



Informatica® Data Integration - Free & PayGo

PostgreSQL Connector

© Copyright Informatica LLC 2018, 2023

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: 2023-04-04

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.	5
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 PostgreSQL Connector.....	 7
PostgreSQL Connector assets.	7
 Chapter 2: PostgreSQL connections.....	 8
PostgreSQL connection properties.	8
 Chapter 3: Mappings and mapping tasks with PostgreSQL Connector.....	 11
PostgreSQL sources in mappings.	11
Adding multiple source objects.	13
Rules and guidelines for adding multiple source objects.	13
PostgreSQL lookups in mappings.	14
Rules and guidelines for custom query and SQL override.	14
 Chapter 4: PostgreSQL objects in mapping tasks.....	 16
PostgreSQL sources in mapping tasks.	16
 Chapter 5: Lookup transformation.....	 17
 Chapter 6: SQL transformation.....	 19
Rules and guidelines for stored procedures.	19
 Chapter 7: PostgreSQL pushdown optimization	 20
Pushdown optimization functions and operators.	20
Pushdown optimization operations and transformations.	25
Configuring a PostgreSQL ODBC connection.	25
Configuring a PostgreSQL ODBC connection on Windows.	25
Configuring a PostgreSQL ODBC connection on Linux.	26
Create an ODBC connection.	27

Chapter 8: Data type reference.....	28
PostgreSQL and transformation data types.	28
PostgreSQL and transformation data types supported for create target at runtime option.	29
Rules and guidelines for data types	30
Index.....	31

Preface

Use *PostgreSQL Connector* to learn how to read from PostgreSQL by using Data Integration. Learn to create a PostgreSQL connection, develop mappings, and run mapping and data transfer tasks in Data Integration. You can also learn how to configure pushdown optimization using an ODBC connection.

Informatica Resources

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

Informatica Documentation

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

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

Informatica Intelligent Cloud Services web site

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

Informatica Intelligent Cloud Services Communities

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

Access the Informatica Intelligent Cloud Services Community at:

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

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

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

Informatica Intelligent Cloud Services Marketplace

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

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

Data Integration connector documentation

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

Informatica Knowledge Base

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

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Intelligent Cloud Services Trust Center

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

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

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

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

Informatica Global Customer Support

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

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

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

CHAPTER 1

Introduction to PostgreSQL Connector

You can use PostgreSQL Connector to securely read data from the PostgreSQL and Amazon Aurora PostgreSQL databases.

PostgreSQL sources represent tables in PostgreSQL. You can create a PostgreSQL connection and use the connection in mappings and mapping tasks to read from PostgreSQL. Create a mapping task to process data based on the data flow logic defined in a mapping or integration template.

PostgreSQL Connector assets

Create assets in Data Integration to integrate data using PostgreSQL Connector.

When you use PostgreSQL Connector, you can include the following Data Integration assets:

- Data transfer task
- Mapping
- Mapping task

For more information about configuring assets and transformations, see *Mappings, Transformations, and Tasks* in the Data Integration documentation.

CHAPTER 2

PostgreSQL connections

PostgreSQL connection enables you to read data from PostgreSQL. You can use PostgreSQL connections to specify sources in mappings and mapping tasks.

Create an PostgreSQL connection on the **Connections** page and associate it with a mapping or mapping task. Define the source properties to read from PostgreSQL.

PostgreSQL connection properties

When you set up a PostgreSQL connection, configure the connection properties.

The following table describes the PostgreSQL connection properties:

Property	Description
Connection Name	Name of the connection. Each connection name must be unique within the organization. Connection names can contain alphanumeric characters, spaces, and the following special characters: _ . + -, Maximum length is 255 characters.
Description	Description of the connection. Maximum length is 4000 characters.
Type	Type of connection. Select PostgreSQL from the list.
Runtime Environment	The name of the runtime environment where you want to run the tasks. Specify a Secure Agent or a Hosted Agent.
Host Name	Host name of the PostgreSQL server to which you want to connect.
Port	Port number for the PostgreSQL server to which you want to connect. Default is 5432.
Schema	The schema name. If you don't specify the schema name, all the schemas available in the database are listed while importing the source object in Data Integration.
Database	The PostgreSQL database name.
User Name	User name to access the PostgreSQL database.

Property	Description
Password	Password for the PostgreSQL database user name.
Encryption Method	<p>Determines whether the data exchanged between the Secure Agent and the PostgreSQL database server is encrypted.</p> <p>Select one of the following encryption methods:</p> <ul style="list-style-type: none"> - noEncryption. Establishes a connection without using SSL. Data is not encrypted. - SSL. Establishes a connection using SSL. Data is encrypted using SSL. If the PostgreSQL database server can't configure SSL, the connection fails. - requestSSL. Attempts to establish a connection using SSL. If the PostgreSQL database server can't configure SSL, the Secure Agent establishes an unencrypted connection. <p>Default is noEncryption.</p> <p>Note: SSL is not applicable when you use the Hosted Agent. You can configure SSL when you use the Secure Agent.</p>
Validate Server Certificate	<p>Applicable if you select SSL or requestSSL as the encryption method.</p> <p>Select the Validate Server Certificate option so that the Secure Agent validates the server certificate that is sent by the PostgreSQL database server. If you specify the Host Name In Certificate property, the Secure Agent also validates the host name in the certificate.</p>
TrustStore	<p>Applicable if you select SSL or requestSSL as the encryption method and the Validate Server Certificate option.</p> <p>The path and name of the truststore file, which contains the list of the Certificate Authorities (CAs) that the PostgreSQL client trusts.</p>
TrustStore Password	<p>Applicable if you select SSL or requestSSL as the encryption method and the Validate Server Certificate option.</p> <p>The password to access the truststore file that contains the SSL certificate.</p>
Host Name In Certificate	<p>Optional when you select SSL or requestSSL as the encryption method and the Validate Server Certificate option.</p> <p>A host name for providing additional security. The Secure Agent validates the host name included in the connection with the host name in the SSL certificate.</p>
KeyStore	<p>Applicable if you select SSL as the encryption method and when client authentication is enabled on the PostgreSQL database server.</p> <p>The path and the file name of the key store. The keystore file contains the certificates that the PostgreSQL client sends to the PostgreSQL server in response to the server's certificate request.</p>
KeyStore Password	<p>Applicable if you select SSL as the encryption method and when client authentication is enabled on the PostgreSQL database server.</p> <p>The password for the keystore file required for secure communication.</p>
Key Password	<p>Applicable if you select SSL as the encryption method and when client authentication is enabled on the PostgreSQL database server.</p> <p>Required when individual keys in the keystore file have a different password than the keystore file.</p>

Property	Description
Additional Connection Properties	Additional connection parameters that you want to use. Provide the connection parameters as semicolon-separated key-value pairs.
Crypto Protocol Versions	Required if you select SSL or requestSSL as the encryption method. A cryptographic protocol or a list of cryptographic protocols to use with an encrypted connection. You can select one of the following protocols: <ul style="list-style-type: none"> - SSLv3 - TLSv1_2 Default is TLSv1_2.

CHAPTER 3

Mappings and mapping tasks with PostgreSQL Connector

Use the Data Integration Mapping Designer to create a mapping. When you create a mapping, configure a Source transformation to represent a PostgreSQL object.

Describe the flow of data from source and target along with the required transformations before the agent writes data to the target. When you create a mapping task, select the mapping that you want to use. Use the Mapping Task wizard to create a mapping task. Validate and run the mapping to read data from sources and write to a target. The mapping task processes data based on the data flow logic you define in the mapping.

When you configure a mapping, you can parameterize the source object and the PostgreSQL connection.

PostgreSQL sources in mappings

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

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

Property	Description
Connection	Name of the source connection, or create a connection parameter.
Source type	Type of the source object. Select Single Object, Multiple Objects, Query, or Parameter. When you select multiple objects as the source type to read from multiple PostgreSQL sources, you can use the advanced relationship option to define the relationship for the objects that you want to join. When you select query as the source type, specify the SQL statement in the Query field. You can partially parameterize the query source type. If you want to overwrite the query object at runtime, select the Allow parameter to be overridden at run time option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.

Property	Description
Object	Name of the source object.
Parameter	<p>A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the source object or click New Parameter to define a new parameter for the source object.</p> <p>The Parameter property appears only if you select parameter as the source type.</p> <p>If you want to overwrite the source object at runtime, select the Allow parameter to be overridden at run time option.</p> <p>When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.</p>

The following table describes the PostgreSQL query options that you can configure in a Source transformation:

Property	Description
Filter	<p>Filter value in a read operation. Click Configure to add conditions to filter records and reduce the number of rows that the Secure Agent reads from the source.</p> <p>You can specify the following filter conditions:</p> <ul style="list-style-type: none"> - Not parameterized. Use a basic filter to specify the object, field, operator, and value to select specific records. - Completely parameterized. Use a parameter to represent the field mapping. - Advanced. Use an advanced filter to define a more complex filter condition.
Sort	<p>Add conditions to sort records.</p> <p>You can specify the following sort conditions:</p> <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. - Sort Order. Sorts data in ascending or descending order, according to a specified sort condition.

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

Property	Description
Pre-SQL	<p>The pre-SQL commands to run a query before you read data from PostgreSQL.</p> <p>You can partially parameterize pre-SQL with values specified in a parameter file.</p>
Post-SQL	<p>The post-SQL commands to run a query after you write data to a target.</p> <p>You can partially parameterize post-SQL with values specified in a parameter file.</p>
Fetch Size	<p>Determines the number of rows to read in one resultant set from PostgreSQL. Specifying a number limits the number of rows to fetch with each trip to the database and avoids unnecessary memory consumption.</p> <p>You can specify a maximum fetch size of 2147483647. Default is 100000.</p>
Schema Name	Overrides the schema name of the source object.
Source Table Name	Overrides the default PostgreSQL source table name.

Property	Description
Tracing Level	Sets the amount of details that appear in the log file. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.
SQL Override	The SQL statement to override the default query generated from the specified source type to read data from the PostgreSQL source. You can partially parameterize SQL override with values specified in a parameter file. Ensure that the list of selected columns, data types, and the order of the columns that appear in the query matches the columns, data types, and order in which they appear in the source object.

Adding multiple source objects

When you create a Source transformation, you can select multiple PostgreSQL objects as the source type, and then configure an advanced relationship to combine the tables.

Perform the following steps to join multiple objects in a Source transformation:

1. In the Source transformation, select the **Source Type** as **Multiple Objects**.
2. From the **Actions** menu, select **Add Source Object**.
3. In the **Select Source Object** window, select the source object that you want to add, and then click **OK**.
4. From the **Actions** menu, select **Advanced Relationship**.
5. In the **Advanced Relationship** window, click **Add Object** to add more objects.
6. In the **Select Source Object** window, select the source object with which you want to define a relationship, and then click **OK**.
7. In the **Advanced Relationship** window, select the required fields, and set the conditions or specify a query to define a relationship between the tables.
8. Click **OK**.

Rules and guidelines for adding multiple source objects

Consider the following rules and guidelines when you add multiple source objects:

- You cannot use a self join when you add multiple source objects.
- You cannot search for a schema when you add a related source object. You must scroll down and manually select the schema.
- When you select parent and child objects that have a primary key and foreign key relationship, ensure that the foreign key of the related object is not a primary key in the table.

PostgreSQL lookups in mappings

In a mapping, you can configure a Lookup transformation to represent an PostgreSQL lookup.

The following table describes the PostgreSQL lookup properties that you can configure in a Lookup transformation:

Property	Description
Connection	Name of the lookup connection, or create a connection parameter. If you want to overwrite the lookup connection properties at runtime, select the Allow parameter to be overridden at run time option. Specify the parameter file directory and name in the advanced session properties.
Source type	Type of the lookup object. Select Single Object or Parameter.
Object	Name of the lookup object.
Parameter	A parameter file where you define values that you want to update without having to edit the task. Select an existing parameter for the lookup object or click New Parameter to define a new parameter for the lookup object. The Parameter property appears only if you select parameter as the lookup type. If you want to overwrite the lookup object at runtime, select the Allow parameter to be overridden at run time option. When the task runs, the Secure Agent uses the parameters from the file that you specify in the advanced session properties.
Multiple Matches	The behavior when the lookup condition returns multiple matches. You can return all rows, any row, the first row, the last row, or an error. You can select from the following options in the lookup object properties to determine the behavior: <ul style="list-style-type: none">- Return first row- Return last row- Return any row- Return all rows- Report error
Schema Name	Overrides the schema name of the lookup object.
Source Table Name	Overrides the default PostgreSQL source table name in the Lookup transformation.

Rules and guidelines for custom query and SQL override

Consider the following rules and guidelines when you configure a custom query or SQL override:

- If you use a custom query in the Source transformation and configure create target at runtime in the Target transformation, you must specify the `ExtendedColumnMetadata=true` property in the **Additional Connection Properties** field in the PostgreSQL connection. The property helps retain the constraints from the source in the target object.

- Ensure that the list of selected columns, the data types, and the order of the columns that appear in the query matches the columns, data types, and order in which they appear in the source object.

CHAPTER 4

PostgreSQL objects in mapping tasks

When you configure a mapping task, you can configure advanced properties for PostgreSQL sources.

PostgreSQL sources in mapping tasks

For PostgreSQL source connections used in template-based mapping tasks, you can configure advanced properties in the Sources page.

You can configure the following advanced properties:

Property	Description
Pre-SQL	The pre-SQL commands to run a query before you read data from PostgreSQL. You can partially parameterize pre-SQL with values specified in a parameter file.
Post-SQL	The post-SQL commands to run a query after you write data to a target. You can partially parameterize post-SQL with values specified in a parameter file.
Fetch Size	Determines the number of rows to read in one resultant set from PostgreSQL. Specifying a number limits the number of rows to fetch with each trip to the database and avoids unnecessary memory consumption. You can specify a maximum fetch size of 2147483647. Default is 100000.
Schema Name	Overrides the schema name of the source object.
Source Table Name	Overrides the default PostgreSQL source table name.
Tracing Level	Sets the amount of details that appear in the log file. You can choose terse, normal, verbose initialization, or verbose data. Default is normal.
SQL Override	The SQL statement to override the default query generated from the specified source type to read data from the PostgreSQL source. You can partially parameterize SQL override with values specified in a parameter file. Ensure that the list of selected columns, data types, and the order of the columns that appear in the query matches the columns, data types, and order in which they appear in the source object.

CHAPTER 5

Lookup transformation

You can configure a connected Lookup transformation when you use a PostgreSQL connection in a mapping to return data from a PostgreSQL source based on a specified lookup condition.

You can configure either a cached or uncached lookup to determine whether to cache the lookup data during the runtime session.

When you enable caching, the Data Integration Server queries the lookup source once and caches the values for use during the session. Caching the lookup values improves the session performance. When you disable caching, each time a row passes into the transformation, a SELECT statement gets the lookup values.

When you configure a Lookup transformation, you can specify an SQL statement you want to use to override the default SQL statement for querying lookup values. The Data Integration Server uses the specified lookup SQL override statement and overrides the default SQL statement to query the lookup table. You must use the SQL override with lookup caching enabled.

The Data Integration Server performs the following steps when you run a connected lookup transformation:

1. The Data Integration Server passes values from another transformation to input ports in the Lookup transformation.
2. For each input row, the Data Integration Server queries the lookup source or the cache based on the lookup ports and the lookup condition in the transformation:
 - If the transformation is uncached, the Data Integration Server returns values from the source based on the lookup query.
 - If the transformation is enabled for caching, the Data Integration Server queries the lookup cache during the session and returns the values from the lookup cache.
3. The Data Integration Server then passes the returned data to the next transformation in the mapping.

Configuring a cached or uncached connected Lookup transformation

To configure a connected Lookup transformation, do not select the **Unconnected Lookup** checkbox in the general properties of the Lookup object. To configure a cached or uncached lookup in the Lookup transformation, in the advanced properties of the lookup object, enable or disable the **Lookup Caching Enabled** based on your requirement. The **Lookup Caching Enabled** checkbox is enabled by default.

The default **Lookup Cache Directory Name** is \$PMCacheDir, which is the directory to store the cached lookup data when you select Lookup Caching Enabled. If you do not want a cached lookup, clear the checkbox.

For more information about configuring the lookup properties in a Lookup transformation, see *Transformations*.

Overriding the default lookup query in a Lookup transformation

When you configure a lookup, you specify the lookup condition to query the lookup table in the **Lookup Condition** tab of the Lookup transformation. When you run the mapping, the Data Integration Server finds the data in the lookup source using the query generated from the lookup condition.

If you want to override the lookup query generated from the lookup condition, you can specify an SQL statement to override the lookup query. The Data Integration Server uses the specified lookup SQL override statement to query the lookup table.

To override, you must specify the SQL statement in the **SQL Override** field in the advanced properties of the **Lookup Object** tab. Ensure that you configure the SQL override with lookup caching enabled. To do this, select the **Lookup Caching Enabled** field in the **Advanced** tab of the Lookup transformation.

CHAPTER 6

SQL transformation

You can configure an SQL transformation in a PostgreSQL mapping to call a stored procedure in PostgreSQL.

When you add an SQL transformation to the mapping, on the **SQL** tab, you define the database connection and the type of SQL that the transformation processes.

The stored procedure must exist in the PostgreSQL database before you create the SQL transformation.

For more information about stored procedures, see *Transformations* in the Data Integration help.

Rules and guidelines for stored procedures

Consider the following rules and guidelines for stored procedures:

- You can't use the same name for multiple stored procedures.
- Mappings fail when the stored procedure contains special characters or Unicode characters.
- You can't configure the **On Stored Procedure Error** advanced property in a mapping task.
- When you import data from PostgreSQL that contains the numeric data type, the Secure Agent imports the numeric data type with a default precision of 28 and scale of 26 irrespective of the precision you define for the numeric data type.
- When you import data from PostgreSQL that contains the char, varchar, text and bytea data types, the Secure Agent imports the data types with a precision of 32000 irrespective of the precision you define for the data types.
- If you add a new stored procedure to the database while you have the mapping open, the new stored procedure does not appear in the list of available stored procedures. To refresh the list, close and reopen the mapping.
- You can't use the bytea, time, boolean, citext, and timestamp with timezone data types in a stored procedure.
- To enable the autocommit behavior in a stored procedure, set the `EnableSqlTxAutoCommitSP` custom property value to **true** for the Secure Agent.

CHAPTER 7

PostgreSQL pushdown optimization

You can use pushdown optimization to push the transformation logic to the PostgreSQL database.

When you use an ODBC connection, you can configure pushdown optimization to push transformation logic to PostgreSQL source 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.

Full pushdown optimization is enabled by default in mapping tasks.

Pushdown optimization functions and operators

The following table summarizes the availability of pushdown functions in a PostgreSQL database. Columns marked with an X indicate that the function can be pushed to the PostgreSQL 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	PostgreSQL
ABORT()	-
ABS()	X
ADD_TO_DATE()	X
AES_DECRYPT()	-
AES_ENCRYPT()	-
ASCII()	X
AVG()	X
CEIL()	X

Function	PostgreSQL
CHOOSE()	-
CHR()	X
CHRCODE()	-
COMPRESS()	-
CONCAT()	X
COS()	X
COSH()	-
COUNT()	X
CRC32()	-
CUME()	-
DATE_COMPARE()	-
DATE_DIFF()	X
DECODE()	X
DECODE_BASE64()	-
DECOMPRESS()	-
ENCODE_BASE64()	-
EXP()	X
FIRST()	-
FLOOR()	X
FV()	-
GET_DATE_PART()	-
GREATEST()	-
IIF()	X
IN()	SF
INDEXOF()	-
INITCAP()	X
INSTR()	-

Function	PostgreSQL
IS_DATE()	-
IS_NUMBER()	-
IS_SPACES()	-
ISNULL()	X
LAST()	-
LAST_DAY()	X
LEAST()	-
LENGTH()	X
LN()	X
LOG()	X
LOOKUP	X
LOWER()	X
LPAD()	X
LTRIM()	X
MAKE_DATE_TIME()	-
MAX()	X
MD5()	-
MEDIAN()	-
METAPHONE()	-
MIN()	X
MOD()	X
MOVINGAVG()	-
MOVINGSUM()	-
NPER()	-
PERCENTILE()	-
PMT()	-
POWER()	X

Function	PostgreSQL
PV()	-
RAND()	-
RATE()	-
REG_EXTRACT()	-
REG_MATCH()	-
REG_REPLACE	-
REPLACECHR()	-
REPLACESTR()	-
REVERSE()	-
ROUND(DATE)	-
ROUND(NUMBER)	X
RPAD()	X
RTRIM()	X
SET_DATE_PART()	-
SIGN()	X
SIN()	X
SINH()	-
SOUNDEX()	-
SQRT()	X
STDDEV()	X
SUBSTR()	X
SUM()	X
SYSDATE()	X
SYSTIMESTAMP()	X
TAN()	X
TANH()	-
TO_BIGINT	X

Function	PostgreSQL
TO_CHAR(DATE)	X
TO_CHAR(NUMBER)	X
TO_DATE()	X
TO_DECIMAL()	X
TO_FLOAT()	X
TO_INTEGER()	X
TRUNC(DATE)	-
TRUNC(NUMBER)	X
UPPER()	X
VARIANCE()	X

The following table lists the pushdown operators that can be used in a PostgreSQL database:

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

Operator	Pushdown
NOT	Supported
$\wedge =$	Supported

Pushdown optimization operations and transformations

The following operations are supported for pushdown optimization:

- Insert
- Delete
- Update
- Date driven

The following transformations are supported for pushdown optimization:

- Filter
- Sorter
- Joiner
- Lookup
- Aggregator
- Union
- Expression

Configuring a PostgreSQL ODBC connection

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

After you create a PostgreSQL ODBC connection, navigate to the **Pushdown Optimization** section, and then from the **Pushdown Optimization** list, select **Full** or **To Source**. You cannot configure target-side pushdown optimization by using PostgreSQL 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.

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

Configuring a PostgreSQL ODBC connection on Windows

Before you establish an ODBC connection to connect to PostgreSQL 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 PostgreSQL ODBC 64-bit driver.
2. Install the PostgreSQL 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.
6. Click **Add**.
The **Create New Data Source** dialog appears.
7. Select the PostgreSQL driver and click **Finish**.
8. Click **Configure**.
The PostgreSQL Configuration Dialog appears.
9. Specify the required connection properties.
10. Click **OK**.

The PostgreSQL ODBC connection is configured successfully on Windows.

After you configure the PostgreSQL ODBC connection, you must create an ODBC connection to connect to PostgreSQL.

Configuring a PostgreSQL ODBC connection on Linux

Before you establish an ODBC connection to connect to PostgreSQL on Linux, you must configure the ODBC connection.

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

1. Download the PostgreSQL ODBC 64-bit driver.
2. Install the PostgreSQL ODBC driver on the machine where the Secure Agent is installed.
3. Configure the `odbc.ini` file properties with the required database details.

For example, see the following sample ODBC.ini snippet:

```
[ODBC_PostgreSQL]
Driver=/root/ODBC_Drivers/DWpsql27.so
Description=PostgreSQL DSN
HostName=<hostname>
PortNumber=5432
Database=<databasename>
```

4. Run the following command to export the `odbc.ini` file:
`Export ODBCINI=/<odbc.ini file path>/odbc.ini`
5. Restart the Secure Agent.

The PostgreSQL ODBC connection on Linux is configured successfully.

After you configure the PostgreSQL ODBC connection, you must create an ODBC connection to connect to PostgreSQL.

Create an ODBC connection

You must create an ODBC connection to connect to PostgreSQL after you configure the ODBC connection.

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

1. In Administrator, click **Connections**.
The Connections page appears.
2. Click **New Connection**.
The **New Connection** page appears.
3. Configure the following connection details in the **Connection Details** section:

Property	Description
Connection Name	Name of the ODBC connection. For example, psql_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 PostgreSQL database.
Password	Password to log in to the PostgreSQL database.
Data Source Name	Enter the name of the ODBC data source name that you created for the PostgreSQL database.
Schema	Name of the PostgreSQL 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 PostgreSQL .
Driver Manager for Linux	The driver that the PostgreSQL ODBC driver manager sends database calls to.

The PostgreSQL ODBC connection is created successfully.

CHAPTER 8

Data type reference

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

PostgreSQL native data types

PostgreSQL data types appear in the source transformations when you choose to edit metadata for the fields.

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 Secure Agent uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

When Data Integration reads source data, it converts the native data types to the comparable transformation data types before transforming the data.

PostgreSQL and transformation data types

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

PostgreSQL Data Type	Transformation Data Type	Description
Smallint/Int2	Integer	Precision 10, scale 0
Int/Int4	Integer	Precision 10, scale 0
Bigint/int8	Bigint	Precision 19, scale 0
Decimal	Decimal	Precision 1 to 28, scale 0 to 28
Numeric	Decimal	Precision 1 to 28, scale 0 to 28
Real/Float4	Double	Precision 15, scale 0
Double/Float8	Double	Precision 15, scale 0
Smallserial/Int2	Integer	Precision 10, scale 0
Serial	Integer	Precision 10, scale 0
Bigserial/Serial8	BigInt	Precision 19, scale 0

PostgreSQL Data Type	Transformation Data Type	Description
Char	String	Precision 1
Char(n)	String(n)	n<=10485760
Varchar	String	Precision 104857600
Varchar(n)	String(n)	n <=10485760
Text	String	Precision 104857600
Bytea	Binary	Precision 104857600
Date	Date/Time	Precision 29, scale 9
Time	Date/Time	Precision 29, scale 9
Timestamp	Date/Time	Precision 29, scale 9
Timestamp with time zone	Date/Time	Precision 29, scale 9
Timestamp without time zone	Date/Time	Precision 29, scale 9
Boolean	String	Precision 6
Citext	Text	Precision 104857600

PostgreSQL and transformation data types supported for create target at runtime option

The following table lists the transformation data types that Data Integration supports and the corresponding PostgreSQL data types when you use the **Create New at Runtime** option:

Transformation Data Type	PostgreSQL Data Type
Integer	Integer
Bigint	Bigint
Decimal	Numeric
Double	Double precision
String	Character varying
Text	Text

Transformation Data Type	PostgreSQL Data Type
Binary	Bytea
Date/Time	Timestamp

Rules and guidelines for data types

Consider the following rules and guidelines for PostgreSQL data types:

- The default precision for reading string, text, or bytea data types is 32,000. If you configure a mapping to read data with the text, string, or bytea data types whose precision is more than 32,000, you must manually increase the precision of these data types in the mapping accordingly.
- Citext data type is considered as case-sensitive text in the following scenarios:
 - When you configure a cached lookup and you define the lookup condition for the column of the Citext data type.
 - When you configure an Expression transformation for a column of the Citext data type.

INDEX

C

Cloud Application Integration community
URL [5](#)
Cloud Developer community
URL [5](#)
connections
PostgreSQL [8](#)

D

Data Integration community
URL [5](#)
data type reference
overview [28](#)

I

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

M

maintenance outages [6](#)
mapping
mapping task [11](#)

P

PostgreSQL
assets [7](#)
connection properties [8](#)
Rules and guidelines for joining multiple source objects [13](#)
transformations [7](#)
PostgreSQL and transformation
data types [28](#)
PostgreSQL connections
overview [8](#)
PostgreSQL connector
rules and guidelines [19](#)
PostgreSQL Connector
overview [7](#)

PostgreSQL lookups
mapping [14](#)
PostgreSQL objects
mapping tasks [16](#)
PostgreSQL ODBC connection
configuration on linux [26](#)
configuration on windows [25](#)
odbc.ini file [26](#)
system DSN [25](#)
PostgreSQL ODBC driver [25](#)
PostgreSQL sources
mapping [11](#)
mapping tasks [16](#)
pushdown optimization
activity log [25](#)
advanced session properties [25](#)
create an ODBC connection [27](#)
full pushdown [20](#)
ODBC subtype [27](#)
pushdown functions [20](#), [25](#)
pushdown operators [20](#), [25](#)
source pushdown [20](#)
target pushdown [20](#)

S

status
Informatica Intelligent Cloud Services [6](#)
system status [6](#)

T

trust site
description [6](#)

U

upgrade notifications [6](#)

W

web site [5](#)