



Informatica® Cloud
Spring 2017

Snowflake Connector Guide

© Copyright Informatica LLC 2017

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: 2017-11-06

Table of Contents

Preface	4
Informatica Resources.	4
Informatica Network.	4
Informatica Documentation.	4
Informatica Product Availability Matrixes.	4
Informatica Velocity.	5
Informatica Marketplace.	5
Chapter 1: Introduction to Snowflake Connector	6
Snowflake Connector Overview.	6
Snowflake Connector Task and Object Types.	6
Informatica Cloud Hosted Agent.	7
Chapter 2: Snowflake Connections	8
Snowflake Connections Overview.	8
Snowflake Connection Properties.	8
Chapter 3: Mappings and Mapping Configuration Tasks with Snowflake Connector	10
Snowflake Objects in Mappings.	10
Pushdown Optimization.	10
Snowflake Sources in Mappings.	13
Key Range Partitioning.	14
Snowflake Targets in Mappings.	15
Snowflake Lookups in Mappings.	17
Rules and Guidelines for Snowflake Objects.	18
Snowflake Mapping Example.	19
Chapter 4: Data Type Reference	21
Data Type Reference Overview.	21
Snowflake and Transformation Data Types.	21
Index	23

Preface

The *Informatica Cloud 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 Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit <https://network.informatica.com>.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com.

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrixes>.

Informatica Velocity

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

If you are an Informatica Network member, you can access 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.

Informatica Marketplace

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

CHAPTER 1

Introduction to Snowflake Connector

This chapter includes the following topics:

- [Snowflake Connector Overview, 6](#)
- [Snowflake Connector Task and Object Types, 6](#)
- [Informatica Cloud Hosted Agent, 7](#)

Snowflake Connector Overview

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

Note: 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**.

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 Configuration Task	Yes	Yes	Yes

Informatica Cloud Hosted Agent

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

Informatica Cloud Secure Agents are installed locally. As an alternative to installing a Secure Agent, you can use a Hosted Agent. Hosted Agents are hosted at Informatica Cloud hosting facility. The Informatica Cloud 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, 8](#)
- [Snowflake Connection Properties, 8](#)

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 Configuration 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	Enter the user name to connect to Snowflake account.
Password	Enter the password to connect to Snowflake account.
Account	Enter the name of the Snowflake account. In the Snowflake URL, your account name is the first segment in the domain. For example, <code>123abc</code> is your account name in <code>https://123abc.snowflakecomputing.com</code> .
Warehouse	Enter the Snowflake warehouse name.
Role	Specify the Snowflake role assigned to user.

Connection Property	Description
Additional JDBC URL Parameters	<p>Enter one or more JDBC connection parameters in the following format:</p> <pre data-bbox="500 394 1166 420"><param1>=<value>&<param2>=<value>&<param3>=<value>....</pre> <p>For example:</p> <pre data-bbox="500 478 1052 504">user=jon&warehouse=mywh&db=mydb&schema=public</pre>
Database/Schema	<p>Enter the database and schema name in the following format :</p> <pre data-bbox="500 583 857 609"><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

Mappings and Mapping Configuration Tasks with Snowflake Connector

This chapter includes the following topics:

- [Snowflake Objects in Mappings, 10](#)
- [Snowflake Mapping Example, 19](#)

Snowflake Objects in Mappings

When you create a mapping, you can configure a Source or Target transformation to represent a Snowflake object.

Pushdown Optimization

Snowflake Connector supports Full and Source pushdown optimization with an ODBC connection that uses Snowflake ODBC drivers.

To perform pushdown optimization, you must specify the ODBC Subtype connection property as Snowflake in the ODBC connection and use the ODBC connection in the Snowflake Mapping Configuration task.

Note: Snowflake Connector does not support upsert operation in a full pushdown optimization.

Add the **Pushdown Optimization** property under **Advanced Session Properties** tab when you create a Mapping Configuration task and select **Full** or **To Source** in the **Session Property Value** field. You cannot configure target-side pushdown optimization.

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.

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	SYSTIMESTAMP()	X
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

Function	Pushdown	Function	Pushdown	Function	Pushdown
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
AND	Supported
OR	Supported
NOT	Supported
^=	Supported

Rules and Guidelines for Pushdown Optimization Functions

Use the following rules and guidelines when you push functions 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 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>

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.

Note: Ensure that the source table name and field names are not case sensitive.

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, or Parameter.
Object	The source object for the task. Select the source object for a single source. When you select the multiple source option, you can add source objects and configure relationship between them.
Filter	Filters records based on the filter condition. Configure a simple filter.
Sort	Sorts records based on the conditions you specify. You can specify the following sort conditions: <ul style="list-style-type: none"> - 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.
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 Configuration 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.

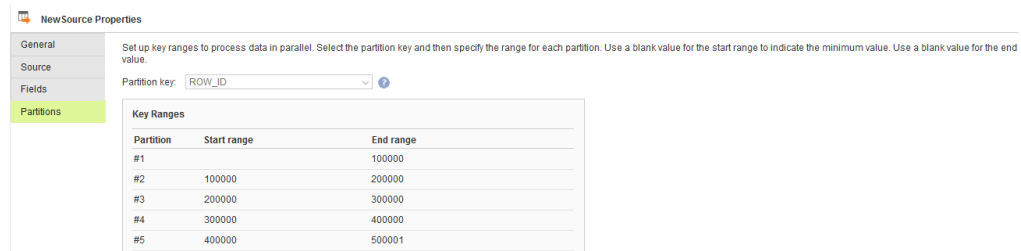
Configuring Key Range Partitioning

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.

Note: Ensure that the target table name and field names are not case sensitive.

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.

Property	Description
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. Note: Make sure that the table name is in all caps.
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 &.
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.

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. Click **Configure > 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.
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 **OK**.
8. Restart the Secure Agent.

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.

Property	Description
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.

Rules and Guidelines for Snowflake Objects

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

- Ensure that the source or target table names and field names are not case sensitive.
- The table name or field name must not contain any special characters other than "_".
- 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 read or write the nanoseconds portion of the 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

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 Configuration 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 Configuration 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 Configuration 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.

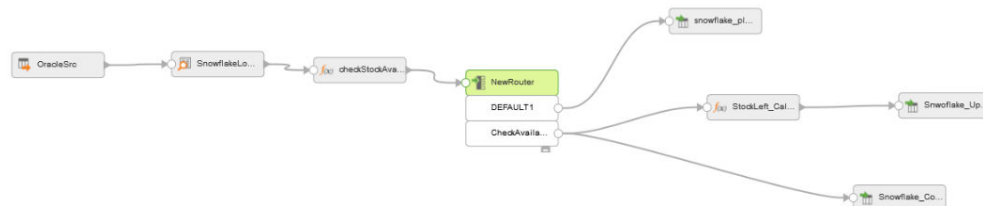
The following image shows the data stored in the PRODUCTDET table:

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.

CHAPTER 4

Data Type Reference

This chapter includes the following topics:

- [Data Type Reference Overview, 21](#)
- [Snowflake and Transformation Data Types, 21](#)

Data Type Reference Overview

Informatica Cloud uses the following data types in Snowflake mappings and Mapping Configuration 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 Informatica Cloud 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 [15](#)
 oneBatch [15](#)
 preserveStageFile [15](#)
 remoteStage [15](#)
 useLocalTimezone [15](#)

C

connection properties
 account [8](#)
 additional JDBC URL parameters [8](#)
 role [8](#)
 schema [8](#)
 warehouse [8](#)
connections
 connection properties [8](#)
connector overview [6](#)

D

data types
 native data types [21](#)
 overview [21](#)
 transformation data types [21](#)

F

filter [13](#)

H

hosted agent [7](#)

I

invalid schema [8](#)

L

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

lookup (*continued*)
 warehouse [17](#)

M

mappings
 database [13, 15](#)
 example [19](#)
 lookup overview [17](#)
 lookup properties [17](#)
 Post-SQL [13, 15](#)
 Pre-SQL [13, 15](#)
 role [13, 15](#)
 schema [13, 15](#)
 source properties [13](#)
 target properties [15](#)
 warehouse [13, 15](#)

N

not parameterized sort [13](#)

P

parameterized sort [13](#)
partitioning
 configuring key range partitioning [15](#)
 key range partitioning [14](#)
pushdown optimization
 ADD_TO_DATE() [13](#)
 DATE_DIFF() [13](#)
 datetime format [13](#)
 full pushdown [10](#)
 GET_DATE_PART() [13](#)
 pushdown functions [10](#)
 pushdown operators [10](#)
 REPLACECHR() [13](#)
 REPLACESTR() [13](#)
 rules and guidelines [13](#)
 source pushdown [10](#)
 SYSDATE() [13](#)
 SYSTEMESTAMP() [13](#)
 target pushdown [10](#)
 TO_BIGINT() [13](#)
 TO_CHAR() [13](#)
 TO_INTEGER() [13](#)
 TRUNC(DATE) [13](#)

S

Snowflake
 connections [8](#)

Snowflake (*continued*)
connector [6](#)
lookup [6](#)
mapping example [19](#)
object types [6](#)
source [6](#)
target [6](#)

Snowflake (*continued*)
task types [6](#)
Snowflake connector
datetime format [18](#)
rules and guidelines [18](#)
sort [13](#)