



Informatica® PowerExchange for Snowflake  
10.2.2

# User Guide

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.

Informatica, the Informatica logo, PowerCenter, and PowerExchange 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.

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.

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.

# Table of Contents

<b>Preface .....</b>	<b>5</b>
Informatica Resources. ....	5
Informatica Network. ....	5
Informatica Knowledge Base. ....	5
Informatica Documentation. ....	5
Informatica Product Availability Matrices. ....	6
Informatica Velocity. ....	6
Informatica Marketplace. ....	6
Informatica Global Customer Support. ....	6
 <b>Chapter 1: Introduction to PowerExchange for Snowflake.....</b>	 <b>7</b>
PowerExchange for Snowflake Overview. ....	7
Introduction to Snowflake. ....	7
PowerExchange for Snowflake Configuration. ....	8
Prerequisites. ....	8
 <b>Chapter 2: Snowflake Connections.....</b>	 <b>9</b>
Snowflake Connections Overview. ....	9
Snowflake Connection Properties. ....	9
Creating a Snowflake Connection. ....	10
 <b>Chapter 3: PowerExchange for Snowflake Data Objects.....</b>	 <b>11</b>
Snowflake Data Object Overview. ....	11
Snowflake Data Object Properties. ....	11
Snowflake Data Object Read Operation. ....	12
Snowflake Data Object Read Operation Properties. ....	12
Snowflake Data Object Write Operation. ....	13
Snowflake Data Object Write Operation Properties. ....	14
Snowflake Data Object Lookup Operation. ....	15
Snowflake Data Object Lookup Operation Properties. ....	15
Creating a Snowflake Data Object. ....	16
Creating a Snowflake Data Object Operation. ....	17
Adding a Snowflake Data Object Operation as a Snowflake Lookup in a Mapping. ....	17
Rules and Guidelines for Snowflake Mappings. ....	17
 <b>Chapter 4: PowerExchange for Snowflake Mappings.....</b>	 <b>19</b>
PowerExchange for Snowflake Mappings Overview. ....	19
Mapping Validation and Run-time Environments. ....	19
Snowflake Mapping Example. ....	20

**Chapter 5: Snowflake Run-Time Processing..... 21**

Snowflake Run-Time Processing Overview. . . . . 21

Lookup Cache. . . . . 21

Partitioning. . . . . 21

Parameterization for Snowflake Sources. . . . . 22

Parameterization for Snowflake Targets. . . . . 22

**Appendix A: Snowflake Data Type Reference..... 24**

Data Type Reference Overview. . . . . 24

Snowflake and Transformation Data Types. . . . . 24

**Index..... 26**

# Preface

The *Informatica PowerExchange for Snowflake User Guide* provides information about extracting data from and loading data to Snowflake. This guide is written for database administrators and developers who are responsible for developing mappings that extract data from and load data to Snowflake. This guide assumes you have knowledge of Snowflake and Informatica Data Services.

## 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 Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

### 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 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>.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at <https://search.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 Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

## Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at [ips@informatica.com](mailto:ips@informatica.com).

## Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

## Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

# CHAPTER 1

## Introduction to PowerExchange for Snowflake

This chapter includes the following topics:

- [PowerExchange for Snowflake Overview, 7](#)
- [Introduction to Snowflake, 7](#)
- [PowerExchange for Snowflake Configuration, 8](#)

### PowerExchange for Snowflake Overview

You can use PowerExchange for Snowflake to extract data from and load data to Snowflake. You can also read data from and write data to Snowflake that is enabled for staging data in Azure or Amazon.

You can use Snowflake objects as sources and targets in mappings. When you use Snowflake objects in mappings, you must configure properties specific to Snowflake.

You can validate and run Snowflake mappings in the native environment or on the Spark engine in the non-native environment. You can also run profiles against Snowflake objects in the native environment. PowerExchange for Snowflake uses the Snowflake Spark Connector APIs to run Snowflake mappings on the Spark engine. To run mappings in the native environment, PowerExchange for Snowflake uses the Snowflake loader APIs.

#### Example

An enterprise application uses an Oracle database to store the product transaction details such as transactionID, customerID, productID, quantity, and order date. You need to analyze the completed transactions, pending transactions, and availability of stock. Use PowerExchange for Snowflake to create a mapping to extract all the transaction records from the Oracle source, and load the records to a Snowflake target for data analysis.

### Introduction to Snowflake

Snowflake is a cloud data warehouse service that organizations can use to store and analyze data.

Snowflake is a Software-as-a-Service (SaaS) application that uses an SQL database engine with an architecture designed for the cloud.

The Snowflake architecture comprises the following three layers, which sets Snowflake apart from other data warehouses with its additional functionalities and capabilities:

- **Database Storage.** When you load data to Snowflake, Snowflake reorganizes and stores the data in the Snowflake database. You can access the data stored in the Snowflake database through SQL query operations.
- **Query Processing.** Snowflake processes all queries in the query processing layer. The processing layer contains all the compute resources that Snowflake needs to run queries. For example, Snowflake uses CPU, memory, and temporary storage to run queries.
- **Cloud Services.** The cloud services layer contains all the different components that Snowflake needs to process the user requests. For example, Snowflake uses authentication, infrastructure management, and access control services to process user requests.

## PowerExchange for Snowflake Configuration

PowerExchange for Snowflake installs with Informatica. You enable PowerExchange for Snowflake with a license key.

### Prerequisites

Before you use PowerExchange for Snowflake, complete the following prerequisite tasks:

- Install or configure Informatica Services. Verify that the domain has a Data Integration Service and a Model Repository Service.
- Verify that you have write permissions on all the directories within the `<INFA_HOME>` directory.
- Get a license for PowerExchange for Snowflake.
- If you want to run Snowflake mappings on a Kerberos-enabled Hadoop cluster, ensure that you have permissions to create a directory in the cluster.  
If you do not have permissions and you run a Snowflake mapping on a Kerberos-enabled cluster, the Data Integration Service fails to create the cache folder on the cluster and the mapping fails.

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:

<https://network.informatica.com/community/informatica-network/product-availability-matrices>



## CHAPTER 2

# Snowflake Connections

This chapter includes the following topics:

- [Snowflake Connections Overview, 9](#)
- [Snowflake Connection Properties, 9](#)
- [Creating a Snowflake Connection, 10](#)

## Snowflake Connections Overview

Use a Snowflake connection to access a Snowflake database.

Use the Snowflake connection to import Snowflake metadata, create data objects, preview data, and run mappings. When you create a Snowflake connection, you define the connection attributes that the Developer tool uses to connect to the Snowflake database.

Use the Developer tool, Administrator tool, or infacmd to create a Snowflake connection.

## Snowflake Connection Properties

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

**Note:** The order of the connection properties might vary depending on the tool where you view them.

The following table describes the Snowflake connection properties:

Property	Description
Name	The name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters:~`!\$%^&*()-+={ } \:;"'<,>.?/
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. The ID must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.

Property	Description
Description	Optional. The description of the connection. The description cannot exceed 4,000 characters.
Location	The domain where you want to create the connection.
Type	The connection type. Select Snowflake.
Username	The user name to connect to the Snowflake account.
Password	The password to connect to the Snowflake account.
Account	The name of the Snowflake account.
Warehouse	The Snowflake warehouse name.
Role	The Snowflake role assigned to the user.
Additional JDBC URL Parameters	<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 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>

## Creating a Snowflake Connection

Create a Snowflake connection before you create a Snowflake data object.

1. In the Developer tool, click **Window > Preferences**.
2. Select **Informatica > Connections**.
3. Expand the domain in the **Available Connections**.
4. Select the connection type **Database > Snowflake**, and click **Add**.
5. Enter a connection name and an optional description.
6. Select SnowflakeConnection as the connection type.
7. Click **Next**.
8. Configure the connection properties.
9. Click **Test Connection** to verify the connection to Snowflake.
10. Click **Finish**.

## CHAPTER 3

# PowerExchange for Snowflake Data Objects

This chapter includes the following topics:

- [Snowflake Data Object Overview, 11](#)
- [Snowflake Data Object Properties, 11](#)
- [Snowflake Data Object Read Operation, 12](#)
- [Snowflake Data Object Write Operation, 13](#)
- [Snowflake Data Object Lookup Operation, 15](#)
- [Creating a Snowflake Data Object, 16](#)
- [Creating a Snowflake Data Object Operation, 17](#)
- [Adding a Snowflake Data Object Operation as a Snowflake Lookup in a Mapping, 17](#)
- [Rules and Guidelines for Snowflake Mappings, 17](#)

## Snowflake Data Object Overview

A Snowflake data object is a physical data object that uses Snowflake as a source or target. A Snowflake data object is a physical data object that represents data based on a Snowflake resource.

You can configure the data object read and write operation properties that determine how you can read data from Snowflake sources and load data to Snowflake targets.

Create a Snowflake data object from the Developer tool. PowerExchange for Snowflake creates the data object read operation and data object write operation for the Snowflake data object. You can edit the advanced properties of the data object read or write operation and run a mapping.

## Snowflake Data Object Properties

Specify the data object properties when you create the data object.

The following table describes the properties that you configure for the Snowflake data objects:

Property	Description
Name	Name of the Snowflake data object.
Location	The project or folder in the Model Repository Service where you want to store the Snowflake data object.
Connection	Name of the Snowflake connection.

## Snowflake Data Object Read Operation

Create a mapping with a Snowflake data object read operation to read data from Snowflake.

### Snowflake Data Object Read Operation Properties

Snowflake data object read operation properties include run-time properties that apply to the Snowflake data object.

The Developer tool displays advanced properties for the Snowflake data object operation in the Advanced view.

The following table describes the advanced properties for a Snowflake data object read operation:

Property	Description
Database	Overrides the database name specified in the connection.
Schema	Overrides the schema name specified in the connection.
Warehouse	Overrides the Snowflake warehouse name specified in the connection.
Role	Overrides the Snowflake user role specified in the connection. <b>Note:</b> The Data Integration Service ignores the role you specify in a source transformation.
Pre SQL	SQL statement that the Data Integration Service executes before extracting data from the source. For example, if you want to update records in the database before you extract the records from the table, specify a pre-SQL statement.
Post SQL	SQL statement that the Data Integration Service executes after extracting data from the source. 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	Overrides the default SQL query used to extract data from the Snowflake source.

### Query Properties

Use the **Query** tab to specify the join, filter, and sort conditions.

You can configure the following query properties for Snowflake sources:

## Select Distinct

Selects unique values from the source. The Data Integration Service filters out unnecessary data when you use the Snowflake data object in a mapping.

## Join

The Join expression you specify in a read operation to join data from multiple sources in a Snowflake data object.

Select the sources on the **Data Object** tab and define the join condition.

You can use the native or platform expression to join specific records:

- When you configure a platform expression, in the **Tables** section, you can select the outer, inner, left, and right joins to join the tables.
- In the Relationship section, define the relationship for the join:
  - Left Field. The column on which you want to apply the join condition.
  - Operator. The operators that you can use to join tables. You can select one of the following operators: =, >, >=, <, <=, !=
  - Right Field. The value you specify to join the tables.
- When you configure a native expression, select the fields and define a join query syntax in the Join text field. You must specify only the condition and not the type of join in the query. The condition you specify in the text box for the native expression is appended to the join condition.

## Filter

The filter value in a read operation. The filter specifies the WHERE clause of the SELECT statement. Use a filter to reduce the number of rows that the Data Integration Service reads from the Snowflake source. When you enter a source filter, the Integration Service adds a WHERE clause to the default query.

You can use the native or platform expression to filter specific records.

- Configure the platform expression, you can specify the filter using the following fields:
  - Left Field. The column on which you want to apply the filter condition.
  - Operator. The operators that you can use to filter tables. You can select one of the following operators: =, >, >=, <, <=, !=
  - Right Field. The value based on which you want to filter the records.
- When you configure a native expression, select the fields and define a filter query syntax in the Filter text field. The condition you specify in the text box for the native expression is appended to the filter condition.

## Sort

Sorts the rows queried from the Snowflake source. The Data Integration Service adds the ports to the ORDER BY clause in the default query.

# Snowflake Data Object Write Operation

Create a mapping to write data to Snowflake. Use the Snowflake connection, and define the write operation properties to write data to Snowflake.

You can perform insert, update, delete, and upsert operations on a Snowflake target.

## Snowflake Data Object Write Operation Properties

Snowflake data object write operation properties include run-time properties that apply to the Snowflake data object.

The Developer tool displays advanced properties for the Snowflake data object operation in the Advanced view.

The following table describes the Advanced properties for a Snowflake data object write operation:

Property	Description
UpdateMode	Loads data to the target based on the mode you specify. Select one of the following modes: <ul style="list-style-type: none"><li>- Update As Update. Updates all rows flagged for update.</li><li>- Update Else Insert. Updates all rows flagged for update if they exist in the target and then inserts any remaining rows marked for insert.</li></ul>
Database	Overrides the database name specified in the connection.
Schema	Overrides the schema name specified in the connection.
Warehouse	Overrides the Snowflake warehouse name specified in the connection.
Role	Overrides the Snowflake user role specified in the connection. <b>Note:</b> The Data Integration Service ignores the role you specify in a target transformation.
Pre SQL	SQL statement that the Data Integration Service executes before extracting data from the source. For example, if you want to assign sequence object to a primary key field of the target table before you load data to the table, specify a pre-SQL.
Post SQL	SQL statement that the Data Integration Service executes after extracting data from the source. For example, if you want to alter the table created by using create target option and assign constraints to the table before you load data to the table, specify a post-SQL.
Batch Row Size	Number of rows that the Data Integration Service writes in a batch to the Snowflake target.
Number of local staging files	Enter the number of local staging files. The Data Integration Service 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"><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
Additional Write Runtime Parameters	<p>Specify additional run-time parameters.</p> <p>Separate multiple runtime parameters with &amp;.</p> <p>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:</p> <pre>remoteStage=CQA.CQA_SCHEMA.CQA_STAGE</pre> <p><b>Note:</b> You cannot use the user-defined Azure Blob staging location for creating temporary tables.</p> <p>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:</p> <pre>Compression=On</pre> <p>By default, compression is on.</p>
Table Name	Overrides the table name of the Snowflake target table.

## Snowflake Data Object Lookup Operation

Create a mapping with a Snowflake data object lookup operation to lookup data from Snowflake based on a lookup condition.

### Snowflake Data Object Lookup Operation Properties

Snowflake data object lookup operation properties include run-time properties that apply to the Snowflake data object.

The Developer tool displays run-time properties for the Snowflake data object lookup operation in the Run-time view.

The following table describes the run-time properties for a Snowflake data object lookup operation:

Property	Description
Lookup caching enabled	<p>Indicates whether the Data Integration Service caches lookup values.</p> <p>Select <b>Lookup caching enabled</b> to enable lookup caching. When you enable lookup caching, the Integration Service queries the lookup source once, caches the values, and looks up values in the cache. Caching the lookup values can increase performance on large lookup tables.</p> <p>When you disable caching, each time a row passes into the transformation, the Integration Service issues a select statement to the lookup source for lookup values.</p> <p><b>Note:</b> The Integration Service supports lookup caching on the Native and Spark engines, but does not support uncached lookup on the Spark engine.</p>

The Developer tool displays advanced properties for the Snowflake data object lookup operation in the Advanced view.

The following table describes the advanced properties for a Snowflake data object read operation:

Property	Description
Database	Overrides the database name specified in the connection.
Schema	Overrides the schema name specified in the connection.
Warehouse	Overrides the Snowflake warehouse name specified in the connection.
Role	Overrides the Snowflake user role specified in the connection. <b>Note:</b> The Data Integration Service ignores the role you specify in a source transformation.
Pre SQL	SQL statement that the Data Integration Service executes before extracting data from the source. For example, if you want to update records in the database before you extract the records from the table, specify a pre-SQL statement.
Post SQL	SQL statement that the Data Integration Service executes after extracting data from the source. 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.

## Creating a Snowflake Data Object

Create a Snowflake data object to add to a mapping.

1. Select a project or folder in the **Object Explorer** view.
2. Click **File > New > Data Object**.
3. Select **Snowflake Data Object** and click **Next**.  
The **Snowflake Data Object** dialog box appears.
4. Enter a name for the data object.
5. Click **Browse** next to the **Location** option and select the target project or folder.
6. Click **Browse** next to the **Connection** option and select the Snowflake connection from which you want to import the Snowflake object.
7. To add a resource, click **Add** next to the **Selected Resources** option.  
The **Add Resource** dialog box appears.
8. Select the checkbox next to the Snowflake object you want to add and click **OK**.
9. Click **Finish**.  
The data object appears under Data Objects in the project or folder in the **Object Explorer** view.



# Creating a Snowflake Data Object Operation

You can create the data object read, write, or lookup operation for Snowflake data objects. You can add the Snowflake data object operation to a mapping.

1. Select the data object in the **Object Explorer** view.
2. Right-click and select **New > Data Object Operation**.  
The **Data Object Operation** dialog box appears.
3. Enter a name for the data object operation.
4. Select the type of data object operation. You can choose to create a read or write operation.
5. Click **Add**.  
The **Select Resources** dialog box appears.
6. Select the Snowflake data object for which you want to create the data object operation and click **OK**.
7. Click **Finish**.

The Developer tool creates the data object operation for the selected data object.

## Adding a Snowflake Data Object Operation as a Snowflake Lookup in a Mapping

Use a Snowflake lookup to look up data in a Snowflake data object.

1. Open a mapping from the **Object Explorer** view.
2. From the **Object Explorer** view, drag a Snowflake data object read operation to the editor.  
The **Add to Mapping** dialog box appears.
3. Select **Lookup** to add the data object read operation as a lookup in the mapping.
4. Click inside the Snowflake object operation and connect the lookup input ports and the lookup output ports.
5. In the **Properties** view, configure the following parameters:
  - a. On the **General** tab, select the option that you want the Data Integration Service to return when it finds multiple rows that match the lookup condition.
  - b. On the **Lookup** tab, enter the lookup condition properties.
6. When the mapping is valid, click **File > Save** to save the mapping to the Model repository.

## Rules and Guidelines for Snowflake Mappings

Use the following rules and guidelines when you create a mapping:

- Ensure that the source and target table names contain only uppercase letters.
- You cannot use the OR operator in a filter condition.

- You must define a primary key in the target table. If you do not define a primary key in the target table, the mapping fails to delete the record from or update the record in the target table.
- When you run mappings on the Spark engine, the Data Integration Service does not consider the JDBC parameters that you specify in the Snowflake connection and the mapping fails.

## CHAPTER 4

# PowerExchange for Snowflake Mappings

This chapter includes the following topics:

- [PowerExchange for Snowflake Mappings Overview, 19](#)
- [Mapping Validation and Run-time Environments, 19](#)
- [Snowflake Mapping Example, 20](#)

## PowerExchange for Snowflake Mappings Overview

After you create a Snowflake data object read or write operation, you can create a mapping to extract data from a Snowflake source or load data to a Snowflake target.

You can define properties in an operation to determine how the Data Integration Service must extract data from a Snowflake source or load data to a Snowflake target. You can extract data from one or more Snowflake sources, and load data to one or more Snowflake targets. When the Data Integration Service extracts data from the source or loads data to the target, it converts the data based on the data types associated with the source or the target.

## Mapping Validation and Run-time Environments

You can validate and run mappings in the native environment or Spark engine.

The Data Integration Service validates whether the mapping can run in the selected environment. You must validate the mapping for an environment before you run the mapping in that environment.

### Native environment

You can configure the mappings to run in the native or non-native environment. When you run mappings in the native environment, the Data Integration Service processes the mapping and runs the mapping from the Developer tool.

### Spark Engine

When you run mappings on the Spark engine, the Data Integration Service pushes the mapping to a compute cluster and processes the mapping on a Spark engine. The Data Integration Service generates an execution plan to run mappings on the Spark engine.

You can view the plan in the Developer tool before you run the mapping and in the Administrator tool after you run the mapping.

For more information about the non-native environment and Spark engines, see the *Informatica Big Data Management™ Administrator Guide*.

## Snowflake Mapping Example

Your organization has a large amount of customer data from across regions stored in flat files. You organization needs to analyze data in the APAC region. Create a mapping that reads all the customer records from the flat file and write those records to a Snowflake table.

You can use the following objects in a Snowflake mapping:

### Flat file input

The input file is a flat file that contains customer names and their details.

Create a flat file data object. Configure the flat file connection and specify the flat file that contains the customer data as a resource for the data object. Use the data object in a mapping as a read data object.

### Transformations

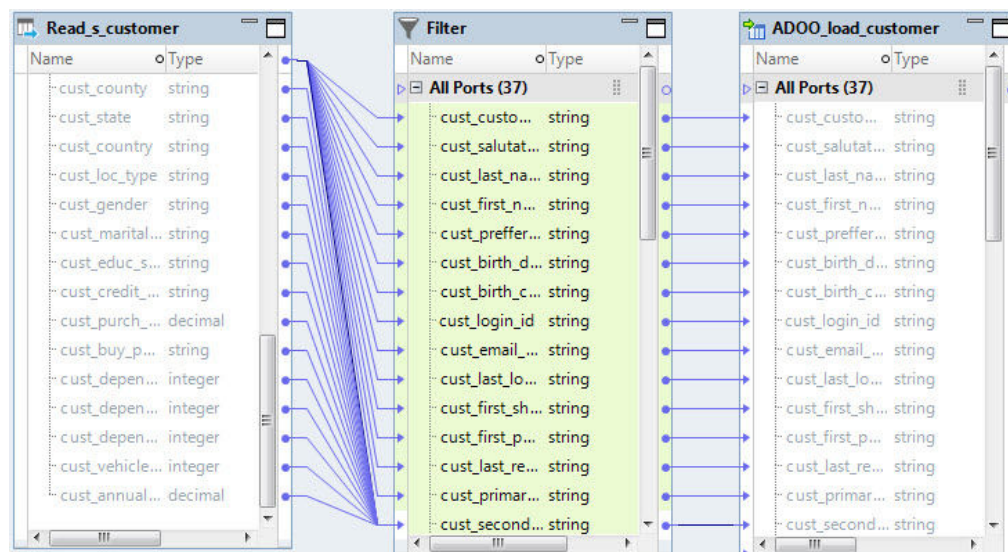
Add a Filter transformation to extract customer data in the APAC region.

The Filter transformation filters the source data based on the value you specify for the region ID column. The Data Integration Service returns the rows that meet the filter condition.

### Snowflake output

Create a Snowflake data object write operation. Configure the Snowflake connection and specify the Snowflake object as a target for the data object. Use the data object in a mapping as a target data object.

The following image shows the Snowflake mapping example:



When you run the mapping, the Data Integration Server reads customer records from the flat file and writes to the Snowflake table.

## CHAPTER 5

# Snowflake Run-Time Processing

This chapter includes the following topics:

- [Snowflake Run-Time Processing Overview, 21](#)
- [Lookup Cache, 21](#)
- [Partitioning, 21](#)
- [Parameterization for Snowflake Sources, 22](#)
- [Parameterization for Snowflake Targets, 22](#)

## Snowflake Run-Time Processing Overview

When you create a Snowflake data object read or write operation, you define properties that determine how the Data Integration Service reads data from or writes data to a Snowflake database.

You can configure lookup caching, partitioning, and parameterization in the run-time properties.

## Lookup Cache

You can enable lookup cache for a Snowflake lookup operation. When you enable lookup caching, the Integration Service caches the lookup source and runs the query on all the rows in the cache. The **Run-time** tab displays the **Lookup Caching Enabled** property that the Data Integration Service uses to lookup data from a Snowflake table. Lookup caching is enabled by default.

When you do not configure lookup caching, the Integration Service queries every input row of the lookup source instead of building and querying the lookup cache. You can enable lookup caching for large lookup tables to improve the performance of the lookup operation.

## Partitioning

When you read data from or write data to Snowflake, you can configure partitioning to optimize the mapping performance at run time. You can configure partitioning for Snowflake mappings that you run in the native or

Spark engine. The partition type controls how the Data Integration Service distributes data among partitions at partition points.

You can define the partition type as key range partitioning. To configure key range partitioning, open the Snowflake data object read or write operation, and select the **Key Range** partition type option on the **Run-time** tab.

When you configure key range partitioning, the Data Integration Service distributes rows of data based on a port or set of ports that you define as the partition key. You can define a range of values for each port. The Data Integration Service uses the key and ranges to send rows to the appropriate partition.

**Note:** A Snowflake source mapping configured for key range partitioning does not work on the Spark engine. If you specify a condition for key range partitioning on the Snowflake source and run the mapping on the Spark engine, the Data Integration Service does not append the condition to the select query at run time.

## Parameterization for Snowflake Sources

You can parameterize the Snowflake connection and data object read operation properties to override the mapping properties at run time.

You can parameterize the following read operation properties for a Snowflake source:

- Database
- Schema
- Warehouse
- Role
- Pre SQL
- Post SQL
- Table Name

## Parameterization for Snowflake Targets

You can parameterize the Snowflake connection and data object write operation properties to override the mapping properties at run time.

You can parameterize the following write operation properties for a Snowflake target:

- Database
- Schema
- Warehouse
- Role
- Pre SQL
- Post SQL
- Table Name
- Batch row size
- Number of local staging files

- Additional write runtime parameters
- Table name

# APPENDIX A

## Snowflake Data Type Reference

This appendix includes the following topics:

- [Data Type Reference Overview, 24](#)
- [Snowflake and Transformation Data Types, 24](#)

### Data Type Reference Overview

Developer Tool uses the following data types in Snowflake mappings:

- Snowflake native data types. Snowflake data types appear in Snowflake definitions in a mapping.
- Transformation data types. Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Data Integration Service uses to move data across platforms. They appear in all transformations in a mapping.

When the Data Integration Service reads source data, it converts the native data types to the comparable transformation data types before transforming the data. When the Data Integration Service 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 Developer Tool 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. <b>Note:</b> You can read and write data of Binary data type only in a native environment.
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, INT, INTEGER, BIGINT, SMALLINT, TINYINT, BYTEINT)	Decimal	Number with 38-bit precision and scale. <b>Note:</b> In the native environment, Decimal values only up to 28-bit precision is supported. PowerExchange for Snowflake does not support Decimal values above 28-bit precision for the source or target because of an SDK limitation. However, you can configure the <code>EnableSDKDecimal38</code> custom flag in the Data Integration Service properties to read or write data of Decimal data type of 38-bit precision and scale.
TIME	Datetime	Date and time values. <b>Note:</b> You can read and write data of Time data type only in a native environment.
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

- create
  - data object operation
    - create [17](#)
  - Snowflake data object [16](#)
- creating
  - Snowflake connection [10](#)

## D

- data object lookup operation
  - properties [15](#)
- data object read operation
  - properties [12](#)
- data object write operation
  - properties [14](#)
- data types [24](#)

## M

- mapping
  - example [20](#)

## N

- native environment
  - mappings [19](#)
- Netezza run-time processing
  - partitioning [22](#)

## P

- PowerExchange for Snowflake
  - EBF [7](#)
  - example [7](#)
  - overview [7](#)
- PowerExchange for Snowflake mappings
  - overview [19](#)
- properties
  - data object lookup operation [15](#)

- properties (*continued*)
  - data object read operation [12](#)
  - data object write operation [14](#)

## S

- snowflake
  - architecture [7](#)
  - cloud data warehouse [7](#)
- Snowflake
  - data object properties [11](#)
  - data object read operation [12](#), [15](#)
  - data object write operation [13](#)
- Snowflake connection
  - properties [9](#)
- Snowflake connections
  - creating [10](#)
  - overview [9](#)
- Snowflake data object
  - create [16](#)
  - overview [11](#)
- Snowflake data types [24](#)
- Snowflake lookup
  - creating [17](#)
- Snowflake parameterization
  - for sources [22](#)
  - for targets [22](#)
- Snowflake read operation properties
  - configuring filter conditions [12](#)
  - configuring join conditions [12](#)
  - configuring sort conditions [12](#)
  - query [12](#)
- Snowflake run-time processing
  - parameterization [22](#)
- Snowflake table
  - lookup cache [21](#)
- Spark engine
  - mappings [19](#)

## T

- transformation data types [24](#)