Informatica® Data Replication
9.7.0

Datatype Mapping Reference

Informatica Data Replication Datatype Mapping Reference
9.7.0
July 2017

Publication Date: 2018-07-23

Table of Contents

Preface

This *Informatica Data Replication Datatype Mapping Reference* describes datatype mappings that Informatica® Data Replication supports for different source and target database combinations.

This guide is intended for system administrators, DBAs, and those who are responsible for configuring and administering data replication jobs.

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 Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

To access the Knowledge Base, visit https://kb.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 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-matrices.

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.

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

CHAPTER 1

Overview

This chapter includes the following topics:

Mapping Reference Overview

The *Data Replication Datatype Mapping Reference* describes recommended datatype mappings for all combinations of supported sources and supported targets other than flat files and target types based on flat files. This reference also includes recommended datatype mappings for the virtual source columns that you use to define Tcl scripts or SQL expressions.

Data Replication uses these datatype mappings to generate a target schema based on a source schema and to replicate CREATE TABLE and ADD COLUMN DDL changes. Data Replication defines the datatype mappings in the *DataReplication_installation*\uiconf\DataTypes.xml file.

If you create target tables manually, Informatica recommends that you use the target datatypes that are documented in this reference for your source datatypes to perform accurate change data replication.

If a source datatype is not listed, Data Replication either cannot extract data from the source columns with this datatype or cannot apply the extracted data to any appropriate target datatype.

Datatype Mapping Variables and Operators

The datatype mappings tables in this publication include expressions for calculating the length of target character datatypes, when the lengths of the target character datatype and source character datatype do not match. Also, if Data Replication allows a source datatype to be mapped to different target datatypes, the datatype mapping tables include mapping conditions for determining the target datatype to use based on

precision, scale, or length. To specify these calculations and mapping conditions, Data Replication uses variables and operators.

The following table describes the variables that the datatype mapping tables use to describe mapping conditions and length calculations for target columns:

Variable	Description
n	The size of the source character datatype.
n'	The size of the target character datatype if it is not equal to the source character datatype.
p	The precision of the source numeric datatype.
p'	The precision of the target numeric datatype if it is not equal to the source numeric datatype.
s	The scale of the source numeric datatype.
s'	The scale of the target numeric datatype if it is not equal to the source numeric datatype.
SrcMinCharsetSize	The minimum number of bytes per character for the source character set.
TgtCharsetSize	The number of bytes per character for the target character set

The following table describes the operators that the datatype mapping tables use to describe mapping conditions and length calculations for target columns:

Operator	Description
+	Addition. Adds the values that are on both sides of the operator.
-	Subtraction. Subtracts the right-hand operand from the left-hand operand.
x	Multiplication. Multiplies values on both sides of the operator.
/	Division. Divides the left-hand operand by the right-hand operand.

The following table describes the comparison operators that the datatype mapping tables use to define mapping conditions:

Operator	Description
==	Verifies whether the two operands are equal.
!=	Verifies whether the two operands are not equal.
>	Verifies whether the left-hand operand is greater than the right-hand operand.
<	Verifies whether the left-hand operand is less than the right-hand operand.

Operator	Description
>=	Verifies whether the left-hand operand is greater than or equal to the right-hand operand.
<=	Verifies whether the left-hand operand is less than or equal to the right-hand operand.

CHAPTER 2

Datatype Mappings for DB2 for Linux, UNIX, and Windows Sources

This chapter includes the following topics:

DB2 for Linux, UNIX, and Windows Source and Amazon Redshift Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and an Amazon Redshift target:

DB2 Source Datatype	Amazon Redshift Target Datatype	Comments
BIGINT	BIGINT	-
CHARACTER(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-

DB2 Source Datatype	Amazon Redshift Target Datatype	Comments

CLOB(*n*)	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
		-

DATE	DATE	-

DBCLOB(*n*)	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
		-

DECFLOAT	VARCHAR(45)	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMERIC(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	DOUBLE PRECISION	When replicating an 8-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
GRAPHIC(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-
INTEGER	INTEGER	-

DB2 Source Datatype	Amazon Redshift Target Datatype	Comments
LONG VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARGRAPHIC(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
REAL	REAL	When replicating a 4-byte floating-point numeric value to a REAL column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLINT	SMALLINT	-
TIME	TIMESTAMP	When replicating a TIME value to a TIMESTAMP column, Data Replication sets the date part of the target TIMESTAMP value to 1 Jan 1900.
TIMESTAMP	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
VARGRAPHIC(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-

DB2 for Linux, UNIX, and Windows Source and DB2 Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a DB2 target:

DB2 Source Datatype	DB2 Target Datatype	Comments
BIGINT	BIGINT	-
BLOB(*n*)	BLOB(*n*)	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

DB2 Source Datatype	DB2 Target Datatype	Comments
CHAR FOR BIT DATA(*n*)	CHAR FOR BIT DATA(*n*)	-
CHARACTER(*n*)	CHARACTER(*n*)	-
CHARACTER FOR BIT DATA(*n*)	CHARACTER FOR BIT DATA(*n*)	-
CLOB(*n*)	CLOB(*n*)	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATE	-
DBCLOB(*n*)	DBCLOB(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

DECFLOAT	- DECFLOAT(16)
	Mapping condition:
	`(p == 8)`
	- DECFLOAT(34)
	Mapping condition:
	`(p == 16)`
		-

DECIMAL(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- DECIMAL(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	DOUBLE	-
FLOAT(*p*)	FLOAT(*p*)	-
GRAPHIC(*n*)	GRAPHIC(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
INTEGER	INTEGER	-

DB2 Source Datatype	DB2 Target Datatype	Comments
LONG VARCHAR	LONG VARCHAR	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARCHAR FOR BIT DATA	LONG VARCHAR FOR BIT DATA	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARGRAPHIC	LONG VARGRAPHIC	Data Replication does not extract LONG VARGRAPHIC data from compressed rows.
REAL	REAL	-
SMALLINT	SMALLINT	-
TIME	TIME	-
TIMESTAMP	TIMESTAMP	The Extractor truncates the fractional seconds part of source TIMESTAMP values to nine digits (nanoseconds).
VARCHAR(*n*)	VARCHAR(*n*)	-
VARCHAR FOR BIT DATA(*n*)	VARCHAR FOR BIT DATA(*n*)	-
VARGRAPHIC(*n*)	VARGRAPHIC(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

DB2 for Linux, UNIX, and Windows Source and Greenplum Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a Greenplum target:

DB2 Source Datatype	Greenplum Target Datatype	Comments
BIGINT	BIGINT	-
BLOB	BYTEA	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHARACTER(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

DB2 Source Datatype	Greenplum Target Datatype	Comments
CHARACTER FOR BIT DATA	BYTEA	-
CLOB(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATE	-
DBCLOB	TEXT	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DECFLOAT	VARCHAR(45)	-
DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*) Mapping condition: *(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)* - NUMERIC(*p*,*s'*) *s' = 0* Mapping condition: *(p <= 31) and (s == -1) and (1 <= p)* - NUMERIC(5,0) Mapping condition: *(p == -1) and (s == -1)*	-
DOUBLE	DOUBLE PRECISION	When replicating an 8-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
GRAPHIC(*n*)	CHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
INTEGER	INTEGER	-
LONG VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARCHAR FOR BIT DATA	BYTEA	Data Replication does not extract LONG VARCHAR FOR BIT DATA data from compressed rows.

DB2 Source Datatype	Greenplum Target Datatype	Comments
LONG VARGRAPHIC	TEXT	Data Replication does not extract LONG VARGRAPHIC data from compressed rows.
REAL	REAL	When replicating a 4-byte floating-point numeric value to a REAL column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLINT	SMALLINT	-
TIME	TIME	-
TIMESTAMP	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
VARCHAR FOR BIT DATA	BYTEA	-
VARGRAPHIC(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-

DB2 for Linux, UNIX, and Windows Source and Microsoft SQL Server Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a Microsoft SQL Server target:

DB2 Source Datatype	Microsoft SQL Server Target Datatype	Comments
BIGINT	BIGINT	-

BLOB(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`(n >= 1) and (n <= 8000)`
	- VARBINARY(max)
	Mapping condition:
	`(n > 8000)`
		Data Replication does not extract LOB data from compressed rows.
		Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

CHAR(*n*)	CHAR(*n*)	-
CHAR FOR BIT DATA(*n*)	BINARY(*n*)	-

DB2 Source Datatype	Microsoft SQL Server Target Datatype	Comments

CLOB(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`(n >= 1) and (n <= 8000)`
	- VARCHAR(max)
	Mapping condition:
	`(n > 8000)`
		Data Replication does not extract LOB data from compressed rows.
		Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

DATE	DATE	-

DBCLOB(*n*)	- NVARCHAR(*n*)
	Mapping condition:
	`(n >= 1) and (n <= 8000)`
	- NVARCHAR(max)
	Mapping condition:
	`(n > 8000)`
		Data Replication does not extract LOB data from compressed rows.
		Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

DECFLOAT	VARCHAR(45)	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMERIC(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE(*p*)	FLOAT(*p'*) `p' = 53`	-
FLOAT(*p*)	FLOAT(*p*)	-
GRAPHIC(*n*)	NCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
INTEGER	INT	-
LONG VARCHAR	VARCHAR(max)	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARCHAR FOR BIT DATA	VARBINARY(max)	-
LONG VARGRAPHIC	NVARCHAR(max)	Data Replication does not extract LONG VARGRAPHIC data from compressed rows.
REAL	REAL	-

DB2 Source Datatype	Microsoft SQL Server Target Datatype	Comments
SMALLINT	SMALLINT	-
TIME	TIME	-
TIMESTAMP	DATETIME2	-
VARCHAR(*n*)	- VARCHAR(MAX) Mapping condition: `(8000 < n)` - VARCHAR(*n*)	-
VARCHAR FOR BIT DATA	VARBINARY(MAX)	-

VARGRAPHIC(*n*)	- NVARCHAR(*n*)
	Mapping condition:
	`(n >= 1) and (n <= 4000)`
	- NVARCHAR(max)
	Mapping condition:
	`(n > 4000)`
		-

DB2 for Linux, UNIX, and Windows Source and MySQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a MySQL target:

DB2 Source Datatype	MySQL Target Datatype	Comments
BIGINT	BIGINT	-
BLOB(*n*)	BLOB(*n*)	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHARACTER(*n*)	CHAR(*n*)	-
CHARACTER FOR BIT DATA(*n*)	BINARY(*n*)	-

DB2 Source Datatype	MySQL Target Datatype	Comments
CLOB(*n*)	TEXT(*n*)	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATE	-
DBCLOB(*n*)	TEXT(*n*)	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DECFLOAT	VARCHAR(45)	-

DECIMAL(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- DECIMAL(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	DOUBLE PRECISION	-
FLOAT(*p*)	FLOAT(*p*)	-
GRAPHIC(*n*)	CHAR(*n*)	-
INTEGER	INT	-
LONG VARCHAR(*n*)	VARCHAR(*n*)	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARCHAR FOR BIT DATA(*n*)	BLOB(*n*)	-
LONG VARGRAPHIC(*n*)	VARCHAR(*n*)	Data Replication does not extract LONG VARGRAPHIC data from compressed rows.
REAL	REAL	-
SMALLINT	SMALLINT	-
TIME	TIME	-

DB2 Source Datatype	MySQL Target Datatype	Comments
TIMESTAMP	DATETIME	-

VARCHAR(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`256 > n`
	- TEXT(*n*)
	Mapping condition:
	`255 < n`
		-

VARCHAR FOR BIT DATA(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`(n <= 65000)`
	- BLOB(*n*)
	Mapping condition:
	`(65000 < n)`
		-

VARGRAPHIC(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`128 > n`
	- TEXT(*n*)
	Mapping condition:
	`127 < n`
		-

DB2 for Linux, UNIX, and Windows Source and Netezza Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a Netezza target:

DB2 Source Datatype	Netezza Target Datatype	Comments
BIGINT	BIGINT	-

BLOB(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n <= 16000`
	- VARCHAR(16000)
	Mapping condition:
	`n > 16000`
		-

CHAR(*n*)	NCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

DB2 Source Datatype	Netezza Target Datatype	Comments

CLOB(*n*)	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 16000`
	- NVARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
		Data Replication does not extract LOB data from compressed rows.
		Maximum extracted size is 50 MB.

DATE	DATE	-

DBCLOB(*n*)	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 16000`
	- NVARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
		Data Replication does not extract LOB data from compressed rows.
		Maximum extracted size is 50 MB.

DECFLOAT	VARCHAR(45)	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMERIC(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	DOUBLE PRECISION	When replicating an 8-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.

DB2 Source Datatype	Netezza Target Datatype	Comments

FLOAT(*p*)	- FLOAT(*p'*)
	`p' = 7`
	Mapping condition:
	`p <= 7`
	- FLOAT(*p'*)
	`p' = 15`
	Mapping condition:
	`p <= 15`
	- DOUBLE PRECISION(*p'*)
	`p' = 15`
	Mapping condition:
	`15 < p`
		When replicating a 4-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.

GRAPHIC(*n*)	NCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
INTEGER	INTEGER	-
LONG VARCHAR	NVARCHAR(16000)	-
LONG VARGRAPHIC	NVARCHAR(8192)	-
REAL	REAL	When replicating a 4-byte floating-point numeric value to a REAL column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLINT	SMALLINT	-
TIME	TIME	-
TIMESTAMP	TIMESTAMP	-

DB2 Source Datatype	Netezza Target Datatype	Comments

VARCHAR(*n*)	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 16000`
	- NVARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
		-

VARGRAPHIC(*n*)	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 16000`
	- NVARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
		-

Notes:

- Because the maximum row size for Netezza is 65535 bytes, the Data Replication does not support replication of DB2 LONG VARCHAR and LONG VARGRAPHIC datatypes to Netezza targets.

  Workaround: If the source LONG VARCHAR and LONG VARGRAPHIC values are small, and the row size in the audit log table that includes the corresponding before image and after image columns does not exceed the maximum Netezza row size, you can manually generate the target schema. Create a SQL script for generating the target schema based on the source schema from the Data Replication Console. In the script, edit the size of the VARCHAR and NVARCHAR columns that correspond to the source LONG VARCHAR and LONG VARGRAPHIC columns. Then execute the edited script to generate the target schema.
- Because Netezza does not use binary datatypes, Data Replication does not support replication of DB2 BLOB, CHAR FOR BIT DATA, and VARCHAR FOR BIT DATA datatypes to Netezza targets.

DB2 for Linux, UNIX, and Windows Source and Oracle Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and an Oracle target:

DB2 Source Datatype	Oracle Target Datatype	Comments
BIGINT	NUMBER(19,0)	-
BLOB	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. Data Replication does not extract LOB data from compressed rows. Maximum extracted size is 50 MB. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
CHAR(*n*)	CHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
CHAR FOR BIT DATA(*n*)	RAW(*n*)	-
CLOB	CLOB	Data Replication does not extract LOB data from compressed rows. Maximum extracted size is 50 MB. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
DATE	DATE	-
DBCLOB	CLOB	Data Replication does not extract LOB data from compressed rows. Maximum extracted size is 50 MB. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
DECFLOAT	VARCHAR(45)	-

DB2 Source Datatype	Oracle Target Datatype	Comments

DECIMAL(*p*,*s*)	- NUMBER(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMBER(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMBER(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE(*p*)	FLOAT(*p'*) `p' = 53`	When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
FLOAT(*p*)	FLOAT(*p*)	-
GRAPHIC(*n*)	VARCHAR2(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
INTEGER	NUMBER(10,0)	-

LONG VARCHAR(*n*)	- VARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`(Chartype != 0) and (n <= 2000)`
	- CLOB
	Mapping condition:
	`(Chartype != 0) and (2000 < n)`
		Data Replication does not extract LONG VARCHAR data from compressed rows.

LONG VARCHAR FOR BIT DATA(*n*)	- RAW(*n*)
	Mapping condition:
	`(n <= 2000)`
	- BLOB
	Mapping condition:
	`(2000 < n)`
		Data Replication does not extract LONG VARCHAR FOR BIT DATA data from compressed rows.
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

LONG VARGRAPHIC(*n*)	- VARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n <= 2000`
	- CLOB
	Mapping condition:
	`2000 < n`
		Data Replication does not extract LONG VARGRAPHIC data from compressed rows.

DB2 Source Datatype	Oracle Target Datatype	Comments
REAL(*p*)	FLOAT(*p'*) `p' = 24`	When replicating a 4-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLINT	NUMBER(5,0)	-
TIME	TIMESTAMP(0)	-
TIMESTAMP	TIMESTAMP(6)	-

VARCHAR(*n*)	- VARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`(Chartype != 0) and (n <= 2000)`
	- CLOB
	Mapping condition:
	`(Chartype != 0) and (2000 < n)`
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

VARCHAR FOR BIT DATA(*n*)	- RAW(*n*)
	Mapping condition:
	`(n <= 2000)`
	- BLOB
	Mapping condition:
	`(2000 < n)`
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

VARGRAPHIC(*n*)	- VARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n <= 2000`
	- CLOB
	Mapping condition:
	`2000 < n`
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

DB2 for Linux, UNIX, and Windows Source and PostgreSQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a PostgreSQL target:

DB2 Source Datatype	PostgreSQL Target Datatype	Comments
BIGINT	BIGINT	By default, the Data Replication Console maps BIGINT columns on the source to BIGINT columns on the target. If you run the Applier on Linux or UNIX, the Applier ends with an error when applying data to BIGINT columns because the DataDirect ODBC driver for PostgreSQL does not support this datatype.
BLOB	BYTEA	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHAR(*n*)	CHAR(*n*)	-
CHAR FOR BIT DATA	BYTEA	-
CLOB	TEXT	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATE	-
DBCLOB	TEXT	Data Replication does not extract LOB data from compressed rows. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DECFLOAT	VARCHAR(45)	-

DB2 Source Datatype	PostgreSQL Target Datatype	Comments

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMERIC(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	DOUBLE PRECISION	-
FLOAT(*p*)	FLOAT(*p*)	-
GRAPHIC(*n*)	CHAR(*n*)	-
INTEGER	INTEGER	-
LONG VARCHAR(*n*)	VARCHAR(*n*)	Data Replication does not extract LONG VARCHAR data from compressed rows.
LONG VARCHAR FOR BIT DATA	BYTEA	Data Replication does not extract LONG VARCHAR FOR BIT DATA data from compressed rows.
LONG VARGRAPHIC(*n*)	VARCHAR(*n*)	Data Replication does not extract LONG VARGRAPHIC data from compressed rows.
REAL	REAL	-
SMALLINT	SMALLINT	-
TIME	TIME	-
TIMESTAMP	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n*)	-
VARCHAR FOR BIT DATA	BYTEA	-
VARGRAPHIC(*n*)	VARCHAR(*n*)	-

DB2 for Linux, UNIX, and Windows Source and Teradata Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a Teradata target:

DB2 Source Datatype	Teradata Target Datatype	Comments
BIGINT	BIGINT	-

BLOB(*n*)	- VARBYTE(*n*)
	Mapping condition:
	`n <= 32000`
	- VARBYTE(32000)
	Mapping condition:
	`(n > 32000)`
		The target value is truncated to the size that is specified in the global.lob_truncation_size runtime parameter.

CHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
CHAR FOR BIT DATA(*n*)	BYTE(*n*)	-

CLOB(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 32000`
	- VARCHAR(32000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 32000`
		The target value is truncated to the size that is specified in the global.lob_truncation_size runtime parameter.

DATE	DATE	-

DBCLOB(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 32000`
	- VARCHAR(32000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 32000`
		Data Replication does not extract LOB data from compressed rows.
		Maximum extracted size is 50 MB.

DECFLOAT	VARCHAR(45)	-

DB2 Source Datatype	Teradata Target Datatype	Comments

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMERIC(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	FLOAT	When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
FLOAT(*p*)	FLOAT(*p*)	When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
GRAPHIC(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
INTEGER	INTEGER	-

LONG VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 32000`
	- VARCHAR(32000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 32000`
		Data Replication does not extract LONG VARCHAR data from compressed rows.

LONG VARCHAR FOR BIT DATA(*n*)	BLOB(*n*)	-

LONG VARGRAPHIC(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 32000`
	- VARCHAR(32000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 32000`
		Data Replication does not extract LONG VARGRAPHIC data from compressed rows.

DB2 Source Datatype	Teradata Target Datatype	Comments
REAL	FLOAT	When replicating a 4-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLINT	SMALLINT	-
TIME(*n*)	TIME(*n*)	-
TIMESTAMP	TIMESTAMP	-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 32000`
	- VARCHAR(32000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 32000`
		-

VARCHAR FOR BIT DATA(*n*)	- VARBYTE(*n*)
	Mapping condition:
	`(n <= 64000)`
	- BLOB(*n*)
	Mapping condition:
	`(64000 < n)`
		-

VARGRAPHIC(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

DB2 for Linux, UNIX, and Windows Source and Vertica Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a DB2 for Linux, UNIX, and Windows source and a Vertica target:

DB2 Source Datatype	Vertica Target Datatype	Comments
BIGINT	INTEGER	-

BLOB(*p*)	- VARBINARY(*p*)
	Mapping condition:
	`n <= 65000`
	- VARBINARY(65000)
	Mapping condition:
	`n > 65000`
		Data Replication does not extract LOB data from compressed rows.
		Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

CHARACTER(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-
CHARACTER FOR BIT DATA(*n*)	VARBINARY(*n*)	-

CLOB(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
		Data Replication does not extract LOB data from compressed rows.
		Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

DATE	DATE	-

DBCLOB(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
		Data Replication does not extract LOB data from compressed rows.
		Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

DECFLOAT	VARCHAR(45)	-

DB2 Source Datatype	Vertica Target Datatype	Comments

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- NUMERIC(5,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

DOUBLE	FLOAT	When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
FLOAT(*p*)	FLOAT(*p*)	-
GRAPHIC(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-
INTEGER(*n*)	INTEGER(*n'*) `n' = 8`	-

LONG VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
		Data Replication does not extract LONG VARCHAR data from compressed rows.

LONG VARCHAR FOR BIT DATA(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`(n <= 65000)`
	- VARBINARY(65000)
	Mapping condition:
	`(65000 < n)`
		-

DB2 Source Datatype	Vertica Target Datatype	Comments

LONG VARGRAPHIC(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
		Data Replication does not extract LONG VARGRAPHIC data from compressed rows.

REAL	FLOAT	When replicating a 4-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLINT(*n*)	INTEGER(*n'*) `n' = 8`	-
TIME	TIME	-
TIMESTAMP	TIMESTAMP	-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) <= 65000`
	- VARCHAR(65000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000`
		-

VARCHAR FOR BIT DATA(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`(n <= 65000)`
	- VARBINARY(65000)
	Mapping condition:
	`(65000 < n)`
		-

VARGRAPHIC(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-

Unsupported DB2 Datatypes

Data Replication does not support data extraction for the following DB2 source datatypes:

- Spatial datatypes
- User-defined datatypes
- XML
- Binary datatypes to Amazon Redshift targets

CHAPTER 3

Datatype Mappings for Microsoft SQL Server Sources

This chapter includes the following topics:

Microsoft SQL Server Source and Amazon Redshift Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and an Amazon Redshift target:

Microsoft SQL Server Source Datatype	Amazon Redshift Target Datatype	Comments
BIGINT	BIGINT	-
BIT	BOOLEAN	-
CHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-

Microsoft SQL Server Source Datatype	Amazon Redshift Target Datatype	Comments
DATE	DATE	-
DATETIME	TIMESTAMP	-
DATETIME2	TIMESTAMP	-
DATETIMEOFFSET	TIMESTAMP	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT	FLOAT	-
INT	INTEGER	-
MONEY	DECIMAL(19,4)	-
NCHAR(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-
NTEXT	VARCHAR(65000)	-

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`n != -1`
	- VARCHAR(65000)
	Mapping condition:
	`n == -1`
		-

Microsoft SQL Server Source Datatype	Amazon Redshift Target Datatype	Comments
REAL	REAL	-
SMALLDATETIME	TIMESTAMP	-
SMALLINT	SMALLINT	-
SMALLMONEY	DECIMAL(10,4)	-
SQL_VARIANT	VARCHAR(65000)	-

SYSNAME(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- VARCHAR(65000)
	Mapping condition:
	`n == -1`
		-

TEXT	VARCHAR(65000)	-
TIME	TIMESTAMP	When replicating a TIME value to a TIMESTAMP column, Data Replication sets the date part of the target TIMESTAMP value to 1 Jan 1900.
TIMESTAMP	BIGINT	-
TINYINT	NUMERIC(3,0)	-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`n != -1`
	- VARCHAR(65000)
	Mapping condition:
	`n == -1`
		-

XML	VARCHAR(65000)	-

Microsoft SQL Server Source and DB2 for Linux, UNIX, and Windows Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a DB2 for Linux, UNIX, and Windows target:

Microsoft SQL Server Source Datatype	DB2 Target Datatype	Comments
BIGINT	BIGINT	-

BINARY(*n*)	- BLOB(1G)
	Mapping condition:
	`n == -1`
	- CHAR FOR BIT DATA(*n*)
	Mapping condition:
	`(n <= 124) and (n != -1)`
	- BLOB(*n*)
		-

BIT	SMALLINT	-

CHAR(*n*)	- CHAR(*n*)
	Mapping condition:
	`(n <= 254)`
	- VARCHAR(*n*)
	Mapping condition:
	`254 < n`
		-

DATE	DATE	-
DATETIME	TIMESTAMP	-
DATETIME2	TIMESTAMP	-
DATETIMEOFFSET	TIMESTAMP	-

DECIMAL(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- VARCHAR(42)
	Mapping condition:
	`(p > 31)`
	- DECIMAL(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

Microsoft SQL Server Source Datatype	DB2 Target Datatype	Comments

FLOAT(*p*)	- FLOAT(*p'*)
	`p' = 7`
	Mapping condition:
	`p <= 7`
	- FLOAT(*p'*)
	`p' = 15`
	Mapping condition:
	`p <= 15`
	- DOUBLE
	Mapping condition:
	`p > 15 and p <= 53`
		-

GEOGRAPHY	BLOB	-
GEOMETRY	BLOB	-
HIERARCHYID	BLOB	-
IMAGE	BLOB(1G)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	INTEGER	-
MONEY	DECIMAL(19,4)	-

NCHAR(*n*)	- GRAPHIC(*n*)
	Mapping condition:
	`(n <= 127)`
	- VARGRAPHIC(*n*)
	Mapping condition:
	`127 < n`
		-

NTEXT	CLOB(1G)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Microsoft SQL Server Source Datatype	DB2 Target Datatype	Comments

NUMERIC(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 31) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 31) and (s == -1) and (1 <= p)`
	- DECIMAL(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
	- VARCHAR(42)
	Mapping condition:
	`(p > 31)`
		-

NVARCHAR(*n*)	- CLOB(1G)
	Mapping condition:
	`n == -1`
	- VARGRAPHIC(*n*)
	Mapping condition:
	`n > 0 and n <= 2023`
	- CLOB
	Mapping condition:
	`2023 < n`
		-

REAL	REAL	-
SMALLDATETIME	TIMESTAMP	-
SMALLINT	SMALLINT	-
SMALLMONEY	DECIMAL(10,4)	-
SQL_VARIANT	CLOB	-
SYSNAME(*n*)	VARGRAPHIC(*n*)	-
TEXT	CLOB(1G)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIMESTAMP	-
TIME(0)	TIME	-
TIME(1)	TIMESTAMP	-
TIME(2)	TIMESTAMP	-
TIME(3)	TIMESTAMP	-

Microsoft SQL Server Source Datatype	DB2 Target Datatype	Comments
TIME(4)	TIMESTAMP	-
TIME(5)	TIMESTAMP	-
TIME(6)	TIMESTAMP	-
TIMESTAMP	BIGINT	-
TINYINT	DECIMAL(3,0)	-
UNIQUEIDENTIFIER(*n*)	CHAR FOR BIT DATA(*n'*) *n' = 16*	-
VARBINARY(*n*)	- BLOB(1G) Mapping condition: *n == -1* - BLOB(*n*)	-
VARCHAR(*n*)	- VARCHAR(*n*) Mapping condition: *(n < 4046) and (n != -1)* - CLOB Mapping condition: *4046 <= n* - CLOB Mapping condition: *n == -1*	-
XML	CLOB	-

Microsoft SQL Server Source and Greenplum Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a Greenplum target:

Microsoft SQL Server Source Datatype	Greenplum Target Datatype	Comments
BIGINT	BIGINT	-
BINARY	BYTEA	-
BIT	BIT	-
CHAR(*n*)	CHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-

Microsoft SQL Server Source Datatype	Greenplum Target Datatype	Comments
DATE	DATE	-
DATETIME	TIMESTAMP	-
DATETIME2	TIMESTAMP	-
DATETIMEOFFSET	TIMESTAMP(6) WITH TIME ZONE	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT	FLOAT	When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
GEOGRAPHY	BYTEA	-
GEOMETRY	BYTEA	-
HIERARCHYID	BYTEA	-
IMAGE	BYTEA	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	INTEGER	-
MONEY	DECIMAL(19,4)	-
NCHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NTEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Microsoft SQL Server Source Datatype	Greenplum Target Datatype	Comments

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n != -1`
	- TEXT
	Mapping condition:
	`n == -1`
		-

REAL	REAL	When replicating a 4-byte floating-point numeric value to a REAL column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLDATETIME	TIMESTAMP(0)	-
SMALLINT	SMALLINT	-
SMALLMONEY	DECIMAL(10,4)	-
SQL_VARIANT	TEXT	-

SYSNAME(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- TEXT
	Mapping condition:
	`n == -1`
		-

TEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIME	-
TIMESTAMP	BIGINT	-
TINYINT	NUMERIC(3,0)	-

Microsoft SQL Server Source Datatype	Greenplum Target Datatype	Comments
UNIQUEIDENTIFIER	- BYTEA - BYTEA	-
VARBINARY	BYTEA	-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n != -1`
	- TEXT
	Mapping condition:
	`n == -1`
		-

XML	TEXT	-

Microsoft SQL Server Source and Microsoft SQL Server Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a Microsoft SQL Server target:

Microsoft SQL Server Source Datatype	Microsoft SQL Server Target Datatype	Comments
BIGINT	BIGINT	-
BINARY(*n*)	BINARY(*n*)	-
BIT	BIT	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-
DATETIME	DATETIME	-
DATETIME2	DATETIME2	-
DATETIMEOFFSET	DATETIMEOFFSET	-

Microsoft SQL Server Source Datatype	Microsoft SQL Server Target Datatype	Comments

DECIMAL(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- DECIMAL(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT	FLOAT	-
GEOGRAPHY	GEOGRAPHY	-
GEOMETRY	GEOMETRY	-
HIERARCHYID	HIERARCHYID	-
IMAGE	IMAGE	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	INT	-
MONEY	MONEY	-
NCHAR(*n*)	NCHAR(*n*)	-
NTEXT	NTEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

Microsoft SQL Server Source Datatype	Microsoft SQL Server Target Datatype	Comments

NVARCHAR(*n*)	- NVARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- NVARCHAR(MAX)
	Mapping condition:
	`n == -1`
		-

REAL	REAL	-
SMALLDATETIME	SMALLDATETIME	-
SMALLINT	SMALLINT	-
SMALLMONEY	SMALLMONEY	-
SQL_VARIANT	SQL_VARIANT	-
SYSNAME(*n*)	NVARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
TEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIME	-
TIME(0)	TIME(0)	-
TIME(1)	TIME(1)	-
TIME(2)	TIME(2)	-
TIME(3)	TIME(3)	-
TIME(4)	TIME(4)	-
TIME(5)	TIME(5)	-
TIME(6)	TIME(6)	-
TIMESTAMP	BIGINT	-
TINYINT	TINYINT	-
UNIQUEIDENTIFIER	UNIQUEIDENTIFIER	-

Microsoft SQL Server Source Datatype	Microsoft SQL Server Target Datatype	Comments

VARBINARY(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`n != -1`
	- VARBINARY(MAX)
	Mapping condition:
	`n == -1`
		-

VARCHAR(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- VARCHAR(MAX)
	Mapping condition:
	`n == -1`
		-

XML	XML	-

Microsoft SQL Server Source and MySQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a MySQL target:

Microsoft SQL Server Source Datatype	MySQL Target Datatype	Comments
BIGINT	BIGINT	-

BINARY(*n*)	- BINARY(*n*)
	Mapping condition:
	`(n <= 255) and (n != -1)`
	- BLOB
	Mapping condition:
	`255 < n`
	- BLOB
	Mapping condition:
	`n == -1`
		-

BIT	TINYINT	-

CHAR(*n*)	- CHAR(*n*)
	Mapping condition:
	`n <= 255`
	- VARCHAR(*n*)
	Mapping condition:
	`255 < n`
		-

DATE	DATE	-

Microsoft SQL Server Source Datatype	MySQL Target Datatype	Comments
DATETIME	DATETIME	-
DATETIME2	DATETIME	-
DATETIMEOFFSET	DATETIME	-

DECIMAL(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- DECIMAL(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT(*p*)	- FLOAT(*p'*)
	`p' = 7`
	Mapping condition:
	`p <= 7`
	- FLOAT(*p'*)
	`p' = 15`
	Mapping condition:
	`p <= 15`
	- FLOAT
	Mapping condition:
	`p > 15`
		-

GEOGRAPHY	BLOB	-
GEOMETRY	BLOB	-
HIERARCHYID	BLOB	-
IMAGE	BLOB	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	INT	-
MONEY	DECIMAL(19,4)	-

NCHAR(*n*)	- CHAR(*n*)
	Mapping condition:
	`n <= 255`
	- VARCHAR(*n*)
	Mapping condition:
	`255 < n`
		-

Microsoft SQL Server Source Datatype	MySQL Target Datatype	Comments
NTEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMERIC(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- DECIMAL(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- DECIMAL(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- LONGTEXT
	Mapping condition:
	`n == -1`
		-

REAL	REAL	-
SMALLDATETIME	DATETIME	-
SMALLINT	SMALLINT	-
SMALLMONEY	DECIMAL(10,4)	-
SQL_VARIANT	TEXT	-

SYSNAME(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- TEXT
	Mapping condition:
	`n == -1`
		-

TEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIME	-
TIME(0)	TIME	-
TIME(1)	TIME	-

Microsoft SQL Server Source Datatype	MySQL Target Datatype	Comments
TIME(2)	TIME	-
TIME(3)	TIME	-
TIME(4)	TIME	-
TIME(5)	TIME	-
TIME(6)	TIME	-
TIMESTAMP	BIGINT	-
TINYINT	SMALLINT	-
UNIQUEIDENTIFIER	BINARY(16)	-

VARBINARY(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`n != -1`
	- LONGBLOB
	Mapping condition:
	`n == -1`
		-

VARCHAR(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- LONGTEXT
	Mapping condition:
	`n == -1`
		-

XML	TEXT	-

Microsoft SQL Server Source and Netezza Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a Netezza target:

Microsoft SQL Server Source Datatype	Netezza Target Datatype	Comments
BIGINT	BIGINT	-
BINARY(*n*)	VARCHAR(*n*)	-
BIT	BYTEINT	-

Microsoft SQL Server Source Datatype	Netezza Target Datatype	Comments

CHAR(*n*)	- NCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 16000`
	- NCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
		-

DATE	DATE	-
DATETIME	TIMESTAMP	-
DATETIME2	TIMESTAMP	-
DATETIMEOFFSET	TIMESTAMP	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT(*p*)	- FLOAT(*p'*)
	`p' = 7`
	Mapping condition:
	`p <= 7`
	- FLOAT(*p'*)
	`p' = 15`
	Mapping condition:
	`p <= 15`
	- FLOAT
	Mapping condition:
	`p > 15`
		When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.

GEOGRAPHY	VARCHAR(16000)	-
GEOMETRY	VARCHAR(16000)	-
HIERARCHYID	VARCHAR(16000)	-

Microsoft SQL Server Source Datatype	Netezza Target Datatype	Comments
IMAGE	VARCHAR(16000)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	INTEGER	-
MONEY	NUMERIC(19,4)	-
NCHAR(*n*)	NCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NTEXT	NVARCHAR(16000)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n != -1`
	- NVARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

REAL	REAL	When replicating a 4-byte floating-point numeric value to a REAL column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
SMALLDATETIME	TIMESTAMP	-
SMALLINT	SMALLINT	-
SMALLMONEY	NUMERIC(10,4)	-
SQL_VARIANT	NVARCHAR(8000)	-

Microsoft SQL Server Source Datatype	Netezza Target Datatype	Comments

SYSNAME(*n*)	- NVARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- NVARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

TEXT	NVARCHAR(16000)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIME	-
TIME(0)	TIME	-
TIME(1)	TIME	-
TIME(2)	TIME	-
TIME(3)	TIME	-
TIME(4)	TIME	-
TIME(5)	TIME	-
TIME(6)	TIME	-
TIMESTAMP	BIGINT	-
TINYINT	SMALLINT	-
UNIQUEIDENTIFIER	VARCHAR(40)	-

VARBINARY(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- VARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

Microsoft SQL Server Source Datatype	Netezza Target Datatype	Comments

VARCHAR(*n*)	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n != -1 and ((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) <= 16000`
	- NVARCHAR(16000)
	Mapping condition:
	`n != -1 and ((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
	- NVARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

XML	NVARCHAR(8000)	-

Because Netezza does not use binary dataypes, Data Replication does not support replication of Microsoft SQL Server binary, varbinary, and varbinary(max) datatypes to Netezza targets.

Microsoft SQL Server Source and Oracle Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and an Oracle target:

Microsoft SQL Server Source Datatype	Oracle Target Datatype	Comments
BIGINT	NUMBER(19,0)	-
BINARY	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
BIT	NUMBER(1)	-

Microsoft SQL Server Source Datatype	Oracle Target Datatype	Comments
CHAR(*n*)	- CHAR(*n'*) *n'* = (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* Mapping condition: (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* <= 2000 - VARCHAR2(*n'*) *n'* = (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* Mapping condition: 2000 < (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* and (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* <= 4000 - CLOB Mapping condition: 4000 < (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize*	The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. To support national character data, the Data Replication Console defines the Oracle VARCHAR2 column size by using character length semantics.
DATE	DATE	-
DATETIME	TIMESTAMP(3)	-
DATETIME2(*p*)	TIMESTAMP(*p*)	-
DATETIMEOFFSET	- TIMESTAMP(0) WITH TIME ZONE Mapping condition: *s* == 0 - TIMESTAMP(1) WITH TIME ZONE Mapping condition: *s* == 1 - TIMESTAMP(2) WITH TIME ZONE Mapping condition: *s* == 2 - TIMESTAMP(3) WITH TIME ZONE Mapping condition: *s* == 3 - TIMESTAMP(4) WITH TIME ZONE Mapping condition: *s* == 4 - TIMESTAMP(5) WITH TIME ZONE Mapping condition: *s* == 5 - TIMESTAMP(6) WITH TIME ZONE Mapping condition: *s* == 6 - TIMESTAMP(7) WITH TIME ZONE Mapping condition: *s* == 7	-

Microsoft SQL Server Source Datatype	Oracle Target Datatype	Comments

DECIMAL(*p*,*s*)	- NUMBER(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMBER(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMBER(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT	- BINARY_DOUBLE
	Mapping condition:
	`p <= 7`
	- BINARY_DOUBLE
	Mapping condition:
	`p <= 15`
	- BINARY_DOUBLE
	Mapping condition:
	`p > 15`
		-

GEOGRAPHY	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
GEOMETRY	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
HIERARCHYID	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

Microsoft SQL Server Source Datatype	Oracle Target Datatype	Comments
IMAGE	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	NUMBER(10,0)	-
MONEY	NUMBER(19,4)	-

NCHAR(*n*)	- NCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`(n + SrcMinCharsetSize - 1) / SrcMinCharsetSize <= 2000`
	- NVARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`2000 < (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize and (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize <= 4000`
	- NCLOB
	Mapping condition:
	`4000 < (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

NTEXT(*n*)	- NCLOB - CLOB(*n*)	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Microsoft SQL Server Source Datatype	Oracle Target Datatype	Comments

NUMERIC(*p*,*s*)	- NUMBER(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMBER(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMBER(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- NVARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`0 < n and (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize <= 4000`
	- NCLOB
	Mapping condition:
	`n == -1 || 4000 < (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

REAL	BINARY_FLOAT	-
SMALLDATETIME	TIMESTAMP(0)	-
SMALLINT	NUMBER(5,0)	-
SMALLMONEY	NUMBER(10,4)	-
SQL_VARIANT	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
SYSNAME(*n*)	NVARCHAR2(*n*)	-
TEXT	CLOB	The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME(*p*)	TIMESTAMP(*p*)	-
TIMESTAMP	NUMBER(25,0)	-

Microsoft SQL Server Source Datatype	Oracle Target Datatype	Comments
TINYINT	NUMBER(3,0)	-
UNIQUEIDENTIFIER	RAW(16)	-
VARBINARY	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

VARCHAR(*n*)	- VARCHAR2(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`(n + SrcMinCharsetSize - 1) / SrcMinCharsetSize <= 4000 and n != -1`
	- CLOB
	Mapping condition:
	`n == -1 || 4000 < (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.
		To support national character data, the Data Replication Console defines the Oracle VARCHAR2 column size by using character length semantics.

XML	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode.

Microsoft SQL Server Source and PostgreSQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a PostgreSQL target:

Microsoft SQL Server Source Datatype	PostgreSQL Target Datatype	Comments
BIGINT	BIGINT	By default, the Data Replication Console maps BIGINT source columns to BIGINT columns on the target. If you run the Applier on Linux or UNIX, map BIGINT source columns to NUMERIC target columns. The Applier ends with an error when applying data to BIGINT columns because the DataDirect ODBC driver for PostgreSQL does not support this datatype.
BINARY	BYTEA	-
BIT	SMALLINT	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-
DATETIME	TIMESTAMP(3)	-
DATETIME2	TIMESTAMP(6)	-
DATETIMEOFFSET	TIMESTAMP(6) WITH TIME ZONE	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT(*p*)	- FLOAT(*p'*)
	`p' = 7`
	Mapping condition:
	`p <= 7`
	- DOUBLE PRECISION
	Mapping condition:
	`p > 7`
		-

GEOGRAPHY	BYTEA	-
GEOMETRY	BYTEA	-
HIERARCHYID	BYTEA	-

Microsoft SQL Server Source Datatype	PostgreSQL Target Datatype	Comments
IMAGE	BYTEA	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
INT	INTEGER	-
MONEY	NUMERIC(19,4)	-
NCHAR(*n*)	CHAR(*n*)	-
NTEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- TEXT
	Mapping condition:
	`n == -1`
	- VARCHAR(*n*)
	Mapping condition:
	`n <= 255`
	- TEXT
	Mapping condition:
	`n > 255`
		-

REAL	REAL	-
SMALLDATETIME	TIMESTAMP(0)	-
SMALLINT	SMALLINT	-
SMALLMONEY	NUMERIC(10,4)	-
SQL_VARIANT	TEXT	-

Microsoft SQL Server Source Datatype	PostgreSQL Target Datatype	Comments

SYSNAME(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- TEXT
	Mapping condition:
	`n == -1`
		-

TEXT	TEXT	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIME(6)	-
TIME(0)	TIME(0)	-
TIME(1)	TIME(1)	-
TIME(2)	TIME(2)	-
TIME(3)	TIME(3)	-
TIME(4)	TIME(4)	-
TIME(5)	TIME(5)	-
TIME(6)	TIME(6)	-
TIMESTAMP	BIGINT	-
TINYINT	SMALLINT	-
UNIQUEIDENTIFIER	BYTEA	-
VARBINARY	BYTEA	-

VARCHAR(*n*)	- TEXT
	Mapping condition:
	`n == -1`
	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
		-

XML	TEXT	-

Microsoft SQL Server Source and Teradata Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a Teradata target:

Microsoft SQL Server Source Datatype	Teradata Target Datatype	Comments
BIGINT	BIGINT	-
BINARY(*n*)	BYTE(*n*)	-
BIT	SMALLINT	-
CHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
DATE	DATE	-
DATETIME	TIMESTAMP(3)	-
DATETIME2	TIMESTAMP(6)	-
DATETIMEOFFSET	TIMESTAMP(6) WITH TIME ZONE	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT(*p*)	- FLOAT(*p'*)
	`p' = 7`
	Mapping condition:
	`p <= 7`
	- FLOAT(*p'*)
	`p' = 15`
	Mapping condition:
	`p <= 15`
	- FLOAT
	Mapping condition:
	`p > 15`
		-

GEOGRAPHY	VARBYTE(16000)	-
GEOMETRY	VARBYTE(16000)	-
HIERARCHYID	VARBYTE(16000)	-

Microsoft SQL Server Source Datatype	Teradata Target Datatype	Comments
IMAGE	VARBYTE(16000)	The target value is truncated to the size that is specified in the global.lob_truncation_size runtime parameter.
INT	INTEGER	-
MONEY	NUMERIC(19,4)	-
NCHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NTEXT	VARCHAR(16000)	The target value is truncated to the size that is specified in the global.lob_truncation_size runtime parameter.

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n != -1`
	- VARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

REAL(*p*)	FLOAT(*p'*) `p' = 7`	When replicating a 4-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision parameter to change the default precision value.
SMALLDATETIME	TIMESTAMP(0)	-
SMALLINT	SMALLINT	-
SMALLMONEY	NUMERIC(10,4)	-
SQL_VARIANT	VARCHAR(16000)	-

Microsoft SQL Server Source Datatype	Teradata Target Datatype	Comments

SYSNAME(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- VARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

TEXT	VARCHAR(16000)	-

TIME(*n*)	- TIME(*n*)
	Mapping condition:
	`s <= 6`
	- TIME(6)
	Mapping condition:
	`s > 6`
		-

TIMESTAMP	BIGINT	-
TINYINT	SMALLINT	-
UNIQUEIDENTIFIER	BYTE(16)	-

VARBINARY(*n*)	- VARBYTE(*n*)
	Mapping condition:
	`n != -1`
	- VARBYTE(16000)
	Mapping condition:
	`n == -1`
		-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`n != -1`
	- VARCHAR(16000)
	Mapping condition:
	`n == -1`
		-

XML	VARCHAR(16000)	The target value is truncated to the size that is specified in the global.lob_truncation_size runtime parameter.

Microsoft SQL Server Source and Vertica Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a Microsoft SQL Server source and a Vertica target:

Microsoft SQL Server Source Datatype	Vertica Target Datatype	Comments
BIGINT	INTEGER	-
BINARY(*n*)	BINARY(*n*)	-
BIT	INTEGER	-
CHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-
DATE	DATE	-
DATETIME	TIMESTAMP(6)	-
DATETIME2	TIMESTAMP(6)	-
DATETIMEOFFSET	TIMESTAMP(6) WITH TIME ZONE	-

DECIMAL(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

FLOAT	FLOAT	When replicating an 8-byte floating-point numeric value to a FLOAT column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision parameter to change the default precision value.
GEOGRAPHY	VARBINARY(65000)	-
GEOMETRY	VARBINARY(65000)	-
HIERARCHYID	VARBINARY(65000)	-
IMAGE	VARBINARY(65000)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Microsoft SQL Server Source Datatype	Vertica Target Datatype	Comments
INT	INTEGER	-
MONEY	NUMERIC(19,4)	-
NCHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`	-
NTEXT	VARCHAR(65000)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMERIC(*p*,*s*)	- NUMERIC(*p*,*s*)
	Mapping condition:
	`(p <= 38) and (s <= p) and (0 <= s) and (1 <= p)`
	- NUMERIC(*p*,*s'*)
	`s' = 0`
	Mapping condition:
	`(p <= 38) and (s == -1) and (1 <= p)`
	- NUMERIC(18,0)
	Mapping condition:
	`(p == -1) and (s == -1)`
		-

NVARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`n != -1`
	- VARCHAR(65000)
	Mapping condition:
	`n == -1`
		-

REAL	FLOAT	-
SMALLDATETIME	TIMESTAMP(0)	-
SMALLINT	INTEGER	-
SMALLMONEY	NUMERIC(10,4)	-
SQL_VARIANT	VARCHAR(8000)	-

SYSNAME(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n != -1`
	- VARCHAR(128)
	Mapping condition:
	`n == -1`
		-

Microsoft SQL Server Source Datatype	Vertica Target Datatype	Comments
TEXT	VARCHAR(65000)	Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
TIME	TIME	-
TIMESTAMP	BIGINT	-
TINYINT	INTEGER	-
UNIQUEIDENTIFIER	BINARY(16)	-

VARBINARY(*n*)	- VARBINARY(*n*)
	Mapping condition:
	`n != -1`
	- VARBINARY(65000)
	Mapping condition:
	`n == -1`
		-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize`
	Mapping condition:
	`n != -1`
	- VARCHAR(65000)
	Mapping condition:
	`n == -1`
		-

XML	VARCHAR(65000)	-

Unsupported Microsoft SQL Server Datatypes

Data Replication does not support data extraction for the following Microsoft SQL Server source datatypes:

- filestream
- rowversion
- sql_variant
- timestamp
- user-defined datatypes
- binary and spatial datatypes to Amazon Redshift targets

CHAPTER 4

Datatype Mappings for MySQL Sources

This chapter includes the following topics:

MySQL Source and Amazon Redshift Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and an Amazon Redshift target:

MySQL Source Datatype	Amazon Redshift Target Datatype	Comments
BIGINT	DECIMAL(20,0)	-
BIT	DECIMAL(20,0)	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-
DATETIME	TIMESTAMP	-

MySQL Source Datatype	Amazon Redshift Target Datatype	Comments

DECIMAL(*p*,*s*)	- DECIMAL(*p'*,*s*)
	`p' = 38`
	Mapping condition:
	`p > 38`
	- NUMERIC(*p*,*s*)
		-

DOUBLE	DOUBLE PRECISION	-
ENUM	VARCHAR(255)	-
FLOAT	REAL	-
INT	BIGINT	-
LONGTEXT	VARCHAR(65000)	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	VARCHAR(65000)	-
SET	VARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	VARCHAR(65000)	-
TIME	TIMESTAMP	-
TIMESTAMP	TIMESTAMP	-
TINYINT	SMALLINT	-
TINYTEXT	VARCHAR(256)	-
VARCHAR(*n*)	VARCHAR(*n*)	-
YEAR	INTEGER	-

MySQL Source and DB2 Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a DB2 for Linux, UNIX, and Windows target:

MySQL Source Datatype	DB2 Target Datatype	Comments
BIGINT	DECIMAL(20,0)	-
BINARY(*n*)	BLOB(*n*)	-

MySQL Source Datatype	DB2 Target Datatype	Comments
BIT	DECIMAL(20,0)	-
BLOB	BLOB	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

DATETIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	- DECIMAL(*p'*,*s*)
	`p' = 31`
	Mapping condition:
	`p > 31`
	- DECIMAL(*p*,*s*)
		-

DOUBLE	REAL	-
ENUM	VARCHAR(255)	-
FLOAT	REAL	-
INT	BIGINT	-
LONGBLOB	BLOB	-
LONGTEXT	CLOB(1G)	-
MEDIUMBLOB	BLOB	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	CLOB(1G)	-
SET	VARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	CLOB(1G)	-
TIME	TIME	-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

MySQL Source Datatype	DB2 Target Datatype	Comments
TINYBLOB	BLOB(255)	-
TINYINT	SMALLINT	-
TINYTEXT	VARCHAR(255)	-
VARBINARY(*n*)	BLOB(*n*)	-
VARCHAR(*n*)	- CLOB Mapping condition: `n > 32672` - VARCHAR(*n*)	-
YEAR	INTEGER	-

MySQL Source and Greenplum Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a Greenplum target:

MySQL Source Datatype	Greenplum Target Datatype	Comments
BIGINT	NUMERIC(20,0)	-
BINARY	BYTEA	-
BIT	NUMERIC(20,0)	-
BLOB	BYTEA	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

DATETIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	FLOAT	-
ENUM	VARCHAR(255)	-
FLOAT	REAL	-

MySQL Source Datatype	Greenplum Target Datatype	Comments
INT	BIGINT	-
LONGBLOB	BYTEA	-
LONGTEXT	TEXT	-
MEDIUMBLOB	BYTEA	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	TEXT	-
SET	VARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	TEXT	-

TIME(*n*)	- TIME(0)
	Mapping condition:
	`n == -1`
	- TIME(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	BYTEA	-
TINYINT	SMALLINT	-
TINYTEXT	VARCHAR(255)	-
VARBINARY	BYTEA	-
VARCHAR(*n*)	VARCHAR(*n*)	-
YEAR	INTEGER	-

MySQL Source and Microsoft SQL Server Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a Microsoft SQL Server target:

MySQL Source Datatype	Microsoft SQL Server Target Datatype	Comments
BIGINT	NUMERIC(20,0)	-
BINARY(*n*)	BINARY(*n*)	-
BIT	NUMERIC(20,0)	-
BLOB	VARBINARY(MAX)	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

DATETIME(*n*)	- DATETIME2(0)
	Mapping condition:
	`n == -1`
	- DATETIME2(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	- NUMERIC(*p'*,*s*)
	`p' = 38`
	Mapping condition:
	`p > 38`
	- NUMERIC(*p*,*s*)
		-

DOUBLE	FLOAT	-
ENUM	VARCHAR(255)	-
FLOAT	REAL	-
INT	BIGINT	-
LONGBLOB	VARBINARY(MAX)	-
LONGTEXT	VARCHAR(MAX)	-
MEDIUMBLOB	VARBINARY(MAX)	-
MEDIUMINT	INT	-
MEDIUMTEXT	VARCHAR(MAX)	-
SET	VARCHAR(4000)	-
SMALLINT	INT	-
TEXT	VARCHAR(MAX)	-

MySQL Source Datatype	Microsoft SQL Server Target Datatype	Comments

TIME(*n*)	- TIME(0)
	Mapping condition:
	`n == -1`
	- TIME(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- DATETIME2(0)
	Mapping condition:
	`n == -1`
	- DATETIME2(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	VARBINARY(255)	-
TINYINT	SMALLINT	-
TINYTEXT	VARCHAR(255)	-
VARBINARY(*n*)	VARBINARY(*n*)	-
VARCHAR(*n*)	- VARCHAR(MAX) Mapping condition: `n > 8000` - VARCHAR(*n*)	-
YEAR	INT	-

MySQL Source and MySQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a MySQL target:

MySQL Source Datatype	MySQL Target Datatype	Comments
BIGINT	BIGINT	-
BINARY(*n*)	BINARY(*n*)	-
BIT	BIT	-
BLOB	BLOB	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

MySQL Source Datatype	MySQL Target Datatype	Comments

DATETIME(*n*)	- DATETIME(0)
	Mapping condition:
	`n == -1`
	- DATETIME(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	DECIMAL(*p*,*s*)	-
DOUBLE	DOUBLE	-
ENUM	VARCHAR(255)	-
FLOAT	FLOAT	-
INT	INT	-
LONGBLOB	LONGBLOB	-
LONGTEXT	LONGTEXT	-
MEDIUMBLOB	MEDIUMBLOB	-
MEDIUMINT	MEDIUMINT	-
MEDIUMTEXT	MEDIUMTEXT	-
SET	VARCHAR(4000)	-
SMALLINT	SMALLINT	-
TEXT	TEXT	-

TIME(*n*)	- TIME(0)
	Mapping condition:
	`n == -1`
	- TIME(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	TINYBLOB	-
TINYINT	TINYINT	-
TINYTEXT	TINYTEXT	-

MySQL Source Datatype	MySQL Target Datatype	Comments
VARBINARY(*n*)	VARBINARY(*n*)	-
VARCHAR(*n*)	VARCHAR(*n*)	-
YEAR	YEAR	-

MySQL Source and Netezza Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a Netezza target:

MySQL Source Datatype	Netezza Target Datatype	Comments
BIGINT	NUMERIC(20,0)	-
BIT	NUMERIC(20,0)	-
CHAR(*n*)	NCHAR(*n*)	-
DATE	DATE	-
DATETIME	TIMESTAMP	-

DECIMAL(*p*,*s*)	- NUMERIC(*p'*,*s*)
	`p' = 38`
	Mapping condition:
	`p > 38`
	- NUMERIC(*p*,*s*)
		-

DOUBLE	FLOAT	-
ENUM	NVARCHAR(255)	-
FLOAT	FLOAT	-
INT	BIGINT	-
LONGTEXT	NVARCHAR(16000)	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	NVARCHAR(16000)	-
SET	NVARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	NVARCHAR(16000)	-

MySQL Source Datatype	Netezza Target Datatype	Comments
TIME	TIME	-
TIMESTAMP	TIMESTAMP	-
TINYINT	SMALLINT	-
TINYTEXT	NVARCHAR(255)	-

VARCHAR(*n*)	- NVARCHAR(*n'*)
	`n' = 64000`
	Mapping condition:
	`n > 64000`
	- NVARCHAR(*n*)
		-

YEAR	INTEGER	-

MySQL Source and Oracle Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and an Oracle target:

MySQL Source Datatype	Oracle Target Datatype	Comments
BIGINT	NUMBER(20,0)	-
BINARY(*n*)	RAW(*n*)	-
BIT	NUMBER(20,0)	-
BLOB	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

DATETIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

MySQL Source Datatype	Oracle Target Datatype	Comments

DECIMAL(*p*,*s*)	- DECIMAL(*p'*,*s*)
	`p' = 38`
	Mapping condition:
	`p > 38`
	- NUMBER(*p*,*s*)
		-

DOUBLE	BINARY_DOUBLE	-
ENUM	VARCHAR(255)	-
FLOAT	BINARY_FLOAT	-
INT	NUMBER(10,0)	-
LONGBLOB	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
LONGTEXT	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
MEDIUMBLOB	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
MEDIUMINT	NUMBER(8,0)	-
MEDIUMTEXT	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
SET	VARCHAR(4000)	-
SMALLINT	NUMBER(5,0)	-

MySQL Source Datatype	Oracle Target Datatype	Comments
TEXT	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.

TIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
TINYINT	NUMBER(3,0)	-
TINYTEXT	VARCHAR2(255)	-
VARBINARY	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
VARCHAR(*n*)	- CLOB Mapping condition: `n > 4000` - VARCHAR2(*n*)	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns.
YEAR	INTEGER	-

MySQL Source and PostgreSQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a PostgreSQL target:

MySQL Source Datatype	PostgreSQL Target Datatype	Comments
BIGINT	NUMERIC(20,0)	-
BINARY	BYTEA	-
BIT	NUMERIC(20,0)	-
BLOB	BYTEA	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

DATETIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	FLOAT	-
ENUM	VARCHAR(255)	-
FLOAT	REAL	-
INT	BIGINT	-
LONGBLOB	BYTEA	-
LONGTEXT	TEXT	-
MEDIUMBLOB	BYTEA	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	TEXT	-
SET	VARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	TEXT	-

MySQL Source Datatype	PostgreSQL Target Datatype	Comments

TIME(*n*)	- TIME(0)
	Mapping condition:
	`n == -1`
	- TIME(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	BYTEA	-
TINYINT	SMALLINT	-
TINYTEXT	VARCHAR(255)	-
VARBINARY	BYTEA	-
VARCHAR(*n*)	VARCHAR(*n*)	-
YEAR	INTEGER	-

MySQL Source and Teradata Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a Teradata target:

MySQL Source Datatype	Teradata Target Datatype	Comments
BIGINT	NUMBER(20,0)	-
BINARY(*n*)	VARBYTE(*n*)	-
BIT	NUMBER(20,0)	-
BLOB	VARBYTE(16000)	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

MySQL Source Datatype	Teradata Target Datatype	Comments

DATETIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	- NUMBER(*p'*,*s*)
	`p' = 38`
	Mapping condition:
	`p > 38`
	- NUMBER(*p*,*s*)
		-

DOUBLE	FLOAT	-
ENUM	VARCHAR(255)	-
FLOAT	FLOAT	-
INT	BIGINT	-
LONGBLOB	VARBYTE(16000)	-
LONGTEXT	VARCHAR(16000)	-
MEDIUMBLOB	VARBYTE(16000)	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	VARCHAR(16000)	-
SET	VARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	VARCHAR(16000)	-

TIME(*n*)	- TIME(0)
	Mapping condition:
	`n == -1`
	- TIME(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	VARBYTE(255)	-

MySQL Source Datatype	Teradata Target Datatype	Comments
TINYINT	SMALLINT	-
TINYTEXT	VARCHAR(255)	-
VARBINARY(*n*)	VARBYTE(*n*)	-

VARCHAR(*n*)	- VARCHAR(*n'*)
	`n' = 16000`
	Mapping condition:
	`n > 16000`
	- VARCHAR(*n*)
		-

YEAR	INTEGER	-

MySQL Source and Vertica Target

The following table identifies the recommended datatype mappings for Data Replication configurations with a MySQL source and a Vertica target:

MySQL Source Datatype	Vertica Target Datatype	Comments
BIGINT	NUMERIC(20,0)	-
BINARY(*n*)	BINARY(*n*)	-
BIT	NUMERIC(20,0)	-
BLOB	VARBINARY(65000)	-
CHAR(*n*)	CHAR(*n*)	-
DATE	DATE	-

DATETIME(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	REAL	-
ENUM	VARCHAR(255)	-
FLOAT	FLOAT	-
INT	BIGINT	-

MySQL Source Datatype	Vertica Target Datatype	Comments
LONGBLOB	VARBINARY(65000)	-
LONGTEXT	VARCHAR(65000)	-
MEDIUMBLOB	VARBINARY(65000)	-
MEDIUMINT	INTEGER	-
MEDIUMTEXT	VARCHAR(65000)	-
SET	VARCHAR(4000)	-
SMALLINT	INTEGER	-
TEXT	VARCHAR(65000)	-

TIME(*n*)	- TIME(0)
	Mapping condition:
	`n == -1`
	- TIME(*n*)
	Mapping condition:
	`n != -1`
		-

TIMESTAMP(*n*)	- TIMESTAMP(0)
	Mapping condition:
	`n == -1`
	- TIMESTAMP(*n*)
	Mapping condition:
	`n != -1`
		-

TINYBLOB	VARBINARY(255)	-
TINYINT	INTEGER	-
TINYTEXT	VARCHAR(255)	-
VARBINARY(*n*)	VARBINARY(*n*)	-

VARCHAR(*n*)	- VARCHAR(*n*)
	- VARCHAR(*n'*)
	`n' = 65000`
	Mapping condition:
	`n > 65000`
		-

YEAR	INTEGER	-

CHAPTER 5

Datatype Mappings for Oracle Sources

This chapter includes the following topics:

Oracle Source and Amazon Redshift Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and an Amazon Redshift target:

Oracle Source Datatype	Amazon Redshift Target Datatype	Comments
BINARY_DOUBLE	NUMERIC(37,15)	-
BINARY_FLOAT	NUMERIC(37,15)	-
CHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
CLOB	VARCHAR(65000)	-

Oracle Source Datatype	Amazon Redshift Target Datatype	Comments
DATE	TIMESTAMP	-
FLOAT	NUMERIC(37,15)	-
LONG	VARCHAR(65000)	-
NCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
NCLOB	VARCHAR(65000)	-
NUMBER(*p*,*s*)	- NUMERIC Mapping condition: *p == -1 and s == -1* - NUMERIC(38, 0) Mapping condition: *p == -1 and s == 0* - SMALLINT Mapping condition: *p <= 4 and s == 0* - INTEGER Mapping condition: *p <= 9 and s == 0* - BIGINT Mapping condition: *p <= 18 and s == 0* - NUMERIC(*p*,*s*) Mapping condition: *p > 18 and s == 0* - NUMERIC(*p*,*s*) Mapping condition: *p != 0 and s != 0*	-
NVARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
NVARCHAR2(*n*)	- VARCHAR(65000) Mapping condition: *((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000* - VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
TIMESTAMP(0)	TIMESTAMP	-
TIMESTAMP(1)	TIMESTAMP	-
TIMESTAMP(2)	TIMESTAMP	-

Oracle Source Datatype	Amazon Redshift Target Datatype	Comments
TIMESTAMP(3)	TIMESTAMP	-
TIMESTAMP(4)	TIMESTAMP	-
TIMESTAMP(5)	TIMESTAMP	-
TIMESTAMP(6)	TIMESTAMP	-
TIMESTAMP(7)	TIMESTAMP	-
TIMESTAMP(8)	TIMESTAMP	-
TIMESTAMP(9)	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
VARCHAR2(*n*)	- VARCHAR(65000) Mapping condition: *((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000* - VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-

Oracle Source and DB2 for Linux, UNIX, and Windows Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a DB2 for Linux, UNIX, and Windows target:

Oracle Source Datatype	DB2 Target Datatype	Comments
BINARY_DOUBLE	DOUBLE	-
BINARY_FLOAT	FLOAT	-

Oracle Source Datatype	DB2 Target Datatype	Comments
BLOB	BLOB(1G)	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

CHAR(*n*)	- CHAR(*n*)
	Mapping condition:
	`n <= 254`
	- CLOB(*n*)
	Mapping condition:
	`254 < n`
		-

CLOB	CLOB(1G)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	TIMESTAMP	-
FLOAT	DECIMAL(31,15)	-
LONG	CLOB(1G)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
LONG RAW	BLOB(1G)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	DB2 Target Datatype	Comments

NCHAR(*n*)	- GRAPHIC(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`(n + SrcMinCharsetSize - 1) / SrcMinCharsetSize <= 127`
	- VARGRAPHIC(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
	Mapping condition:
	`(n + SrcMinCharsetSize - 1) / SrcMinCharsetSize > 127`
	- DBCLOB(32K)
	Mapping condition:
	`(n + SrcMinCharsetSize - 1) / SrcMinCharsetSize > 16352`
		-

NCLOB	DBCLOB(1G)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMBER(*p*,*s*)	- DECIMAL(31, 10)
	Mapping condition:
	`p == -1 and s == -1`
	- DECFLOAT(34)
	Mapping condition:
	`p > 31 and s != -1`
	- DECIMAL(31, 0)
	Mapping condition:
	`p <= 31 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 5 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 10 and s == 0`
	- BIGINT
	Mapping condition:
	`p <= 19 and s == 0`
	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

Oracle Source Datatype	DB2 Target Datatype	Comments
NVARCHAR(*n*)	VARGRAPHIC(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
NVARCHAR2(*n*)	- DBCLOB(32K) Mapping condition: *((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16336* - VARGRAPHIC(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
RAW	BLOB(1G)	Oracle piecewise operations are supported only for inline LOBs.
TIMESTAMP(0)	TIMESTAMP	-
TIMESTAMP(1)	TIMESTAMP	-
TIMESTAMP(2)	TIMESTAMP	-
TIMESTAMP(3)	TIMESTAMP	-
TIMESTAMP(4)	TIMESTAMP	-
TIMESTAMP(5)	TIMESTAMP	-
TIMESTAMP(6)	TIMESTAMP	-
TIMESTAMP(7)	TIMESTAMP	-
TIMESTAMP(8)	TIMESTAMP	-
TIMESTAMP(9)	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n*)	-
VARCHAR2(*n*)	- CLOB(32K) Mapping condition: *n > 32000* - VARCHAR(*n*)	Oracle piecewise operations are supported only for inline LOBs.

Oracle Source and Greenplum Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a Greenplum target:

Oracle Source Datatype	Greenplum Target Datatype	Comments
BINARY_DOUBLE	NUMERIC(37,15)	-
BINARY_FLOAT	NUMERIC(37,15)	-
BLOB	BYTEA	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHAR(*n*)	CHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
CLOB	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	TIMESTAMP(0)	-
FLOAT	NUMERIC(37,15)	-
LONG	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
LONG RAW	BYTEA	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	Greenplum Target Datatype	Comments
NCHAR(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NCLOB	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMBER(*p*,*s*)	- NUMERIC
	Mapping condition:
	`p == -1 and s == -1`
	- NUMERIC(39, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 4 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 9 and s == 0`
	- BIGINT
	Mapping condition:
	`p <= 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NVARCHAR2(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
RAW	BYTEA	Oracle piecewise operations are supported only for inline LOBs. The target value is truncated to the size that is specified in the global.lob_truncation_size runtime parameter.
TIMESTAMP(0)	TIMESTAMP(0)	-
TIMESTAMP(1)	TIMESTAMP(1)	-

Oracle Source Datatype	Greenplum Target Datatype	Comments
TIMESTAMP(2)	TIMESTAMP(2)	-
TIMESTAMP(3)	TIMESTAMP(3)	-
TIMESTAMP(4)	TIMESTAMP(4)	-
TIMESTAMP(5)	TIMESTAMP(5)	-
TIMESTAMP(6)	TIMESTAMP(6)	-
TIMESTAMP(7)	TIMESTAMP(6)	-
TIMESTAMP(8)	TIMESTAMP(6)	-
TIMESTAMP(9)	TIMESTAMP(6)	-
VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
VARCHAR2(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	Oracle piecewise operations are supported only for inline LOBs.

Oracle Source and Microsoft SQL Server Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a Microsoft SQL Server target:

Oracle Source Datatype	Microsoft SQL Server Target Datatype	Comments
BINARY_DOUBLE	FLOAT	-
BINARY_FLOAT	FLOAT	-
BLOB	IMAGE	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHAR(*n*)	CHAR(*n*)	-

Oracle Source Datatype	Microsoft SQL Server Target Datatype	Comments
CLOB	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATETIME2(0)	-
FLOAT	NUMERIC(38,15)	-
LONG	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
LONG RAW	IMAGE	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
NCHAR(*n*)	NCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
NCLOB	NTEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	Microsoft SQL Server Target Datatype	Comments

NUMBER(*p*,*s*)	- FLOAT
	Mapping condition:
	`p == -1 and s == -1`
	- NUMERIC(38, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 4 and s == 0`
	- INT
	Mapping condition:
	`p <= 9 and s == 0`
	- BIGINT
	Mapping condition:
	`p <= 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	NVARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

NVARCHAR2(*n*)	- NVARCHAR(MAX)
	Mapping condition:
	`n > 4000`
	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		-

RAW(*n*)	- VARBINARY(MAX) Mapping condition: `n > 2000` - VARBINARY(*n*)	Oracle piecewise operations are supported only for inline LOBs.
TIMESTAMP(0)	DATETIME2(0)	-
TIMESTAMP(1)	DATETIME2(1)	-
TIMESTAMP(2)	DATETIME2(2)	-
TIMESTAMP(3)	DATETIME2(3)	-
TIMESTAMP(4)	DATETIME2(4)	-
TIMESTAMP(5)	DATETIME2(5)	-
TIMESTAMP(6)	DATETIME2(6)	-

Oracle Source Datatype	Microsoft SQL Server Target Datatype	Comments
TIMESTAMP(7)	DATETIME2(7)	-
TIMESTAMP(8)	DATETIME2(7)	-
TIMESTAMP(9)	DATETIME2(7)	-
VARCHAR(*n*)	VARCHAR(*n*)	-
VARCHAR2(*n*)	- VARCHAR(MAX) Mapping condition: `n > 4000` - VARCHAR(*n*)	Oracle piecewise operations are supported only for inline LOBs.
XMLTYPE	XML	-

Oracle Source and MySQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a MySQL target:

Oracle Source Datatype	MySQL Target Datatype	Comments
BINARY_DOUBLE	DOUBLE	-
BINARY_FLOAT	DOUBLE	-
BLOB	BLOB	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

CHAR(*n*)	- CHAR(*n*)
	Mapping condition:
	`n < 256`
	- TEXT
	Mapping condition:
	`n >= 256`
		-

Oracle Source Datatype	MySQL Target Datatype	Comments
CLOB	LONGTEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATETIME	-
FLOAT	DOUBLE	-
LONG	LONGTEXT	Maximum supported size is 8 KB. Oracle piecewise operations are supported only for inline LOBs.
LONG RAW	BLOB	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NCHAR(*n*)	- VARCHAR(*n*)
	Mapping condition:
	`n < 256`
	- TEXT
	Mapping condition:
	`n >= 256`
		-

NCLOB	LONGTEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	MySQL Target Datatype	Comments

NUMBER(*p*,*s*)	- DECIMAL(38, 10)
	Mapping condition:
	`p == -1 and s == -1`
	- DECIMAL(38, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 4 and s == 0`
	- INT
	Mapping condition:
	`p <= 9 and s == 0`
	- BIGINT
	Mapping condition:
	`p <= 18 and s == 0`
	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	VARCHAR(*n*)	-
NVARCHAR2(*n*)	VARCHAR(*n*)	-
RAW(*n*)	VARBINARY(*n*)	Oracle piecewise operations are supported only for inline LOBs.
TIMESTAMP(0)	DATETIME	-
TIMESTAMP(1)	DATETIME	-
TIMESTAMP(2)	DATETIME	-
TIMESTAMP(3)	DATETIME	-
TIMESTAMP(4)	DATETIME	-
TIMESTAMP(5)	DATETIME	-
TIMESTAMP(6)	DATETIME	-
TIMESTAMP(7)	DATETIME	-
TIMESTAMP(8)	DATETIME	-
TIMESTAMP(9)	DATETIME	-

Oracle Source Datatype	MySQL Target Datatype	Comments
VARCHAR(*n*)	VARCHAR(*n*)	-
VARCHAR2(*n*)	VARCHAR(*n*)	Oracle piecewise operations are supported only for inline LOBs.

Oracle Source and Netezza Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a Netezza target:

Oracle Source Datatype	Netezza Target Datatype	Comments
BINARY_DOUBLE	DOUBLE PRECISION	When replicating an 8-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
BINARY_FLOAT	DOUBLE PRECISION	When replicating a 4-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
BLOB	VARCHAR(16000)	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance.
CHAR(*n*)	NCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
CLOB	NVARCHAR(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	TIMESTAMP	-
FLOAT	NUMERIC(37,15)	-

Oracle Source Datatype	Netezza Target Datatype	Comments
LONG	NVARCHAR(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
LONG RAW	VARCHAR(16000)	-
NCHAR(*n*)	NCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NCLOB	NVARCHAR(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMBER(*p*,*s*)	- NUMERIC(38, 10)
	Mapping condition:
	`p == -1 and s == -1`
	- NUMERIC(38, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 4 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 9 and s == 0`
	- BIGINT
	Mapping condition:
	`p <= 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	NVARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

Oracle Source Datatype	Netezza Target Datatype	Comments

NVARCHAR2(*n*)	- NVARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		Maximum length of Netezza NVARCHAR datatype is 16000 characters.

RAW(*n*)	VARCHAR(*n*)	-
TIMESTAMP(0)	TIMESTAMP	-
TIMESTAMP(1)	TIMESTAMP	-
TIMESTAMP(2)	TIMESTAMP	-
TIMESTAMP(3)	TIMESTAMP	-
TIMESTAMP(4)	TIMESTAMP	-
TIMESTAMP(5)	TIMESTAMP	-
TIMESTAMP(6)	TIMESTAMP	-
TIMESTAMP(7)	TIMESTAMP	-
TIMESTAMP(8)	TIMESTAMP	-
TIMESTAMP(9)	TIMESTAMP	-
VARCHAR(*n*)	NVARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

VARCHAR2(*n*)	- NVARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
	- NVARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		Maximum length of Netezza VARCHAR datatype is 64000 characters.
		Oracle piecewise operations are supported only for inline LOBs.

Because Netezza does not use binary dataypes, Data Replication does not support replication of Oracle BLOB, LONG RAW, and RAW datatypes to Netezza targets.

Oracle Source and Oracle Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and an Oracle target:

Oracle Source Datatype	Oracle Target Datatype	Comments
BINARY_DOUBLE	BINARY_DOUBLE	When replicating an 8-byte floating-point numeric value to a BINARY_DOUBLE column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
BINARY_FLOAT	BINARY_FLOAT	When replicating a 4-byte floating-point numeric value to a BINARY_FLOAT column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
BLOB	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported for SQL Apply mode only. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHAR(*n*)	CHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

Oracle Source Datatype	Oracle Target Datatype	Comments
CLOB	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. Oracle piecewise operations are supported for SQL Apply mode only. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	DATE	-
FLOAT(*p*)	FLOAT(*p*)	-
INTERVAL DAY TO SECOND	INTERVAL DAY (%s) TO SECOND (%s)	-

INTERVAL YEAR TO MONTH	- INTERVAL YEAR(1) TO MONTH
	Mapping condition:
	`p == 1`
	- INTERVAL YEAR(2) TO MONTH
	Mapping condition:
	`p == 2`
	- INTERVAL YEAR(3) TO MONTH
	Mapping condition:
	`p == 3`
	- INTERVAL YEAR(4) TO MONTH
	Mapping condition:
	`p == 4`
	- INTERVAL YEAR(5) TO MONTH
	Mapping condition:
	`p == 5`
	- INTERVAL YEAR(6) TO MONTH
	Mapping condition:
	`p == 6`
	- INTERVAL YEAR(7) TO MONTH
	Mapping condition:
	`p == 7`
	- INTERVAL YEAR(8) TO MONTH
	Mapping condition:
	`p == 8`
	- INTERVAL YEAR(9) TO MONTH
	Mapping condition:
	`p == 9`
		-

Oracle Source Datatype	Oracle Target Datatype	Comments
LONG	CLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. For Merge Apply mappings, map source columns with the LONG datatype to target columns with the VARCHAR(40000) datatype. LONG columns are not supported for materializing Oracle targets with DBLinks. Maximum supported size is 8 KB. Oracle piecewise operations are supported for SQL Apply mode only.
LONG RAW	BLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. LONG RAW columns are not supported for materializing Oracle targets with DBLinks. Oracle piecewise operations are supported for SQL Apply mode only. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
NCHAR(*n*)	NCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
NCLOB	NCLOB	By default, the Applier cannot insert change data into LOB and LONG columns if the total row data size exceeds 4000 bytes. In this case, set the apply.oracle.use_returning_into runtime parameter to 1 to configure the Applier to use the RETURNING INTO clause to insert change data into LOB and LONG columns. Oracle piecewise operations are supported for SQL Apply mode only. The Applier cannot load change data to Oracle LOB columns in Merge Apply mode. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	Oracle Target Datatype	Comments

NUMBER(*p*,*s*)	- NUMBER
	Mapping condition:
	`p == -1 and s == -1`
	- NUMBER(*, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- NUMBER(*p*,*s*)
	Mapping condition:
	`p != 0 and s == 0`
	- NUMBER(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	NVARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NVARCHAR2(*n*)	NVARCHAR2(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

RAW(*n*)	- RAW(*n*)
	Mapping condition:
	`n <= 2000`
	- BLOB
	Mapping condition:
	`n > 2000`
		Oracle piecewise operations are supported for SQL Apply mode only.

TIMESTAMP(0)	TIMESTAMP(0)	-
TIMESTAMP(0) WITH LOCAL TIME ZONE	TIMESTAMP(0) WITH LOCAL TIME ZONE	-
TIMESTAMP(0) WITH TIME ZONE	TIMESTAMP(0) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(1)	TIMESTAMP(1)	-
TIMESTAMP(1) WITH LOCAL TIME ZONE	TIMESTAMP(1) WITH LOCAL TIME ZONE	-
TIMESTAMP(1) WITH TIME ZONE	TIMESTAMP(1) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(2)	TIMESTAMP(2)	-
TIMESTAMP(2) WITH LOCAL TIME ZONE	TIMESTAMP(2) WITH LOCAL TIME ZONE	-
TIMESTAMP(2) WITH TIME ZONE	TIMESTAMP(2) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(3)	TIMESTAMP(3)	-

Oracle Source Datatype	Oracle Target Datatype	Comments
TIMESTAMP(3) WITH LOCAL TIME ZONE	TIMESTAMP(3) WITH LOCAL TIME ZONE	-
TIMESTAMP(3) WITH TIME ZONE	TIMESTAMP(3) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(4)	TIMESTAMP(4)	-
TIMESTAMP(4) WITH LOCAL TIME ZONE	TIMESTAMP(4) WITH LOCAL TIME ZONE	-
TIMESTAMP(4) WITH TIME ZONE	TIMESTAMP(4) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(5)	TIMESTAMP(5)	-
TIMESTAMP(5) WITH LOCAL TIME ZONE	TIMESTAMP(5) WITH LOCAL TIME ZONE	-
TIMESTAMP(5) WITH TIME ZONE	TIMESTAMP(5) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(6)	TIMESTAMP(6)	-
TIMESTAMP(6) WITH LOCAL TIME ZONE	TIMESTAMP(6) WITH LOCAL TIME ZONE	-
TIMESTAMP(6) WITH TIME ZONE	TIMESTAMP(6) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(7)	TIMESTAMP(7)	-
TIMESTAMP(7) WITH LOCAL TIME ZONE	TIMESTAMP(7) WITH LOCAL TIME ZONE	-
TIMESTAMP(7) WITH TIME ZONE	TIMESTAMP(7) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(8)	TIMESTAMP(8)	-
TIMESTAMP(8) WITH LOCAL TIME ZONE	TIMESTAMP(8) WITH LOCAL TIME ZONE	-
TIMESTAMP(8) WITH TIME ZONE	TIMESTAMP(8) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.
TIMESTAMP(9)	TIMESTAMP(9)	-
TIMESTAMP(9) WITH LOCAL TIME ZONE	TIMESTAMP(9) WITH LOCAL TIME ZONE	-
TIMESTAMP(9) WITH TIME ZONE	TIMESTAMP(9) WITH TIME ZONE	Oracle TIMESTAMP WITH TIME ZONE in the time zone region (TZR) format is not supported.

Oracle Source Datatype	Oracle Target Datatype	Comments
VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
VARCHAR2(*n*)	VARCHAR2(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	Oracle piecewise operations are supported for SQL Apply mode only.

Oracle Source and PostgreSQL Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a PostgreSQL target:

Oracle Source Datatype	PostgreSQL Target Datatype	Comments
BINARY_DOUBLE	DOUBLE PRECISION	-
BINARY_FLOAT	REAL	-
BLOB	BYTEA	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHAR(*n*)	CHAR(*n*)	-
CLOB	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	TIMESTAMP(0)	-
FLOAT(*p*,*s*)	NUMERIC(*p'*,*s'*) *p' = -1* *s' = -1*	-

Oracle Source Datatype	PostgreSQL Target Datatype	Comments
LONG	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
LONG RAW	BYTEA	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
NCHAR(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-
NCLOB	TEXT	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMBER(*p*,*s*)	- NUMERIC
	Mapping condition:
	`p == -1 and s == -1`
	- NUMERIC(39, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 4 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 9 and s == 0`
	- BIGINT
	Mapping condition:
	`p <= 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		The Data Replication Console might map NUMBER source columns to BIGINT target columns. If you run the Applier on Linux or UNIX, ensure that the replication configuration does not map source columns to BIGINT target columns. The Applier ends with an error when applying data to BIGINT columns because the DataDirect ODBC driver for PostgreSQL does not support this datatype.

Oracle Source Datatype	PostgreSQL Target Datatype	Comments
NVARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
NVARCHAR2(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
RAW	BYTEA	Oracle piecewise operations are supported only for inline LOBs.
TIMESTAMP(0)	TIMESTAMP(0) WITHOUT TIME ZONE	-
TIMESTAMP(1)	TIMESTAMP(1) WITHOUT TIME ZONE	-
TIMESTAMP(2)	TIMESTAMP(2) WITHOUT TIME ZONE	-
TIMESTAMP(3)	TIMESTAMP(3) WITHOUT TIME ZONE	-
TIMESTAMP(4)	TIMESTAMP(4) WITHOUT TIME ZONE	-
TIMESTAMP(5)	TIMESTAMP(5) WITHOUT TIME ZONE	-
TIMESTAMP(6)	TIMESTAMP(6) WITHOUT TIME ZONE	-
TIMESTAMP(7)	TIMESTAMP(6) WITHOUT TIME ZONE	-
TIMESTAMP(8)	TIMESTAMP(6) WITHOUT TIME ZONE	-
TIMESTAMP(9)	TIMESTAMP(6) WITHOUT TIME ZONE	-
VARCHAR(*n*)	VARCHAR(*n*)	-
VARCHAR2(*n*)	VARCHAR(*n*)	Oracle piecewise operations are supported only for inline LOBs.

Oracle Source and Teradata Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a Teradata target:

Oracle Source Datatype	Teradata Target Datatype	Comments
BINARY_DOUBLE	DOUBLE PRECISION	When replicating an 8-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 15 digits after the decimal point by default. Use the apply.merge.double_precision runtime parameter to change the default precision value.
BINARY_FLOAT	DOUBLE PRECISION	When replicating a 4-byte floating-point numeric value to a DOUBLE PRECISION column on the target, the Applier rounds this value and preserves only 8 digits after the decimal point by default. Use the apply.merge.float_precision runtime parameter to change the default precision value.
BLOB	VARBYTE(16000)	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
CHAR(*n*)	CHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
CLOB	VARCHAR(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	TIMESTAMP(0)	Mappings to Teradata TIME and DATE datatypes are not supported.
FLOAT	DECIMAL(38, 10)	-
LONG	VARCHAR(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	Teradata Target Datatype	Comments
LONG RAW	VARBYTE(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
NCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-
NCLOB	VARCHAR(16000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMBER(*p*,*s*)	- DECIMAL(38, 10)
	Mapping condition:
	`p == -1 and s == -1 and tgt_version < 14`
	- NUMBER
	Mapping condition:
	`p == -1 and s == -1 and tgt_version > 13`
	- DECIMAL(38, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- SMALLINT
	Mapping condition:
	`p <= 4 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 9 and s == 0`
	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p > 9 and s == 0`
	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize*	-

Oracle Source Datatype	Teradata Target Datatype	Comments

NVARCHAR2(*n*)	- VARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		-

RAW(*n*)	- VARBYTE(16000) Mapping condition: `n > 16000` - VARBYTE(*n*)	-
TIMESTAMP(0)	TIMESTAMP(0)	-
TIMESTAMP(1)	TIMESTAMP(1)	-
TIMESTAMP(2)	TIMESTAMP(2)	-
TIMESTAMP(3)	TIMESTAMP(3)	-
TIMESTAMP(4)	TIMESTAMP(4)	-
TIMESTAMP(5)	TIMESTAMP(5)	-
TIMESTAMP(6)	TIMESTAMP(6)	-
TIMESTAMP(7)	TIMESTAMP(6)	-
TIMESTAMP(8)	TIMESTAMP(6)	-
TIMESTAMP(9)	TIMESTAMP(6)	-
VARCHAR(*n*)	VARCHAR(*n'*) `n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`	-

VARCHAR2(*n*)	- VARCHAR(16000)
	Mapping condition:
	`((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize) > 16000`
	- VARCHAR(*n'*)
	`n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize`
		Oracle piecewise operations are supported only for inline LOBs.

Oracle Source and Vertica Target

The following table identifies the recommended datatype mappings for Data Replication configurations with an Oracle source and a Vertica target:

Oracle Source Datatype	Vertica Target Datatype	Comments
BINARY_DOUBLE	NUMERIC(37,15)	-
BINARY_FLOAT	NUMERIC(37,15)	-
BLOB	VARBINARY(65000)	If the global.lob_truncation_size runtime parameter is set to 64000 bytes or less, InitialSync handles BLOB data as RAW data to improve performance. Oracle piecewise operations are supported only for inline LOBs. The Extractor truncates the replicated value to 50 MB. The Applier truncates the replicated value to the size that is specified in the global.lob_truncation_size runtime parameter.
CHAR(*n*)	CHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
CLOB	VARCHAR(65000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
DATE	TIMESTAMP(0)	-
FLOAT	NUMERIC(37,15)	-
LONG	VARCHAR(65000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
LONG RAW	VARBINARY(65000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.
NCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-

Oracle Source Datatype	Vertica Target Datatype	Comments
NCLOB	VARCHAR(65000)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

NUMBER(*p*,*s*)	- NUMERIC(37, 10)
	Mapping condition:
	`p == -1 and s == -1`
	- NUMERIC(39, 0)
	Mapping condition:
	`p == -1 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 4 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 9 and s == 0`
	- INTEGER
	Mapping condition:
	`p <= 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p > 18 and s == 0`
	- NUMERIC(*p*,*s*)
	Mapping condition:
	`p != 0 and s != 0`
		-

NVARCHAR(*n*)	VARCHAR(*n'*) *n'* = (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* x *TgtCharsetSize*	-
NVARCHAR2(*n*)	- VARCHAR(65000) Mapping condition: ((*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* x *TgtCharsetSize*) > 65000 - VARCHAR(*n'*) *n'* = (*n* + *SrcMinCharsetSize* - 1) / *SrcMinCharsetSize* x *TgtCharsetSize*	-
RAW(*n*)	VARBINARY(*n*)	Oracle piecewise operations are supported only for inline LOBs. Use the global.lob_truncation_size runtime parameter to set the maximum size of a source LOB value that the Applier and InitialSync can replicate to the target. Maximum supported size for change data capture is 50 MB.

Oracle Source Datatype	Vertica Target Datatype	Comments
TIMESTAMP(0)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(1)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(2)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(3)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(4)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(5)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(6)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(7)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(8)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
TIMESTAMP(9)	TIMESTAMP	For Vertica targets, the maximum number of fractional digits in the seconds field is 6.
VARCHAR(*n*)	VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	-
VARCHAR2(*n*)	- VARCHAR(65000) Mapping condition: *((n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize) > 65000* - VARCHAR(*n'*) *n' = (n + SrcMinCharsetSize - 1) / SrcMinCharsetSize x TgtCharsetSize*	Oracle piecewise operations are supported only for inline LOBs.

Oracle Datatypes with Limited Support

Data Replication supports the following Oracle datatypes only for configurations that replicate data to Apache Kafka, Cloudera, Flat File, Hortonworks, or Oracle targets:

- INTERVAL DAY TO SECOND
- INTERVAL YEAR TO MONTH
- TIMESTAMP WITH LOCAL TIME ZONE

- TIMESTAMP WITH TIME ZONE

Unsupported Oracle Datatypes

Data Replication does not support data extraction for the following Oracle source datatypes:

- ANY types
- BFILE
- Expression Filter Types
- Extended types
- Media types
- MLSLABEL
- REF
- ROWID
- Spatial types
- URI types
- UROWID
- User-defined types
- XMLType

  Note: The Oracle Extractor can now extract data from XMLTYPE columns that are defined with the STORE AS CLOB clause.
- Binary datatypes to Amazon Redshift targets

CHAPTER 6

Datatype Mappings for Virtual Source Columns

This chapter includes the following topics:

Virtual Column Datatypes

The following table describes the virtual column datatypes:

Datatype	Description
BIGINT	Big integers. Storage size: 8 bytes Range of values: -2^63 (-9223372036854775808) through 2^63-1 (9223372036854775807)
DATE	Combined date and time value. Date format for Tcl scripts: yyyy-mm-dd hh:mm:ss Date format for SQL expressions: target database format

Datatype	Description
DECIMAL(*p*,*s*)	Decimal numbers with the declared precision and scale. Scale must be less than or equal to precision. Valid precision values: 1 through 38 Valid scale values: 0 through 38
DOUBLE	Double-precision floating-point numbers. Storage size: 8 bytes Range of values: -7.2E+75 through 7.2E+75
FLOAT	Single-precision floating-point numbers. Storage size: 4 bytes Range of values: -3.40E + 38 through -1.18E - 38, 0 and 1.18E - 38 through 3.40E + 38
INTEGER	Large integers. Storage size: 4 bytes Range of values: -2^31 (-2147483648) through 2^31 - 1 (2147483647)
NVARCHAR(*n*)	Variable-length Unicode data. Valid length values: 1 through 4000 bytes
TIMESTAMP(*f*)	Date and time value that includes the year, month, day, hour, minutes, and seconds. Timestamp format for Tcl scripts: yyyy-mm-dd hh:mm:ss ff Timestamp format for SQL expressions: target database format The *f* value is the number of digits in the fractional part of seconds. Valid fractional seconds precision values: 0 through 9
VARCHAR(*n*)	Variable-length non-Unicode data. Valid length values: 1 through 4000 bytes

Virtual Source Columns and Amazon Redshift Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Amazon Redshift target columns:

Virtual Source Datatype	Amazon Redshift Target Datatype	Comments
BIGINT	BIGINT	-
DATE	TIMESTAMP WITHOUT TIME ZONE	-
DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	DOUBLE PRECISION	-

Virtual Source Datatype	Amazon Redshift Target Datatype	Comments
FLOAT	REAL	-
INTEGER	INTEGER	-
NVARCHAR(*n*)	CHARACTER VARYING(*n*)	-
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	-
VARCHAR(*n*)	CHARACTER VARYING(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and DB2 for Linux, UNIX, and Windows Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and DB2 target columns:

Virtual Source Datatype	DB2 Target Datatype	Comments
BIGINT	DECIMAL(19)	-
DATE	TIMESTAMP	-

DECIMAL(*p*,*s*)	- DECIMAL(31, 10)
	Mapping condition:
	`p == -1 and s == -1`
	- DECFLOAT(34)
	Mapping condition:
	`p > 31 and s != -1`
	- DECIMAL(31, 0)
	Mapping condition:
	`p <= 31 and s == 0`
	- DECIMAL(*p*,*s*)
		-

DOUBLE	- DOUBLE - DECFLOAT	-
FLOAT	DOUBLE	-
INTEGER	DECIMAL(10)	-
NVARCHAR(*n*)	VARGRAPHIC(*n*)	-

Virtual Source Datatype	DB2 Target Datatype	Comments
TIMESTAMP	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and Greenplum Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Greenplum target columns:

Virtual Source Datatype	Greenplum Target Datatype	Comments
BIGINT	BIGINT	-
DATE	TIMESTAMP WITHOUT TIME ZONE	-
DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	DOUBLE PRECISION	-
FLOAT	REAL	-
INTEGER	INTEGER	-
NVARCHAR(*n*)	CHARACTER VARYING(*n*)	-
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	-
VARCHAR(*n*)	CHARACTER VARYING(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and Microsoft SQL Server Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Microsoft SQL Server target columns:

Virtual Source Datatype	Microsoft SQL Server Target Datatype	Comments
BIGINT	BIGINT	-
DATE	DATETIME	-
DECIMAL(*p*,*s*)	DECIMAL(*p*,*s*)	-
DOUBLE	FLOAT(53)	-
FLOAT	REAL	-
INTEGER	INT	-
NVARCHAR(*n*)	NVARCHAR(*n*)	-

Virtual Source Datatype	Microsoft SQL Server Target Datatype	Comments

TIMESTAMP	- DATETIME2(0)
	Mapping condition:
	`s == 0`
	- DATETIME2(1)
	Mapping condition:
	`s == 1`
	- DATETIME2(2)
	Mapping condition:
	`s == 2`
	- DATETIME2(3)
	Mapping condition:
	`s == 3`
	- DATETIME2(4)
	Mapping condition:
	`s == 4`
	- DATETIME2(5)
	Mapping condition:
	`s == 5`
	- DATETIME2(6)
	Mapping condition:
	`s == 6`
	- DATETIME2(7)
	Mapping condition:
	`s == 7`
	- DATETIME2(7)
		-

VARCHAR(*n*)	VARCHAR(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and MySQL Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and MySQL target columns:

Virtual Source Datatype	MySQL Target Datatype	Comments
BIGINT	BIGINT	-
DATE	DATETIME	-
DECIMAL(*p*,*s*)	DECIMAL(*p*,*s*)	-

Virtual Source Datatype	MySQL Target Datatype	Comments
DOUBLE	DOUBLE	-
FLOAT	FLOAT	-
INTEGER	INT	-
NVARCHAR(*n*)	VARCHAR(*n*)	-
TIMESTAMP	DATETIME	-
VARCHAR(*n*)	VARCHAR(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and Netezza Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Netezza target columns:

Virtual Source Datatype	Netezza Target Datatype	Comments
BIGINT	BIGINT	-
DATE	DATE	-
DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	DOUBLE PRECISION	-
FLOAT	REAL	-
INTEGER	INTEGER	-
NVARCHAR(*n*)	NVARCHAR(*n*)	-
TIMESTAMP	TIMESTAMP	-
VARCHAR(*n*)	NVARCHAR(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and Oracle Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Oracle target columns:

Virtual Source Datatype	Oracle Target Datatype	Comments
BIGINT	NUMBER(19, 0)	-
DATE	DATE	-
DECIMAL(*p*,*s*)	NUMBER(*p*,*s*)	-
DOUBLE	FLOAT(64)	-
FLOAT	REAL	-
INTEGER	NUMBER(10, 0)	-
NVARCHAR(*n*)	NVARCHAR2(*n*)	-
TIMESTAMP	TIMESTAMP(9)	-
VARCHAR(*n*)	VARCHAR2(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and PostgreSQL Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and PostgreSQL target columns:

Virtual Source Datatype	PostgreSQL Target Datatype	Comments
BIGINT	BIGINT	By default, the Data Replication Console maps BIGINT virtual source columns to BIGINT columns on the target. If you run the Applier on Linux or UNIX, map BIGINT source columns to NUMERIC target columns. The Applier ends with an error when applying data to BIGINT columns because the DataDirect ODBC driver for PostgreSQL does not support this datatype.
DATE	TIMESTAMP WITHOUT TIME ZONE	-
DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-

Virtual Source Datatype	PostgreSQL Target Datatype	Comments
DOUBLE	DOUBLE PRECISION	-
FLOAT	REAL	-
INTEGER	INTEGER	-
NVARCHAR(*n*)	CHARACTER VARYING(*n*)	-
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	-
VARCHAR(*n*)	CHARACTER VARYING(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and Teradata Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Teradata target columns:

Virtual Source Datatype	Teradata Target Datatype	Comments
BIGINT	DECIMAL(20, 0)	-
DATE	TIMESTAMP(0)	-

DECIMAL(*p*,*s*)	- DECIMAL(*p*,*s*)
	Mapping condition:
	`p <= 38 and s <= 10`
	- DECIMAL(38, 10)
	Mapping condition:
	`p == -1 and s == -1`
		-

DOUBLE	DOUBLE PRECISION	-
FLOAT	FLOAT	-
INTEGER	INTEGER	-
NVARCHAR(*n*)	VARCHAR(*n*)	-

Virtual Source Datatype	Teradata Target Datatype	Comments
TIMESTAMP	TIMESTAMP(6)	-
VARCHAR(*n*)	VARCHAR(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

Virtual Source Columns and Vertica Target Columns

The following table identifies the recommended datatype mappings for virtual source columns and Vertica target columns:

Virtual Source Datatype	Vertica Target Datatype	Comments
BIGINT	BIGINT	-
DATE	DATE	-
DECIMAL(*p*,*s*)	NUMERIC(*p*,*s*)	-
DOUBLE	FLOAT	-
FLOAT	FLOAT	-
INTEGER	INTEGER	-
NVARCHAR(*n*)	VARCHAR(*n*)	-
TIMESTAMP	TIMESTAMP	-
VARCHAR(*n*)	VARCHAR(*n*)	If a SQL script gets values from CHAR or NCHAR source columns that include trailing space characters, the Applier and InitialSync truncate the trailing spaces on the target. The truncation occurs in Merge Apply mode or in Audit Apply mode if the apply.direct_load_for_audit_tables runtime parameter is set to 1.

INDEX

P

T

V