



Informatica® Cloud Data Integration

Snowflake Connector Guide

© Copyright Informatica LLC 2017, 2019

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, the Informatica logo, Informatica Cloud, and PowerCenter 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.

See patents at <https://www.informatica.com/legal/patents.html>.

DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect") which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

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.

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: 2019-08-12

Table of Contents

Preface	5
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
Chapter 1: Introduction to Snowflake Connector	7
Snowflake Connector Overview.	7
Snowflake Connector Task and Object Types.	7
Administration of Snowflake Connector.	8
Data Integration Hosted Agent.	8
Chapter 2: Snowflake Connections	9
Snowflake Connections Overview.	9
Snowflake connection properties.	9
Chapter 3: Snowflake Pushdown Optimization	11
Pushdown Optimization.	11
Pushdown Optimization Functions.	12
Configuring a Snowflake ODBC Connection.	14
Configuring a Snowflake Cloud Data Warehouse ODBC Connection on Windows.	14
Configuring a Snowflake ODBC Connection on Linux.	17
Create an ODBC Connection.	18
Cross-Schema Pushdown Optimization.	20
Configuring Cross-Schema Optimization for a Snowflake Mapping Task.	20
Rules and Guidelines for Pushdown Optimization.	21
Troubleshooting.	22
Chapter 4: Mappings and Mapping Tasks with Snowflake Connector	23
Snowflake Objects in Mappings.	23
Snowflake Sources in Mappings.	23
Key Range Partitioning.	24
Snowflake Targets in Mappings.	25
Snowflake Lookups in Mappings.	27
Snowflake Mapping Example.	28

Rules and Guidelines for Snowflake Objects.	30
Target Statistics of Processed Rows in Snowflake Write Operations.	31
Configuring Directory for Local Staging Files.	32
Chapter 5: Data Type Reference.	33
Data Type Reference Overview.	33
Snowflake and Transformation Data Types.	33
Index.	35

Preface

The *Data Integration Snowflake Connector Guide* contains information about how to set up and use Snowflake Connector. The guide explains how organization administrators and business users can use Snowflake Connector to read data from or write data to Snowflake.

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.

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 Data Integration editions and applications as well as information about other 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>

To find resources on using Application Integration (the Informatica Cloud Real Time service), access the community at:

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

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/community/collections/cloud_integration

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.

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 the [Informatica Intelligent Cloud Services Status](#) page 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 Connector

This chapter includes the following topics:

- [Snowflake Connector Overview, 7](#)
- [Snowflake Connector Task and Object Types, 7](#)
- [Administration of Snowflake Connector, 8](#)
- [Data Integration Hosted Agent, 8](#)

Snowflake Connector Overview

You can use Snowflake Connector to securely read data from or write data to Snowflake. You can 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 connection and use the connection in mappings and mapping tasks. Create a mapping task to process data based on the data flow logic defined in a mapping.

Example

An enterprise application uses the Oracle database to store the product transaction details such as transactionID, customerID, productID, quantity, and OrderPlacedOn. You need to analyze the completed transactions, pending transactions and availability of stocks. Use Snowflake Connector to create a mapping to read all the transaction records from Oracle source, apply lookup on the Snowflake table, apply conditions, and write the records to Snowflake target for data analysis.

Snowflake Connector Task and Object Types

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

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

Task Type	Source	Target	Lookup
Mapping	Yes	Yes	Yes
Mapping Task	Yes	Yes	Yes

Administration of Snowflake Connector

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

Data Integration Hosted Agent

You can use the Data Integration Hosted Agent (Hosted Agent) as a runtime environment for a Snowflake connection if you have the Cloud Runtime license.

Data Integration Secure Agents are installed locally. As an alternative to installing a Secure Agent, you can use a Hosted Agent. Hosted Agents are hosted at Data Integration hosting facility. The Data Integration hosting facility manages the Hosted Agent runtime environment and the agents that run in it. You cannot add, delete, or configure a Hosted Agent runtime environment. Because you do not install a Hosted Agent, you do not have access to files normally stored in the Secure Agent directory, such as configuration, success, and reject files.

CHAPTER 2

Snowflake Connections

This chapter includes the following topics:

- [Snowflake Connections Overview, 9](#)
- [Snowflake connection properties, 9](#)

Snowflake Connections Overview

Create a Snowflake connection to securely read data from or write data to Snowflake. You can use Snowflake connections to specify sources and targets in mappings and mapping tasks.

You create a Snowflake connection on the Connections page. You can then use the connection in the Mapping Designer when you create a mapping.

Snowflake connection properties

When you set up a Snowflake connection, you must configure the connection properties.

The following table describes the Snowflake connection properties:

Connection property	Description
Runtime Environment	The name of the runtime environment where you want to run the tasks.
Username	The user name to connect to Snowflake account.
Password	The password to connect to Snowflake 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 user.

Connection property	Description
Additional JDBC URL Parameters	<p>The additional connection parameters. You can specify one or more parameters in the following format:</p> <pre data-bbox="496 422 1154 443"><param1>=<value>&<param2>=<value>&<param3>=<value>...</pre> <p>For example:</p> <pre data-bbox="496 506 1045 527">user=jon&warehouse=mywh&db=mydb&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 data-bbox="496 611 1170 632">ProcessConnDB=<DB name>&ProcessConnSchema=<schema_name></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 data-bbox="496 716 1170 737">authenticator=https://<Your_Okta_Account_Name>.okta.com</pre> <p>Note: Microsoft ADFS is not supported.</p> <p>For more information about configuring Okta authentication, see the following website: https://docs.snowflake.net/manuals/user-guide/admin-security-fed-auth-configure-snowflake.html#configuring-snowflake-to-use-federated-authentication</p>
Database/Schema	<p>The Snowflake database and schema name. Specify the parameters in the following format:</p> <pre data-bbox="496 947 850 968"><database name>/<schema name></pre> <p>Note: You must specify both database and schema name. If you specify only database name, source objects do not appear in the Select Source Object window. If you specify only schema name, you get an <code>Invalid Schema</code> exception when you read data.</p>

CHAPTER 3

Snowflake Pushdown Optimization

This chapter includes the following topics:

- [Pushdown Optimization, 11](#)
- [Pushdown Optimization Functions, 12](#)
- [Configuring a Snowflake ODBC Connection, 14](#)
- [Create an ODBC Connection, 18](#)
- [Cross-Schema Pushdown Optimization, 20](#)
- [Rules and Guidelines for Pushdown Optimization, 21](#)
- [Troubleshooting, 22](#)

Pushdown Optimization

You can use pushdown optimization to push transformation logic to source databases or target databases. Use pushdown optimization when you use database resources to improve the performance of the task.

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.

Snowflake Connector supports Full and Source pushdown optimization for the ODBC connection type that uses Snowflake ODBC drivers for mapping.

Note: You need to apply EBF CON-7357 to use Pushdown Optimization option. Contact Informatica Global Customer Support to install the Informatica EBF CON-7357.

Example

You are a sales manager in a rapidly growing manufacturing organization. Your organization stores the product transaction details such as transactionID, customerID, productID, quantity, product_revenue, and OrderDate in Snowflake database. You need to calculate the total revenue generated from the sales of a particular product. Use Snowflake Connector to create a mapping to read all the product revenue details of a particular product from the Snowflake source, apply aggregate function to calculate the total revenue, and write the records to Snowflake target for data analysis.

The organization plans to implement a business intelligence service to build visualization and perform real-time analysis. Therefore, you need to port the vast amount of data stored in Snowflake database to the business intelligence service. You can use Snowflake Connector to read data from Snowflake. To read this

large amount of data, you can use full or source pushdown for the ODBC connection type. Using the ODBC connection type with pushdown optimization enhances the performance.

Pushdown Optimization Functions

The following table summarizes the availability of pushdown functions in a Snowflake database. Columns marked with an X indicate that the function can be pushed to the Snowflake 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	LOOKUP	-	SINH()	X
COS()	X	LOWER()	X	SOUNDEX()	-
COSH()	X	LPAD()	X	SQRT()	X
COUNT()	X	LTRIM()	X	STDDEV()	X
CRC32()	-	MAKE_DATE_TIME()	-	SUBSTR()	X
CUME()	-	MAX()	X	SUM()	X
DATE_COMPARE()	X	MD5()	-	SYSDATE()	X
DATE_DIFF()	X	MEDIAN()	X	SYSTEMSTAMP()	X

Function	Pushdown	Function	Pushdown	Function	Pushdown
DECODE()	X	METAPHONE()	-	TAN()	X
DECODE_BASE64()	-	MIN()	X	TANH()	X
DECOMPRESS()	-	MOD()	X	TO_BIGINT	X
ENCODE_BASE64()	-	MOVINGAVG()	-	TO_CHAR(DATE)	X
EXP()	X	MOVINGSUM()	-	TO_CHAR(NUMBER)	X
FIRST()	-	NPER()	-	TO_DATE()	X
FLOOR()	X	PERCENTILE()	-	TO_DECIMAL()	X
FV()	-	PMT()	-	TO_FLOAT()	X
GET_DATE_PART()	X	POWER()	X	TO_INTEGER()	X
GREATEST()	-	PV()	-	TRUNC(DATE)	-
IIF()	X	RAND()	-	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 database:

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

Operator	Pushdown
AND	Supported
OR	Supported
NOT	Supported
^=	Supported

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.

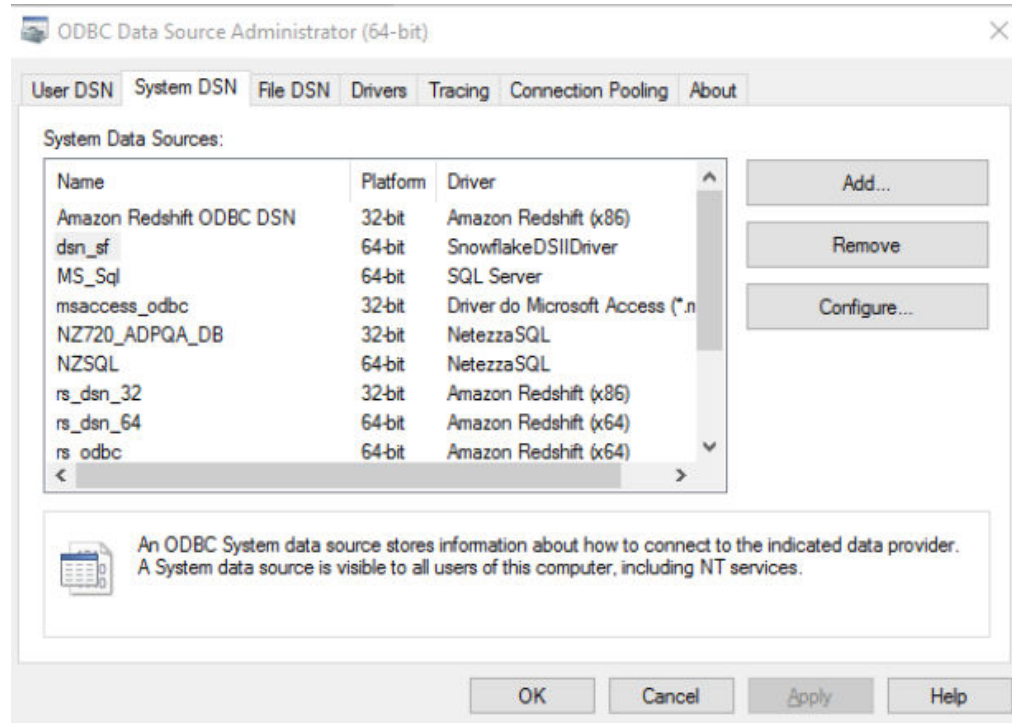
Configuring a Snowflake Cloud Data Warehouse ODBC Connection on Windows

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:

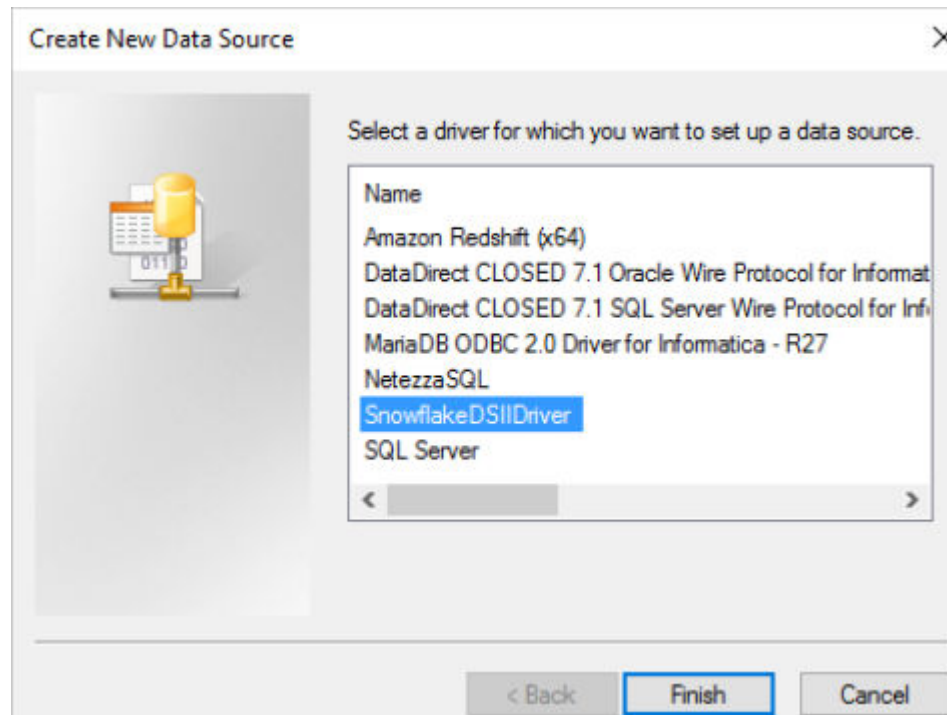
1. Download the Snowflake ODBC driver from your Snowflake account.
You must download the Snowflake ODBC 64-bit driver.
2. Install the Snowflake ODBC driver on the machine where the Secure Agent is installed.
3. Open the folder in which ODBC data source file is installed.
4. Run the `odbcad32.exe` file.
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 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 database.
Password	Password to access the Snowflake database.
Server	Domain name of your Snowflake account.
Database	Name of the Snowflake database.
Schema	Name of the Snowflake schema.
Warehouse	Name of the Snowflake warehouse.

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

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. For more information about how to create an ODBC connection to connect to Snowflake, see [“Create an ODBC Connection” on page 18](#).

Configuring a Snowflake ODBC Connection on Linux

Before you establish an ODBC connection to connect to Snowflake 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 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=
Server=domain_name
role=role
```

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

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

Property	Description
Schema	Name of the Snowflake Cloud Data Warehouse schema.
Warehouse	Name of the warehouse in Snowflake Cloud Data 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.

For more information about how to create an ODBC connection to connect to Snowflake, see [“Create an ODBC Connection” on page 18](#).


Create an ODBC Connection

You must create an ODBC connection to connect to Snowflake 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:

3. 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 ODBC .

4. 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 database.

Property	Description
Password	Password to log in to the Snowflake database.
Data Source Name	Enter the name of the ODBC data source name that you created for the Snowflake database.
Schema	Name of the Snowflake schema.
Code Page	The code page of the database server or flat file defined in the connection.
ODBC Subtype	Enter the value of the ODBC Subtype field as Snowflake .
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 objects associated with different schemas within the same Snowflake database.

To use cross-schema pushdown optimization, create two Snowflake 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.

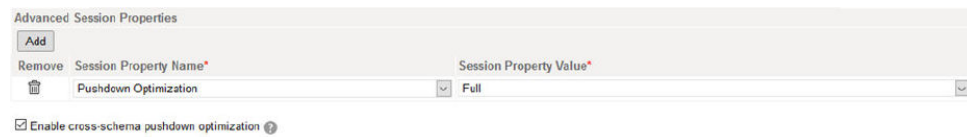
Configuring Cross-Schema Optimization for a Snowflake Mapping Task

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

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



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

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

Rules and Guidelines for Pushdown Optimization

Use the following rules and guidelines when you configure pushdown optimization to a Snowflake database:

- To push the `TRUNC(DATE)` function to the Snowflake database, you must define the date and format arguments.
- 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.
- You can use nanosecond values in the `ADD_TO_DATE()` and `TRUNC(DATE)` functions only.
- To push the `TRUNC(DATE)`, `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>

Troubleshooting

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

You must add the custom property **AddQuotesAlways** and set the value to **Yes** for the Data Integration Server in the Secure Agent properties.

The following image shows the custom configuration that must configure:



Custom Configuration			
Service	Type	Name	Value
Data_Integration_Server55.0	PMRDTM_CFG	AddQuotesAlways	yes

You can then run the Snowflake ODBC mapping with the truncate table option and with pushdown optimization enabled.

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 4

Mappings and Mapping Tasks with Snowflake Connector

This chapter includes the following topics:

- [Snowflake Objects in Mappings, 23](#)
- [Snowflake Mapping Example, 28](#)
- [Rules and Guidelines for Snowflake Objects, 30](#)
- [Target Statistics of Processed Rows in Snowflake Write Operations, 31](#)
- [Configuring Directory for Local Staging Files, 32](#)

Snowflake Objects in Mappings

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

Snowflake Sources in Mappings

In a mapping, you can configure a source transformation to represent a Snowflake source.

You can configure partitioning to optimize the mapping performance at run time when you read data from Snowflake. 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 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. Note: When you use a custom SQL query to import Snowflake 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.

Property	Description
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"> - Not parameterized. Select the fields and type of sorting to use. - Parameterized. Use a parameter to specify the sort option.

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 warehouse name specified in the connection.
Role	Overrides the Snowflake role assigned to user, specified in the connection.
Pre SQL	SQL statement that is executed prior to start of a read operation. 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	SQL statement that is executed after completion of a read operation. For example, if you want to delete some records after the latest records load, specify a Post-SQL statement.
Table Name	Overrides the table name of the imported Snowflake source table.
SQL Override	The SQL statement to override the default query used to read data from the Snowflake 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 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.

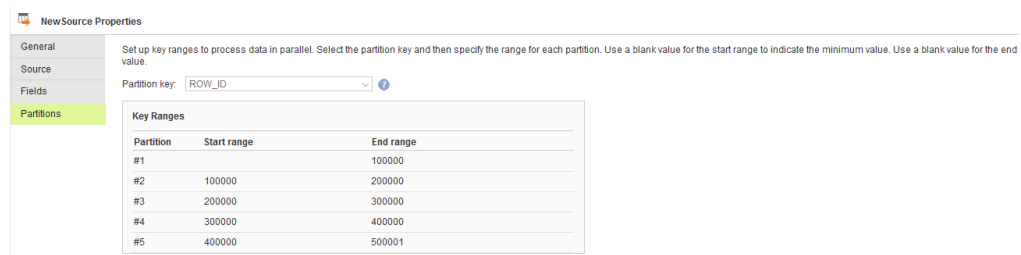
Configuring Key Range Partitioning

Perform the following steps to configure key range partitioning for Snowflake 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:



Snowflake Targets in Mappings

In a mapping, you can configure a target transformation to represent a Snowflake 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 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 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 Target	Creates a target. Enter the table name.

Property	Description
Operation	The target operation. Select Insert, Update, Upsert, or Delete. Note: You cannot use Data Driven operation in Target transformation.
Update columns	The temporary key column to update data to or delete data from a Snowflake target. If you perform an update, update else insert, or delete operation and the Snowflake 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 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	SQL statement that is executed prior to start of a write operation. 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.
Post SQL	SQL statement that is executed after completion of 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.
Batch Row Size	Number of rows that the agent writes in a batch to the Snowflake target.
Number of local staging files	Enter the number of local staging files. The agent writes data to the target, after the specified number of local staging files are created.
Truncate Target Table	Truncates the database target table before inserting new rows. Select one of the following options: <ul style="list-style-type: none"> - True. Truncates the target table before inserting all rows. - False. Inserts new rows without truncating the target table Default is false.
Additional Write Runtime Parameters	Specify additional runtime parameters. For example: <pre>remoteStage=CQA.CQA_SCHEMA.CQA_STAGE</pre> Separate multiple runtime parameters with &.
Table Name	Overrides the table name of the Snowflake target table.
Success File Directory	Not supported.

Advanced Property	Description
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.

Snowflake Lookups in Mappings

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

When you configure a lookup in Snowflake, 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 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 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.

Snowflake 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 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 and update the INSTOCK field in PRODUCTDET table based on the completed transactions. You use the following objects in the Snowflake 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, 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	ORDERCOMPLETEDO
Tran511	P45	100	2016-04-05 00:00:00.0	2016-04-05 00:00:00.0
Tran512	P48	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	P48	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	Tran518	80	2017-01-02 00:00:00.0
P49	Tran518	80	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.

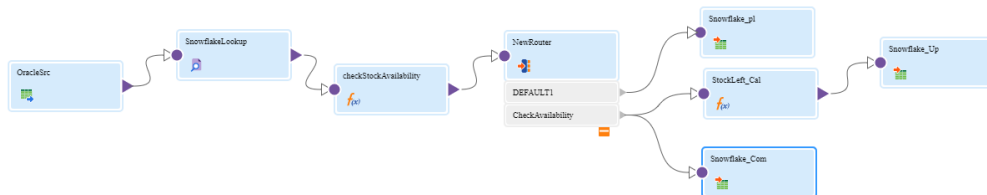
The following image shows the data stored in the PRODUCTDET table:

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
P48	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 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 Snowflake Objects

Consider the following rules and guidelines for Snowflake 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 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 tables, you must specify a fully qualified table name.
- You can read or write to Snowflake 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 table contains field names with the # and @ characters. If the Secure Agent is installed on Windows, you cannot write data to the Snowflake target table when the table names contain the following special characters: \:.*? "<>|
- When you use the Create Target option to create a table in Snowflake, 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 contains a fully qualified table name.
- Snowflake 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 Targets that Receive Duplicate Values for Primary Keys from Source Records

When you perform an update or delete operation in Snowflake, 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.

Snowflake on Azure

You cannot use the user-defined Azure Blob staging location for creating temporary tables.

Target Statistics of Processed Rows in Snowflake 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 job:

Instance Name	Location	Subtasks	Start Time	End Time	Rows Processed	State
m_update_noreject-1	snowflake		Jan 28, 2019, 7:07 AM	Jan 28, 2019, 7:09 AM	40	Success
m_update_norejection-1	snowflake		Jan 28, 2019, 6:49 AM	Jan 28, 2019, 6:50 AM	40	Failed
m_rowsto_insert_reject-2	snowflake		Jan 28, 2019, 6:46 AM	Jan 28, 2019, 6:48 AM	40	Warning
m_rowsto_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 task:

Job Properties		Results	
Task Name:	m_rowsto_insert_reject	State:	Warning
Instance ID:	1	Success Rows:	39
Task Type:	Mapping	Error Rows:	1
Created By:	snowflake_mig through UI	Session Log:	Download Session Log
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:	ADAPGAIHER001		
Secure Agent:	ADAPGAIHER001		

Individual Source/Target Results				
Name	Success Rows	Error Rows	Error Message	Actions
Source	40	0		
TSTATS_BATCHROW_TGT1	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 write operations:

- In insert, update, or delete operation scenarios where the Secure Agent rejects rows due to a constraint violation, the mapping fails. In this case, see the **My Jobs** page to view the total number of processed rows for that job and the session log for error messages.
- 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 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.

Configuring Directory for Local Staging Files

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.

CHAPTER 5

Data Type Reference

This chapter includes the following topics:

- [Data Type Reference Overview, 33](#)
- [Snowflake and Transformation Data Types, 33](#)

Data Type Reference Overview

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

- Snowflake 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 and Transformation Data Types

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

Snowflake 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.
DATE	datetime	Date and time values.

Snowflake Data Type	Transformation Data Type	Range and Description
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 38 bit precision and scale.
NUMBER (INT, INTEGER, BIGINT, SMALLINT, TINYINT, BYTEINT)	decimal	Number with 38 bit precision and scale as 0. Maximum value: 9.999999999999999E+37 Minimum value: -9.999999999999999E+36
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

A

additional write runtime parameters
 csvFileSize [25](#)
 oneBatch [25](#)
 preserveStageFile [25](#)
 remoteStage [25](#)
 useLocalTimezone [25](#)

C

Cloud Application Integration community
 URL [5](#)
Cloud Developer community
 URL [5](#)
connections
 Snowflake [9](#)
connector overview [7](#)

D

Data Integration community
 URL [5](#)
data types
 native data types [33](#)
 overview [33](#)
 transformation data types [33](#)

F

filter [23](#)

H

hosted agent [8](#)

I

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

L

local staging files
 directory configuration [32](#)
 JVM option [32](#)
lookup
 database [27](#)

lookup (*continued*)
 multiple matches [27](#)
 role [27](#)
 schema [27](#)
 warehouse [27](#)

M

maintenance outages [6](#)
mappings
 database [23, 25](#)
 example [28](#)
 lookup overview [27](#)
 lookup properties [27](#)
 Post-SQL [23, 25](#)
 Pre-SQL [23, 25](#)
 role [23, 25](#)
 schema [23, 25](#)
 source properties [23](#)
 target properties [25](#)
 warehouse [23, 25](#)

N

not parameterized sort [23](#)

P

parameterized sort [23](#)
partitioning
 configuring key range partitioning [25](#)
 key range partitioning [24](#)
pushdown optimization
 activity log [14](#)
 ADD_TO_DATE() [21](#)
 advanced session properties [14](#)
 create an ODBC connection [18](#)
 DATE_DIFF() [21](#)
 datetime format [21](#)
 full pushdown [11](#)
 GET_DATE_PART() [21](#)
 ODBC subtype [18](#)
 pushdown functions [12](#)
 pushdown operators [12](#)
 REPLACECHR() [21](#)
 REPLACESTR() [21](#)
 rules and guidelines [21](#)
 source pushdown [11](#)
 SYSDATE() [21](#)
 SYSTIMESTAMP() [21](#)
 target pushdown [11](#)
 TO_BIGINT() [21](#)
 TO_CHAR() [21](#)

pushdown optimization (*continued*)

TO_INTEGER() [21](#)

TRUNC(DATE) [21](#)

S

Snowflake

connection properties [9](#)

connections [9](#)

connector [7](#)

lookup [7](#)

mapping example [28](#)

object types [7](#)

source [7](#)

target [7](#)

task types [7](#)

Snowflake connector

datetime format [30](#)

rules and guidelines [30](#)

Snowflake ODBC connection

configuration on linux [17](#)

configuration on windows [14](#)

odbc.ini file [17](#)

Snowflake ODBC connection (*continued*)

system DSN [14](#)

Snowflake ODBC driver [14](#)

sort [23](#)

status

Informatica Intelligent Cloud Services [6](#)

system status [6](#)

T

trust site

description [6](#)

U

upgrade notifications [6](#)

W

web site [5](#)