.

# Configuring custom temp directory for local staging

The Secure Agent creates the local staging files in a default temp directory. You can configure a different directory to store the local staging files.

To configure a different directory for the local staging files, perform the following steps:

1. In Administrator, click **Runtime Environments**.  
The Runtime Environments page appears.
2. Select the Secure Agent for which you want to set the custom configuration property.
3. Click **Edit Secure Agent** icon corresponding to the Secure Agent you want to edit in **Actions**.  
The Edit Secure Agent page appears.
4. Select the **Service** as **Data Integration Server** in the **System Configuration Details** section.
5. Select the **Type** as **DTM** in the **System Configuration Details** section.
6. Set the JVM option to `-Djava.io.tmpdir=E:\Snowflake\temp`.
7. Click **Save**.
8. Restart the Secure Agent.

## Data Encryption in Snowflake Cloud Data Warehouse sources and targets

To protect data, Snowflake Cloud Data Warehouse supports end-to-end-encryption. For more information, see the following Snowflake website:

<https://docs.snowflake.net/manuals/user-guide/security-encryption.html#end-to-end-encryption>

## CHAPTER 4

# Mass ingestion tasks with Snowflake Cloud Data Warehouse V2 Connector

This chapter includes the following topics:

- [Mass ingestion task overview, 28](#)
- [Before you begin, 28](#)
- [Snowflake Cloud Data Warehouse V2 targets in mass ingestion tasks, 29](#)
- [Creating a mass ingestion task, 35](#)
- [Viewing mass ingestion task details, 37](#)
- [Running a mass ingestion task, 37](#)
- [Rules and Guidelines for Snowflake Cloud Data Warehouse mass ingestion tasks, 37](#)

## Mass ingestion task overview

Use mass ingestion tasks to upload a large number of files from on-premise and cloud storage systems to Snowflake. The file type for the files that you transfer must be supported by Snowflake Cloud Data Warehouse.

Create a Snowflake Cloud Data Warehouse V2 connection and use the connection to perform a mass ingestion task. When you create a mass ingestion task, select the target connection and specify which files you want to move from the source to the Snowflake Cloud Data Warehouse target.

## Before you begin

Before you create a mass ingestion task, verify that the following conditions exist:

- The organization has the following licenses:
  - Mass Ingestion

- Mass Ingestion Runtime
- The Mass Ingestion application is running on the Secure Agent.
- Source and target connections exist based on the sources from where you want to transfer files and the targets to where you want to transfer files.

## Snowflake Cloud Data Warehouse V2 targets in mass ingestion tasks

In a mass ingestion task, you can configure the Snowflake Cloud Data Warehouse V2 target properties to transfer files from any source that mass ingestion task supports to a Snowflake Cloud Data Warehouse target.

The following table describes the Snowflake Cloud Data Warehouse V2 target properties that you can configure in a mass ingestion task:

Target Property	Description
Connection Type	Type of the target connection. Select <b>Snowflake Cloud Data Warehouse V2</b> as the connection type.
Connection	Select the connection from a list of configured connections.

The following table describes the Snowflake Cloud Data Warehouse V2 advanced target properties that you can configure in a mass ingestion task:

Property	Description
Warehouse	Overrides the Snowflake Cloud Data Warehouse name specified in the Snowflake Cloud Data Warehouse V2 connection.
Database	The database name of Snowflake Cloud Data Warehouse.
Schema	The schema name in Snowflake Cloud Data Warehouse.
Target Table Name	The table name of the Snowflake Cloud Data Warehouse target table. The target table name is case-sensitive.
Role	Overrides the Snowflake Cloud Data Warehouse user role specified in the connection.
Pre SQL	SQL statement to run on the target before the start of the write operation.
Post SQL	SQL statement to run on the target table after the write operation completes.
Truncate Target Table	Truncates the database target table before inserting new rows. Select one of the following options: <ul style="list-style-type: none"> <li>- True. Truncates the target table before inserting all rows.</li> <li>- False. Inserts new rows without truncating the target table</li> </ul> Default is false.

Property	Description
File Format and Copy Options	<p>The copy option and the file format to load the data to Snowflake Cloud Data Warehouse.</p> <p>The copy option specifies the action that the task performs when an error is encountered while loading data from a file:</p> <p>You can specify the following copy option to abort the COPY statement if any error is encountered:</p> <pre>ON_ERROR = ABORT_STATEMENT</pre> <p>When you load files, you can specify the file format as CSV or JSON and define the rules for the data files. The task uses the specified file format and rules while bulk loading data into Snowflake Cloud Data Warehouse tables.</p> <p>Specify the following format:</p> <ul style="list-style-type: none"> <li>- For CSV, specify as <code>file_format = TYPE = {CSV} [formatTypeOptions]</code></li> <li>- For JSON, specify as <code>file_format = TYPE = {JSON} [formatTypeOptions]</code></li> </ul>
External Stage	<p>Specifies the external stage directory to use for loading files into Snowflake Cloud Data Warehouse tables.</p> <p>Ensure that the source folder path you specify is the same as the folder path provided in the URL of the external stage for the specific connection type in Snowflake Cloud Data Warehouse.</p> <p>Applicable when the source for mass ingestion is Microsoft Azure Blob Storage and Amazon S3. The external stage is mandatory when you use the connection type Microsoft Azure Blob Storage V3, but is optional for Amazon S3 V2. If you do not specify an external stage for Amazon S3 V2, Snowflake Cloud Data Warehouse creates an external stage by default.</p>
File Compression	<p>Determines whether or not files are compressed before they are transferred to the target directory.</p> <p>The following options are available:</p> <ul style="list-style-type: none"> <li>- None. Files are not compressed.</li> <li>- GZIP. Files are compressed using GZIP compression.</li> </ul> <p>Applicable for all sources that support the mass ingestion task except for Microsoft Azure Blob Storage V3 and Amazon S3 V2.</p>

## File format and copy options

When you configure a mass ingestion task to transfer a large number of files to Snowflake Cloud Data Warehouse, specify the copy option and the file format to load the data.

Select a Snowflake Cloud Data Warehouse V2 connection in a mass ingestion task and then specify the copy option and the file format in the target options to determine how to load the files to a Snowflake Cloud Data Warehouse target table.

The copy option specifies the action that the task performs when an error is encountered while loading data from a file.

You can specify the following copy option to abort the COPY statement if any error is encountered:

```
ON_ERROR = ABORT_STATEMENT
```

**Note:** The mass ingestion task for Snowflake Cloud Data Warehouse V2 is certified for only the ABORT\_STATEMENT for ON\_ERROR copy option.

When you load files, you can specify the file format as CSV or JSON and define the rules for the data files. The task uses the specified file format and rules while bulk loading data into Snowflake Cloud Data Warehouse tables.

Specify the following format:

- For CSV, specify as `file_format = TYPE = {CSV} [formatTypeOptions]`
- For JSON, specify as `file_format = TYPE = {JSON} [formatTypeOptions]`

The following list describes some of the format type options:

- `RECORD_DELIMITER = '<character>' | NONE`. Single character string that separates records in an input file.
- `FIELD_DELIMITER = '<character>' | NONE`. Specifies the single character string that separates records in an input file.
- `FILE_EXTENSION = '<string>' | NONE`. Specifies the extension for files unloaded to a stage.
- `SKIP_HEADER = <integer>`. Number of lines at the start of the file to skip.
- `DATE_FORMAT = '<string>' | AUTO`. Defines the format of date values in the data files or table.
- `TIME_FORMAT = '<string>' | AUTO`. Defines the format of time values in the data files or table.
- `TIMESTAMP_FORMAT = <string>' | AUTO`. Defines the format of timestamp values in the data files or table.

### Example 1: File format and copy options for loading files from Amazon S3 to Snowflake

You want to create a CSV file format and define the following rules to load files from Amazon S3 to Snowflake::

- Delimit the fields using the pipe character (|).
- Files include a single header line that will be skipped.

Specify the following file format: `file_format = (type = csv field_delimiter = '|' skip_header = 1)`

You can specify both the copy options and file format by using the following character: `&&`

For example, `file_format = (type = csv field_delimiter = ',' skip_header = 2)&&on_error=ABORT_STATEMENT`

### Example 2: File format and copy options for loading data from flat files to Snowflake

You want to load data from a flat file to Snowflake. You must specify additional properties in the file format and copy options field.

For example, specify the file type as CSV, skip the header, and use quote as the text qualifier in the following format:

```
file_format = (type = csv field_delimiter = ',' skip_header = 1 FIELD_OPTIONALLY_ENCLOSED_BY =
'')&&on_error=CONTINUE
```

For more information about the various file formats that you can specify and the copy option, see the Snowflake Cloud Data Warehouse documentation at the following website:

<https://docs.snowflake.net/manuals/sql-reference/sql/copy-into-table.html#copy-options-copyoptions>

## Rules and guidelines for loading JSON files

Consider the following rule and guideline when you load files of the JSON format to Snowflake Cloud Data Warehouse.

When you load files of the JSON format to Snowflake Cloud Data Warehouse, the target table must have only one column of variant type.

To load files of JSON format to columnar format, consider the following tasks:

For example, see the following data view in a table with variant column:

Row	SRC
1	{"email": "adownthwaite0@abc.net.au", "first_name": "Alvera", "gender": "Female", "id": 1, "ip_address": "72.187.204.87", "last_name": "Dowthwaite"}
2	{"email": "juggar1@apache.org", "first_name": "Jenelle", "gender": "Female", "id": 2, "ip_address": "157.128.235.206", "last_name": "Luggar"}
3	{"email": "ddishmon2@ihg.com", "first_name": "Dynah", "gender": "Female", "id": 3, "ip_address": "236.248.69.73", "last_name": "Dishmon"}
4	{"email": "cseer3@facebook.com", "first_name": "Carin", "gender": "Female", "id": 4, "ip_address": "252.93.30.239", "last_name": "Seer"}
5	{"email": "mmacmichael4@ftc.gov", "first_name": "Martha", "gender": "Female", "id": 5, "ip_address": "163.39.79.183", "last_name": "MacMichael"}
6	{"email": "apeckham5@woothemes.com", "first_name": "Ambros", "gender": "Male", "id": 6, "ip_address": "135.186.51.120", "last_name": "Peckham"}
7	{"email": "yptaff5@disqus.com", "first_name": "Yankee", "gender": "Male", "id": 7, "ip_address": "236.35.9.207", "last_name": "Ptaff"}
8	{"email": "aferronier7@army.mil", "first_name": "Abner", "gender": "Male", "id": 8, "ip_address": "27.151.53.0", "last_name": "Feronier"}

To update the table to columnar format, run the following SQL query from the **Post-processing Commands** field in the mapping task:

```
INSERT INTO PERSONS_JSON SELECT parse_json($1):email,
parse_json($1):first_name,
parse_json($1):gender,
parse_json($1):id,
parse_json($1):ip_address,
parse_json($1):last_name from PERSONS_JSON_VARIANT
```

After you run the mapping task, the Secure Agent copies the data in columnar format to Snowflake:

Row	EMAIL	FIRST_NAME	GENDER	ID	IP_ADDRESS	LAST_NAME
1	adownthwaite0@abc.net.au	Alvera	Female	1	72.187.204.87	Dowthwaite
2	juggar1@apache.org	Jenelle	Female	2	157.128.235.206	Luggar
3	ddishmon2@ihg.com	Dynah	Female	3	236.248.69.73	Dishmon
4	cseer3@facebook.com	Carin	Female	4	252.93.30.239	Seer
5	mmacmichael4@ftc.gov	Martha	Female	5	163.39.79.183	MacMichael
6	apeckham5@woothemes.com	Ambros	Male	6	135.186.51.120	Peckham
7	yptaff5@disqus.com	Yankee	Male	7	236.35.9.207	Ptaff
8	aferronier7@army.mil	Abner	Male	8	27.151.53.0	Feronier

## External stage

When you configure a mass ingestion task to load files from a Microsoft Azure Blob Storage or Amazon S3 source to the Snowflake Cloud Data Warehouse tables, specify the external staging directory to use in Snowflake.

You must specify the external stage name for the specific connection type that you want to use in the **Target Options** section in the mass ingestion task.

The external stage field value is mandatory when you run a mass ingestion task to load files from Microsoft Azure Blob Storage to Snowflake Cloud Data Warehouse where the connection type in the source is Microsoft Azure Blob Storage V2. When the source connection type is Amazon S3 V2, and you do not specify an external stage for Amazon S3 V2 in the Snowflake Cloud Data Warehouse target options, Snowflake creates an external stage directory by default.

Ensure that the source directory path in the **Source Options** of the mass ingestion task is the same as the directory path provided in the URL of the external stage created for the Microsoft Azure Blob Storage V2 or Amazon S3 V2 connection in Snowflake Cloud Data Warehouse.

For example, an external stage for Microsoft Azure Blob Storage created using an Azure account name and a blob container with a folder path has the following stage URL: 'azure:///<Blob SAS Token URL/>blob

container>/<folder path>'. The stage uses the file format you specify in the **Target Options** of the mass ingestion task.

The following image shows the stage name and the stage URL for a Microsoft Azure Blob Storage V2 connection in Snowflake Cloud Data Warehouse:

**Create Stage**

Staged files will be stored in the specified Azure location

Name \*

Schema Name  ▼

URL \*

Azure SAS Token

Encryption Master Key

Comment

[Show SQL](#)

In the example, the stage URL is `azure://adapterdevblob.blob.core.windows.net/snowflakemi/MI/` and the external stage name is `MFT_BLOB1`.

When you create a mass ingestion job, in the **Folder Path** field in the **Source Options** of the Microsoft Azure Blob Storage V2 source, specify the following `<Blob Container>/<folder path>` path from the stage URL: `/snowflakemi/MI`

The following image shows the specified source folder path in the **Source Options** section:

• BlobToDW

1 Definition 2 Source 3 Target 4 Schedule

**Source Type**

☒ Source Connection  
☐ File Listener

**Source Connection Details**

Connection Type: \* Microsoft Azure Blob Storage V3

Connection: \* AzureBlob View

Description:

Account Name: adapterdevblob

**Source Options**

Folder Path: \* snowflakemi/MI

☐ Include files from sub-folders

☐ Skip duplicate files

File Pickup: \*

☐ File Pattern Wildcard File Pattern

☒ File Date Greater than or Equal 2019-05-30 00:00:59

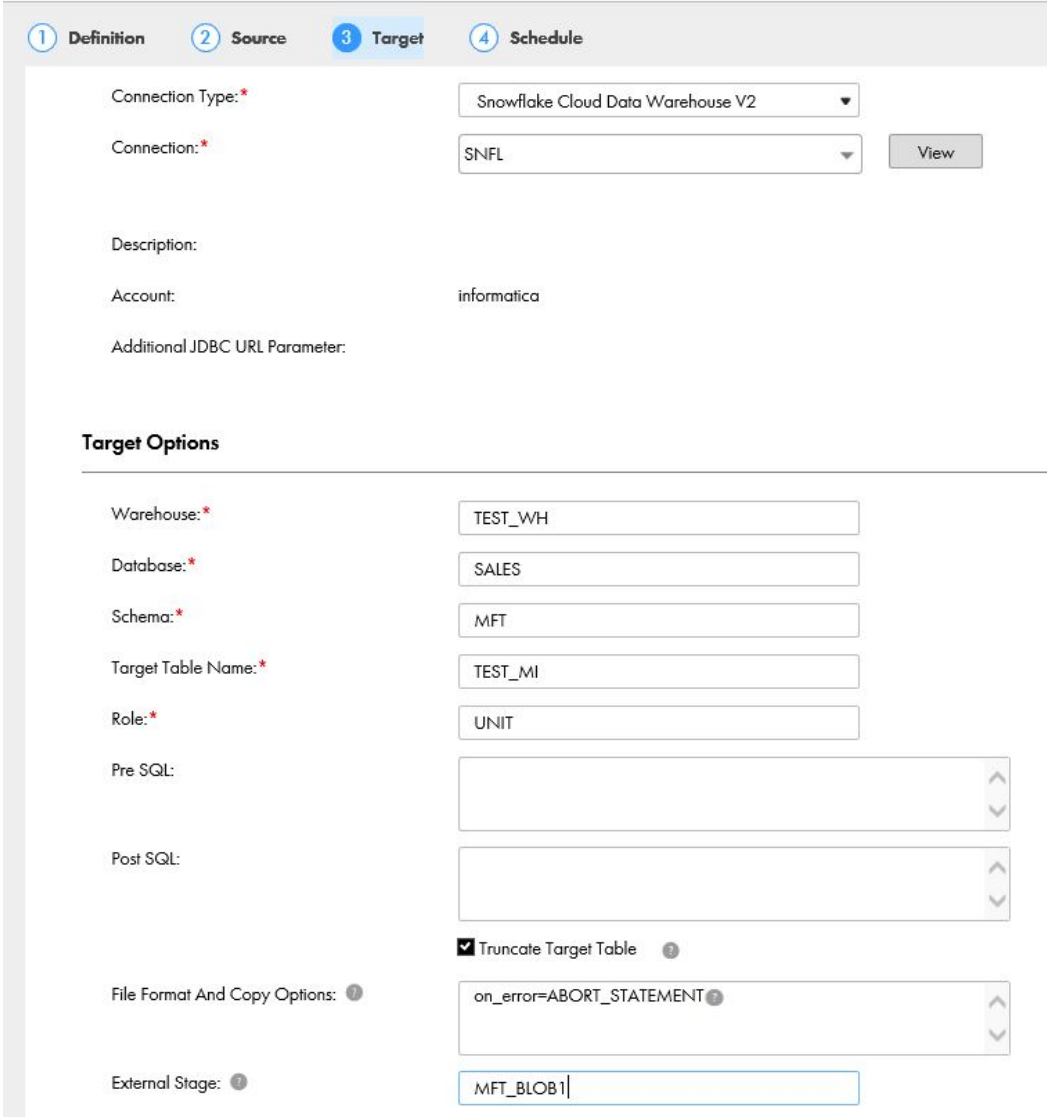
Time Zone: \* Indian Standard Time, Bombay, Delhi

☐ File Size Greater than or Equal File Size Bytes

In the **Target Options** for Snowflake Cloud Data Warehouse V2, specify the following name of the created external stage: MFT\_BLOB1

The following figure shows the configured external stage field in the **Target Options** section:

• s3\_to\_snowflake



1 Definition 2 Source 3 Target 4 Schedule

Connection Type: \* Snowflake Cloud Data Warehouse V2

Connection: \* SNFL View

Description:

Account: informatica

Additional JDBC URL Parameter:

**Target Options**

Warehouse: \* TEST\_WH

Database: \* SALES

Schema: \* MFT

Target Table Name: \* TEST\_MI

Role: \* UNIT

Pre SQL:

Post SQL:

☒ Truncate Target Table ?

File Format And Copy Options: ? on\_error=ABORT\_STATEMENT ?

External Stage: ? MFT\_BLOB1

## Creating a mass ingestion task

You can create a mass ingestion task to transfer files from any source that mass ingestion task supports to a Snowflake Cloud Data Warehouse target.

1. In Data Integration, click **New > Tasks**.
2. Select **Mass Ingestion** and then click **Create**.

The **Definition** tab appears.

3. In the **Definition** tab, configure the following properties:

Property	Description
Task Name	Name of the mass ingestion task. The names of mass ingestion tasks must be unique within the organization. Task names can contain alphanumeric characters, spaces, and underscores. Names must begin with an alphabetic character or underscore. Task names are not case sensitive.
Location	Project folder in which the task resides.
Description	Optional description of the task. Maximum length is 1024 characters.
Runtime Environment	Runtime environment that contains the Secure Agent used to run the task. The Mass Ingestion application must run on the Secure Agent.

4. Click **Next**.

The **Source** tab appears.

5. On the **Source Details** page, select connection from a list of configured connections in the **Connection Type** field.

You can select one of the following sources that mass ingestion task supports:

- Local folder
- Advanced FTP V2
- Advanced FTPS V2
- Advanced SFTP V2
- Amazon S3 V2
- Google Cloud Storage V2
- Microsoft Azure Blob Storage V3
- Hadoop Files V2
- Microsoft Azure Data Lake Store V3
- Azure Data Lake Store Gen2

6. Click **View** to view the connection details.

7. Click **Test** to test the connection in the **View Connection** dialog.

8. Click **Next**.

The **Target** tab appears.

9. On the **Target Details** section, select the **Connection Type** as **Snowflake Cloud Data Warehouse V2** and configure the Snowflake Cloud Data Warehouse target properties.

10. Click **View** to view the connection details.

11. Click **Test** to test the connection in the **View Connection** dialog.

12. Click **Next**.

The **Schedule** tab appears where you can select whether to run the task on a schedule or without a schedule.

13. Click **Run this task on schedule** to run a task on a schedule and select the schedule you want to use.

If you want to remove a task from a schedule, click **Do not run this task on a schedule**.

14. Click **Finish** to save and close the task wizard.

You can edit, run, or view the mass ingestion task on the **Explore** page after you create the mass ingestion task.

## Viewing mass ingestion task details

You can view details about a mass ingestion task, including the source and target connections and the associated schedule.

1. On the **Explore** page, navigate to the task.
2. In the row that contains the task, click **Actions** and select **View**.  
The **Task Details** page appears with task, source, target, and schedule details.
3. You can edit or run the task that you selected to view. On the **Task Details** page, click **Edit** to modify the task or click **Run** to run the task.

## Running a mass ingestion task

You can run a mass ingestion task in the following ways:

1. To run a mass ingestion task manually, on the **Explore** page, navigate to the task. In the row that contains the task, click **Actions** and select **Run**.  
Alternatively, you can run the task manually from the **Task Details** page. To access the **Task Details** page, click **Actions** and select **View**. In the **Task Details** page, select **Run**.
2. To run a mass ingestion task on a schedule, edit the task in the mass ingestion task wizard to associate the task with a schedule.

## Rules and Guidelines for Snowflake Cloud Data Warehouse mass ingestion tasks

When you configure a mass ingestion task to write to Snowflake Cloud Data Warehouse from the supported mass ingestion sources, you must specify a batch size for the maximum number of files to be transferred in a batch. Specify a value for the batch size in the required source properties of the mass ingestion task. When you specify a batch size, the performance of the task is optimized.

The default batch size is 5. When you write from Amazon S3 or Azure Blob Storage sources to a Snowflake target, you can specify a maximum batch size of 1000 in the Amazon S3 or Azure Blob Storage source properties. For other mass ingestion supported sources, you must specify a batch size between 1 and 20.

## CHAPTER 5

# Snowflake pushdown optimization

This chapter includes the following topics:

- [Pushdown optimization, 38](#)
- [Pushdown optimization functions, 39](#)
- [Pushdown Optimization Transformations, 41](#)
- [Configuring a Snowflake ODBC connection, 41](#)
- [Create an ODBC connection, 46](#)
- [Cross-schema pushdown optimization, 48](#)
- [Cross-database pushdown optimization , 49](#)
- [Rules and guidelines for pushdown optimization, 50](#)
- [Troubleshooting, 51](#)

## Pushdown optimization

When you use a Snowflake ODBC connection and select the ODBC subtype as Snowflake, you can configure pushdown optimization in a mapping to push transformation logic to the Snowflake Cloud Data Warehouse source or target database. The ODBC connection must use the Snowflake ODBC driver.

When you run a task configured for pushdown optimization, the task converts the transformation logic to an SQL query. The task sends the query to the database, and the database executes the query. Use pushdown optimization to improve the performance of the task.

You can configure full and source pushdown optimization in a Snowflake Cloud Data Warehouse mapping.

For information about advanced session properties that you can use with pushdown optimization, such as **Allow Temporary View for Pushdown**, **Allow Temporary Sequence for Pushdown**, and **Allow Pushdown for User Incompatible Connections**, see the topic "Advanced Session Properties" under Tasks > Mapping Tasks in the Informatica Cloud Data Integration Help.

# Pushdown optimization functions

The following table summarizes the availability of pushdown functions in a Snowflake Cloud Data Warehouse database. Columns marked with an X indicate that the function can be pushed to the Snowflake Cloud Data Warehouse database by using source-side or full pushdown optimization. Columns marked with a dash (-) symbol indicate that the function cannot be pushed to the database.

Function	Pushdown	Function	Pushdown	Function	Pushdown
ABORT()	-	INITCAP()	X	REG_MATCH()	-
ABS()	X	INSTR()	X	REG_REPLACE	-
ADD_TO_DATE()	X	IS_DATE()	-	REPLACECHR()	X
AES_DECRYPT()	-	IS_NUMBER()	-	REPLACESTR()	X
AES_ENCRYPT()	-	IS_SPACES()	-	REVERSE()	-
ASCII()	X	ISNULL()	X	ROUND(DATE)	-
AVG()	X	LAST()	-	ROUND(NUMBER)	X
CEIL()	X	LAST_DAY()	X	RPAD()	X
CHOOSE()	-	LEAST()	-	RTRIM()	X
CHR()	X	LENGTH()	X	SET_DATE_PART()	-
CHRCODE()	-	LN()	X	SIGN()	X
COMPRESS()	-	LOG()	X	SIN()	X
CONCAT()	X	LOWER()	X	SINH()	X
COS()	X	LPAD()	X	SOUNDEX()	-
COSH()	X	LTRIM()	X	SQRT()	X
COUNT()	X	MAKE_DATE_TIME()	-	STDDEV()	X
CRC32()	-	MAX()	X	SUBSTR()	X
CUME()	-	MD5()	-	SUM()	X
DATE_COMPARE()	X	MEDIAN()	X	SYSDATE()	X
DATE_DIFF()	X	METAPHONE()	-	SYSTIMESTAMP()	X
DECODE()	X	MIN()	X	TAN()	X
DECODE_BASE64()	-	MOD()	X	TANH()	X
DECOMPRESS()	-	MOVINGAVG()	-	TO_BIGINT	X

Function	Pushdown	Function	Pushdown	Function	Pushdown
ENCODE_BASE64()	-	MOVINGSUM()	-	TO_CHAR(DATE)	X
EXP()	X	NPER()	-	TO_CHAR(NUMBER)	X
FIRST()	-	PERCENTILE()	-	TO_DATE()	X
FLOOR()	X	PMT()	-	TO_DECIMAL()	X
FV()	-	POWER()	X	TO_FLOAT()	X
GET_DATE_PART()	X	PV()	-	TO_INTEGER()	X
GREATEST()	-	RAND()	-	TRUNC(DATE)	X
IIF()	X	RANK()	-	TRUNC(NUMBER)	X
IN()	X	RATE()	-	UPPER()	X
INDEXOF()	-	REG_EXTRACT()	-	VARIANCE()	X

The following table lists the pushdown operators that can be used in a Snowflake Cloud Data Warehouse database:

Operator	Pushdown
+	Supported
-	Supported
*	Supported
/	Supported
%	Supported
	Supported
>	Supported
=	Supported
>=	Supported
<=	Supported
!=	Supported
AND	Supported
OR	Supported

Operator	Pushdown
NOT	Supported
^=	Supported

## Pushdown Optimization Transformations

When you configure pushdown optimization, the Secure Agent tries to push the configured transformation to the database.

The following table shows the supported pushdown types for Snowflake database to which you can push the transformation:

Transformations	Pushdown
Aggregator	Source, Full
Expression	Source, Full
Filter	Source, Full
Joiner	Source, Full
Lookup	Source, Full
Sorter	Source, Full
Union	Source, Full
Router	Full
Update Strategy	-

## Configuring a Snowflake ODBC connection

You can set the pushdown optimization for the ODBC connection type that uses Snowflake ODBC driver to enhance the mapping performance. To use an ODBC connection to connect to Snowflake, you must configure the ODBC connection.

After you create a Snowflake ODBC connection, add the **Pushdown Optimization** property under **Advanced Session Properties** tab when you create a mapping task and select **Full** or **To Source** in the **Session Property Value** field. You cannot configure target-side pushdown optimization by using Snowflake ODBC driver. To verify that the pushdown optimization has taken place, you can check the session log for the job. In Monitor, view the log for jobs.

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

## Configuring a Snowflake ODBC connection on Windows

Before you establish an ODBC connection to connect to Snowflake Cloud Data Warehouse on Windows, you must configure the ODBC connection.

Perform the following steps to configure an ODBC connection on Windows:

1. Download the Snowflake ODBC driver from your Snowflake Cloud Data Warehouse account.  
You must download the Snowflake ODBC 64-bit driver.

2. Install the Snowflake ODBC driver on the machine where the Secure Agent is installed.

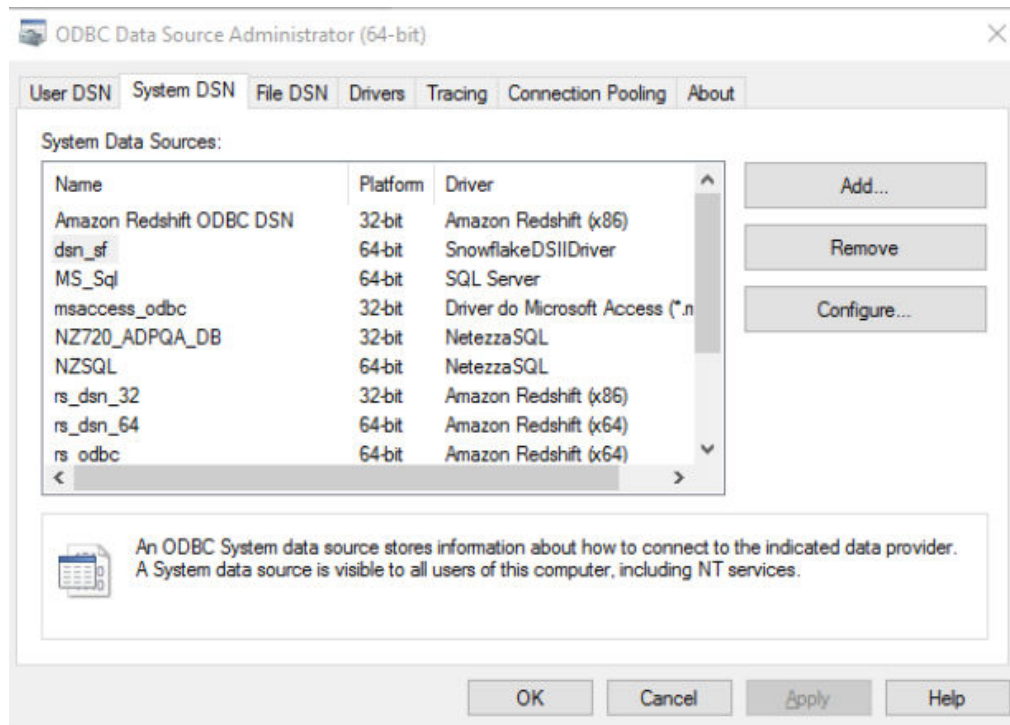
3. Open the folder in which ODBC data source file is installed.

4. Run the `odbcad32.exe` file.

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

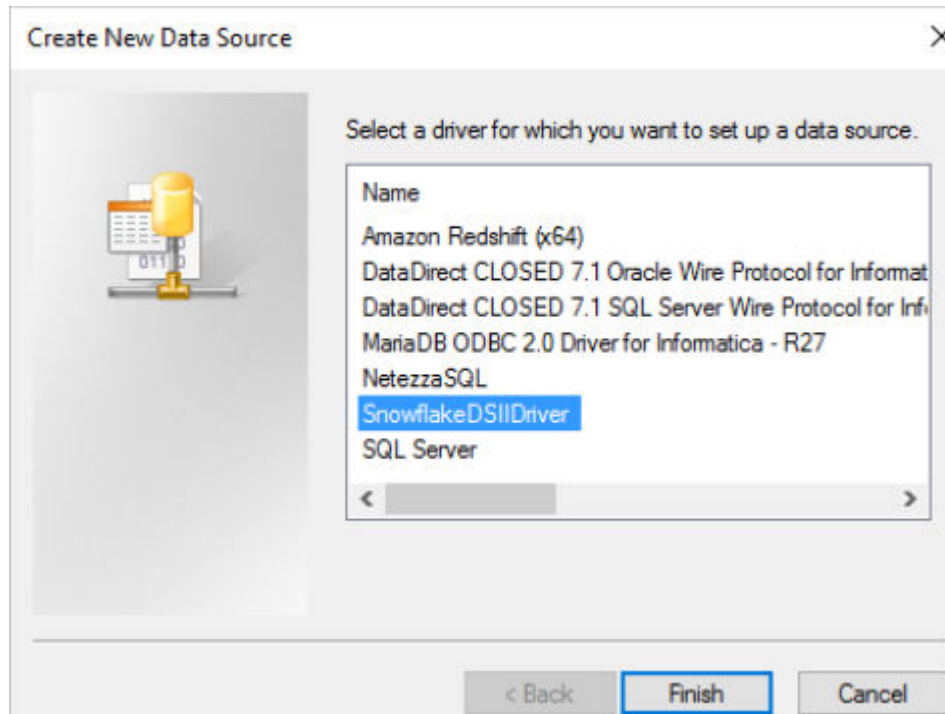
5. Click **System DSN**.

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 Cloud Data Warehouse 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:

The image shows a 'Snowflake Configuration Dialog' window. It contains the following fields and values:

- Data Source: dsn\_sf
- User: INFAADPQA
- Password: (empty)
- Server: (masked with dots)
- Database: CQA
- Schema: CQA\_SCHEMA
- Warehouse: TEST\_WH
- Role: CQA\_ODBCTEST
- Tracing(0-6): 4

At the bottom, there are 'OK' and 'Cancel' buttons.

9. Specify the following connection properties:

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

Property	Description
Role	The Snowflake Cloud Data Warehouse 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**.

The Snowflake ODBC connection is configured successfully on Windows.

After you configure the Snowflake ODBC connection, you must create an ODBC connection to connect to Snowflake Cloud Data Warehouse.

## Configuring a Snowflake ODBC connection on Linux

Before you establish an ODBC connection to connect to Snowflake Cloud Data Warehouse on Linux, you must configure the ODBC connection.

Perform the following steps to configure an ODBC connection on Linux:

1. Download the Snowflake ODBC driver from your Snowflake Cloud Data Warehouse account.  
You must download the Snowflake ODBC 64-bit driver.
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.

Property	Description
Database	Name of the Snowflake database.
Schema	Name of the Snowflake schema.
Warehouse	Name of the Snowflake warehouse.
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.

The Snowflake ODBC connection on Linux is configured successfully.

After you configure the Snowflake ODBC connection, you must create an ODBC connection to connect to Snowflake Cloud Data Warehouse.


## Create an ODBC connection

You must create an ODBC connection to connect to Snowflake Cloud Data Warehouse after you configure the ODBC connection.

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	Username to log in to the Snowflake Cloud Data Warehouse database.

Property	Description
Password	Password to log in to the Snowflake Cloud Data Warehouse database.
Data Source Name	Enter the name of the ODBC data source name that you created for the Snowflake Cloud Data Warehouse database.
Schema	Name of the Snowflake Cloud Data Warehouse 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.

The Snowflake ODBC connection is created successfully.

## Cross-schema pushdown optimization

You can use cross-schema pushdown optimization for a mapping task to read from or write data to Snowflake Cloud Data Warehouse objects associated with different schemas within the same Snowflake Cloud Data Warehouse database.

To use cross-schema pushdown optimization, create two Snowflake Cloud Data Warehouse ODBC connections and specify the schema in each connection. Ensure that the schema in the source connection is different from the schema in the target connection, but both the schemas must belong to the same database. When you configure pushdown optimization for the mapping task, enable cross-schema pushdown optimization in the advanced session properties. By default, the check box is selected.

### Configuring cross-schema optimization for Snowflake Cloud Data Warehouse mapping

Create two Snowflake Cloud Data Warehouse mappings. For example, perform the following steps to configure cross-schema pushdown optimization for a Snowflake Cloud Data Warehouse mapping task:

1. Create the following two Snowflake Cloud Data Warehouse ODBC connections, each defined with a different schema:
  - a. Create an `sf_odbc1` Snowflake Cloud Data Warehouse ODBC connection and specify `CQA_SCHEMA1` schema in the connection properties.
  - b. Create `sf_odbc2` Snowflake Cloud Data Warehouse ODBC connection and specify `CQA_SCHEMA2` schema in the connection properties.
2. Create a Snowflake Cloud Data Warehouse mapping, `m_sf_pdo_acrossSchema`. Perform the following tasks:
  - a. Add a Source transformation and include a Snowflake Cloud Data Warehouse source object and connection `sf_odbc1` to read data using `CQA_SCHEMA1`.
  - b. Add a Target transformation and include a Snowflake Cloud Data Warehouse target object and connection `sf_odbc2` to write data using `CQA_SCHEMA2`.

3. Create a Snowflake Cloud Data Warehouse mapping task, and perform the following tasks:
  - a. Select the configured Snowflake Cloud Data Warehouse mapping, `m_sf_pdo_acrossSchema`.
  - b. In the **Advanced Options** on the **Schedule** tab, add Pushdown Optimization and set the value to Full.
  - c. Select **Enable cross-schema pushdown optimization**.

The following image shows the configured **Enable cross-schema pushdown optimization** property:

Remove	Session Property Name*	Session Property Value*
	Pushdown Optimization	Full

☒ Enable cross-schema pushdown optimization ⓘ

- d. Save the task and click **Finish**.

When you run the mapping task, the Secure Agent reads data from the Snowflake Cloud Data Warehouse source object associated with the `CQA_SCHEMA1` schema and writes data to the Snowflake Cloud Data Warehouse target object associated with `CQA_SCHEMA2` schema.

## Cross-database pushdown optimization

You can use an ODBC connection in a Snowflake mapping to enable cross-database pushdown optimization to run queries on data spread across multiple Snowflake databases.

You can configure cross-database pushdown optimization in the mapping task. You must ensure that the Snowflake source and target transformations in the mapping must use two different ODBC connections that point to different DSN entries.

### Configuring cross-database pushdown optimization

1. Create a Snowflake mapping task.
2. In the **Advanced Options** on the **Schedule** tab, select **Allow Pushdown across Databases** as the **Session Property Value** and set the **Session Property Value** to **Yes**.

The following image shows the cross-database pushdown optimization configuration:

Remove	Session Property Name*	Session Property Value*
	Allow Pushdown across Databases	<input checked="" type="radio"/> Yes <input type="radio"/> No
	Custom Properties	AddQuotesAlwaysPDO=Yes
	Pushdown Optimization	Full

☒ Enable cross-schema pushdown optimization ⓘ

Save  Back Next > Finish Cancel

3. Save the task and click **Finish**.

# Rules and guidelines for pushdown optimization

Use the following rules and guidelines when you configure pushdown to a Snowflake Cloud Data Warehouse database:

- 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.
- To push the TO\_CHAR() function to the Snowflake database, you must define the date and format arguments.
- When you push the SYSTIMESTAMP() and SYSDATE() functions 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 REPLACECHR() or REPLACESTR() function to the Snowflake database, the agent ignores the caseFlag argument.  
For example, both REPLACECHR(false, in\_F\_CHAR, 'a', 'b') and REPLACECHR(true, in\_F\_CHAR, 'a', 'b') return the same value.
- You cannot use millisecond and microsecond values when you push functions to the Snowflake database.
- You cannot push time data types to the Snowflake database.
- To push the TRUNC(DATE) function to the Snowflake database, you must define the date and format arguments.
- You can use only the nanosecond values in the ADD\_TO\_DATE() and TRUNC(DATE) functions.
- To push the TRUNC(DATE) functions to the Snowflake database, you must use the following time formats as arguments:
  - D
  - DD
  - MI
  - MM
  - SS
  - YYYY
- To push the GET\_DATE\_PART() and DATE\_DIFF() functions to the Snowflake database, you must use the following time formats as arguments:
  - D
  - DDD
  - HH
  - MI
  - MM
  - SS
  - YYYY

For example, TRUNC(<datefieldname>, 'dd').

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>

- When you enable pushdown optimization for a mapping to read data from two Snowflake tables and you define a filter condition for a column for one of the Snowflake tables and a similar column is available in the other table, ensure that the column name specified in the filter is fully qualified. For example, specify the column name as `tablename.colname`.
- When you configure a lookup, you can use only 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.
- When you configure an upsert, update, or delete operation with pushdown optimization, ensure that you enable the **Allow Temporary View for Pushdown** property in the session properties of the mapping task.

## Troubleshooting

**When you select the truncate table option for a Snowflake target that contains special characters and enable pushdown optimization, the mapping fails.**

To rectify this issue, see the Knowledge Base article:

<https://kb.informatica.com/howto/6/Pages/23/579219.aspx>

**How do you configure pushdown optimization using an ODBC connection in a mapping task for an upsert or update operation to Snowflake?**

In the advanced session properties on the **Schedule** tab, select **Allow Temporary View for Pushdown** as the session property name and select **Yes** as the session property value.

For more information, see the KB article: <https://informatica.com/solution/23/Pages/71/579225.aspx>

## CHAPTER 6

# Data type reference

This chapter includes the following topics:

- [Data type reference overview, 52](#)
- [Snowflake Cloud Data Warehouse V2 and transformation data types, 52](#)

## Data type reference overview

Cloud Data Integration uses the following data types in Snowflake Cloud Data Warehouse V2 mappings and mapping tasks:

- Snowflake Cloud Data Warehouse V2 native data types appear in the source and target transformations when you choose to edit metadata for the fields.
- Transformation data types. Set of data types that appear in the transformations. These are internal data types based on ANSI SQL-92 generic data types, which the Secure Agent uses to move data across platforms. They appear in all transformations in a mapping.

When the Secure Agent reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the Secure Agent writes to a target, it converts the transformation data types to the comparable native data types.

## Snowflake Cloud Data Warehouse V2 and transformation data types

The following table lists the Snowflake Cloud Data Warehouse V2 data types that Cloud Data Integration supports and the corresponding transformation data types:

Snowflake Cloud Data Warehouse V2 Data Type	Transformation Data Type	Range and Description
BINARY (VARBINARY)	binary	Maximum value: 8,388,60 Default value is 8,388,60.
BOOLEAN	string	A Boolean attribute.

Snowflake Cloud Data Warehouse V2 Data Type	Transformation Data Type	Range and Description
DATE	datetime	Date and time values.
FLOAT (DOUBLE, DOUBLE PRECISION, REAL, FLOAT, FLOAT4, FLOAT8)	double	Floating point numbers with double-precision (64 bit). Maximum value: 1.7976931348623158e+307 Minimum value: -1.79769313486231E+307
NUMBER (DECIMAL, NUMERIC)	decimal	Number with 28-bit precision and scale.
NUMBER (INT, INTEGER, BIGINT, SMALLINT, TINYINT, BYTEINT)	decimal	Number with 28-bit precision and scale as 0. Maximum value: 9.99999999999999E+27 Minimum value: -9.99999999999999E+26
TIME	datetime	Date and time values.
TIMESTAMP_LTZ	datetime	Date and time values.
TIMESTAMP_NTZ (TIMESTAMP_NTZ, datetime)	datetime	Date and time values.
TIMESTAMP_TZ	datetime	Date and time values.
VARCHAR (TEXT, CHAR, CHARACTER, STRING)	string	Maximum value: 16,777,216 Default value is 16,777,216.

# INDEX

## C

Cloud Application Integration community  
URL [5](#)  
Cloud Developer community  
URL [5](#)  
connections  
Snowflake V2 [10](#)  
connector overview [7](#)

## D

Data Integration community  
URL [5](#)  
data types  
native data types [52](#)  
overview [52](#)  
transformation data types [52](#)

## F

filter [11](#)

## I

Informatica Global Customer Support  
contact information [6](#)  
Informatica Intelligent Cloud Services  
web site [5](#)

## L

local staging files  
directory configuration [27](#)  
JVM option [27](#)  
lookup  
database [17](#)  
multiple matches [17](#)  
role [17](#)  
schema [17](#)  
warehouse [17](#)

## M

maintenance outages [6](#)  
mappings  
database [11](#), [13](#)  
example [22](#)  
lookup overview [17](#)  
lookup properties [17](#)  
Post-SQL [11](#), [13](#)

mappings (*continued*)  
Pre-SQL [11](#), [13](#)  
role [11](#), [13](#)  
schema [11](#), [13](#)  
source properties [11](#)  
target properties [13](#)  
warehouse [11](#), [13](#)  
mass ingestion task  
example [35](#)  
overview [28](#)  
running [37](#)  
viewing details [37](#)  
mass ingestion tasks  
prerequisites [28](#)

## N

not parameterized sort [11](#)

## P

parameterized sort [11](#)  
partitioning  
configuring key range partitioning [13](#)  
key range partitioning [13](#)  
pushdown optimization  
activity log [41](#)  
ADD\_TO\_DATE() [50](#)  
advanced session properties [41](#)  
create an ODBC connection [46](#)  
DATE\_DIFF() [50](#)  
datetime format [50](#)  
full pushdown [38](#)  
GET\_DATE\_PART() [50](#)  
ODBC subtype [46](#)  
pushdown functions [39](#)  
pushdown operators [39](#)  
REPLACECHR() [50](#)  
REPLACESTR() [50](#)  
rules and guidelines [50](#)  
source pushdown [38](#)  
SYSDATE() [50](#)  
SYSTEMESTAMP() [50](#)  
target pushdown [38](#)  
TO\_BIGINT() [50](#)  
TO\_CHAR() [50](#)  
TO\_INTEGER() [50](#)  
transformations [41](#)  
TRUNC(DATE) [50](#)

## S

- Snowflake Cloud Data Warehouse lookups
  - mapping tasks [18](#)
- Snowflake Cloud Data Warehouse V2
  - connection properties [10](#)
  - connections [9](#)
  - connector [7](#)
  - lookup [7](#)
  - mapping example [22](#)
  - object types [7](#)
  - source [7](#)
  - target [7](#)
  - task types [7](#)
- Snowflake Cloud Data Warehouse V2 connector
  - datetime format [24](#)
  - rules and guidelines [24](#)
- Snowflake Cloud Data Warehouse V2 targets
  - mass ingestion task [29](#)
  - properties [29](#)
- Snowflake ODBC connection
  - configuration on linux [45](#)
  - configuration on windows [42](#)
  - odbc.ini file [45](#)
  - system DSN [42](#)

- Snowflake ODBC driver [41](#)
- sort [11](#)
- status
  - Informatica Intelligent Cloud Services [6](#)
- system status [6](#)

## T

- transformations
  - pushdown optimization [41](#)
- trust site
  - description [6](#)

## U

- upgrade notifications [6](#)

## W

- web site [5](#)