



Informatica® PowerExchange for MapR-DB
10.4.0

User Guide

Informatica PowerExchange for MapR-DB User Guide
10.4.0
December 2019

© Copyright Informatica LLC 2017, 2019

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2019-12-12

Table of Contents

Preface	5
Informatica Resources.	5
Informatica Network.	5
Informatica Knowledge Base.	5
Informatica Documentation.	5
Informatica Product Availability Matrices.	6
Informatica Velocity.	6
Informatica Marketplace.	6
Informatica Global Customer Support.	6
 Chapter 1: Introduction to PowerExchange for MapR-DB.....	 7
PowerExchange for MapR-DB Overview.	7
PowerExchange for MapR-DB Implementation.	7
PowerExchange for MapR-DB Configuration.	8
Prerequisites.	8
 Chapter 2: HBase Connections for MapR-DB.....	 9
HBase Connections for MapR-DB Overview.	9
HBase Connection Properties for MapR-DB.	9
Creating an HBase Connection for MapR-DB.	10
 Chapter 3: HBase Data Objects for MapR-DB.....	 11
HBase Data Objects for MapR-DB Overview.	11
Data Object Column Configuration.	12
Add Columns.	12
Search and Add Columns.	12
Get All Columns.	12
HBase Data Object Properties for MapR-DB.	13
HBase Data Object Read Operation Properties for MapR-DB.	14
HBase Data Object Write Operation Properties for MapR-DB.	14
Partitioning.	15
Configuring Maximum Parallelism for MapR-DB Mappings.	15
Parameterization of HBase Data Objects for MapR-DB.	16
Creating an HBase Data Object for MapR-DB.	16
Creating an HBase Data Object Operation for MapR-DB.	17
 Chapter 4: MapR-DB Mappings.....	 19
MapR-DB Mappings Overview.	19
Filtering MapR-DB Source Data.	19

Appendix A: HBase Data Types Reference..... 21

HBase Data Type Reference Overview. 21

HBase and Transformation Data Types. 22

Index..... 23

Preface

Use the *Informatica® PowerExchange® for MapR-DB User Guide* to learn how to read from or write to MapR-DB binary tables by using the Developer tool. Learn to create a connection and develop and run mappings in the native environment and Hadoop environments.

Informatica Resources

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

Informatica Network

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

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

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

Informatica Knowledge Base

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

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

Informatica Documentation

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

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

Informatica Product Availability Matrices

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

Informatica Velocity

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

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

Informatica Marketplace

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

Informatica Global Customer Support

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

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

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

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

CHAPTER 1

Introduction to PowerExchange for MapR-DB

This chapter includes the following topics:

- [PowerExchange for MapR-DB Overview, 7](#)
- [PowerExchange for MapR-DB Implementation, 7](#)
- [PowerExchange for MapR-DB Configuration, 8](#)

PowerExchange for MapR-DB Overview

PowerExchange for MapR-DB provides connectivity to MapR-DB binary tables.

Use PowerExchange for MapR-DB to read data from column families in a MapR-DB binary table or write data to column families in a MapR-DB binary table. You can read data from or write data to a column family or a single binary column.

You can connect to a MapR-DB data store and view all the MapR-DB binary tables. When you select a MapR-DB table, you can view all the column families. If you know the schema of the data source, you can specify the columns in the column families when you create the data object. If you do not know the schema of the data source, you can search the rows in a MapR-DB table to identify the columns and their occurrence probability.

You can validate and run MapR-DB mappings in the native environment or on the Blaze and Spark engines in the Hadoop environment.

PowerExchange for MapR-DB Implementation

PowerExchange for MapR-DB uses the HBase API to connect to MapR-DB.

To connect to a MapR-DB table, you must create an HBase connection in which you specify the database type as MapR-DB and define the database path of the MapR-DB table. You must then create an HBase data object based on the MapR-DB table, and create an HBase data object read or write operation. Add the read or write operation to a mapping to read data from or write data to MapR-DB.

PowerExchange for MapR-DB Configuration

PowerExchange for MapR-DB installs with Informatica. You enable PowerExchange for MapR-DB with a license key.

Before you use PowerExchange for MapR-DB, complete the prerequisites.

Prerequisites

Before you use PowerExchange for MapR-DB, complete the following prerequisite tasks:

- Install and configure Informatica Services. Verify that the domain has a Data Integration Service and a Model Repository Service.
- Verify that a cluster configuration is created in the domain.
- Verify that a Metadata Access Service is created in the domain.
- Get a license for PowerExchange for MapR-DB.
- Verify that the Data Integration Service user and the user who runs MapR-DB mappings have read and write permissions on the column families in the MapR-DB source and target tables.

CHAPTER 2

HBase Connections for MapR-DB

This chapter includes the following topics:

- [HBase Connections for MapR-DB Overview, 9](#)
- [HBase Connection Properties for MapR-DB, 9](#)
- [Creating an HBase Connection for MapR-DB, 10](#)

HBase Connections for MapR-DB Overview

To connect to a MapR-DB table, you must create an HBase connection in which you must specify the database type as MapR-DB.

Use the Developer tool, Administrator tool, or infacmd to create an HBase connection for MapR-DB.

HBase Connection Properties for MapR-DB

Use an HBase connection to connect to a MapR-DB table. The HBase connection is a NoSQL connection. You can create and manage an HBase connection in the Administrator tool or the Developer tool. HBase connection properties are case sensitive unless otherwise noted.

The following table describes the HBase connection properties for MapR-DB:

Property	Description
Name	Name of the connection. The name is not case sensitive and must be unique within the domain. You can change this property after you create the connection. The name cannot exceed 128 characters, contain spaces, or contain the following special characters: ~ ` ! \$ % ^ & * () - + = { [] } \ : ; " ' < , > . ? /
ID	String that the Data Integration Service uses to identify the connection. The ID is not case sensitive. It must be 255 characters or less and must be unique in the domain. You cannot change this property after you create the connection. Default value is the connection name.
Description	Description of the connection. The description cannot exceed 4,000 characters.

Property	Description
Location	Domain where you want to create the connection.
Type	Connection type. Select HBase .
Database Type	Type of database that you want to connect to. Select MapR-DB to create a connection for a MapR-DB table.
Cluster Configuration	The name of the cluster configuration associated with the Hadoop environment.
MapR-DB Database Path	Database path that contains the MapR-DB table that you want to connect to. Enter a valid MapR cluster path. When you create an HBase data object for MapR-DB, you can browse only tables that exist in the MapR-DB path that you specify in the Database Path field. You cannot access tables that are available in sub-directories in the specified path. For example, if you specify the path as <code>/user/customers/</code> , you can access the tables in the <code>customers</code> directory. However, if the <code>customers</code> directory contains a sub-directory named <code>regions</code> , you cannot access the tables in the following directory: <code>/user/customers/regions</code>

Creating an HBase Connection for MapR-DB

Create an HBase connection to import a MapR-DB table into the Developer tool.

1. In the Developer tool, click **Window > Preferences**.
2. Select **Informatica > Connections**.
3. Expand the domain in the **Available Connections** list.
4. Select the connection type as **NoSQL > HBase**, and then click **Add**.
5. Enter a connection name.
6. Optionally, enter a connection ID and description.
7. Select the domain where you want to create the connection.
8. Select **HBase** as the connection type.
9. Click **Next**.
10. Select the database type as **MapR-DB**.
11. Configure the connection properties.
12. Click **Test Connection**. If a default Metadata Access Service is not set, a message appears to configure the Metadata Access Service. Click **OK** and set one Metadata Access Service as default. After you set a default Metadata Access Service, connection to the MapR-DB table is tested. If the Metadata Access Service does not exist, contact the Informatica administrator to create a new Metadata Access Service.
13. Click **Finish**.

CHAPTER 3

HBase Data Objects for MapR-DB

This chapter includes the following topics:

- [HBase Data Objects for MapR-DB Overview, 11](#)
- [Data Object Column Configuration, 12](#)
- [HBase Data Object Properties for MapR-DB, 13](#)
- [HBase Data Object Read Operation Properties for MapR-DB, 14](#)
- [HBase Data Object Write Operation Properties for MapR-DB, 14](#)
- [Partitioning, 15](#)
- [Parameterization of HBase Data Objects for MapR-DB, 16](#)
- [Creating an HBase Data Object for MapR-DB, 16](#)
- [Creating an HBase Data Object Operation for MapR-DB, 17](#)

HBase Data Objects for MapR-DB Overview

Use an HBase data object to import a MapR-DB table. Then, add the MapR-DB table to an HBase data object operation, and add the operation to a mapping to read or write data.

When you create an HBase connection, select the database type as **MapR-DB** and define the MapR-DB path in which the MapR-DB table exists. Then, create an HBase data object for the MapR-DB table.

When you create an HBase data object, you can select the MapR-DB table and view all the column families in the table. You can specify the column names in the column family if you know the column name and data type, or you can search the rows in the MapR-DB table and specify the columns.

You can read data from and write data to a column family or a single binary column. When you create the data object, specify the column families to read or choose to get all the data as a single stream of binary data. You can also specify the column families to which you want to write or choose to write all the data as a single stream of binary data.

Create an HBase data object read operation to read data from the MapR-DB column families. Create an HBase data object write operation to insert data to the MapR-DB column families.

Data Object Column Configuration

When you want to read data from or write data to columns in a column family, you can specify the columns when you create the HBase data object for MapR-DB.

You can add the columns in the column families or you can search for the columns names in the column family and select the columns. You can also choose to read from or write to a single binary port.

Add Columns

When you create a data object, you can specify the columns in one or more column families in a MapR-DB table.

When you add a MapR-DB table as the resource for an HBase data object, all the column families in the MapR-DB table appear. If you know the details of the columns in the column families, you can select a column family and add the column details. Column details include column name, data type, precision, and scale.

Although data is stored in binary format in MapR-DB tables, you can specify the associated data type of the column to transform the data. To avoid data errors or incorrect data, verify that you specify the correct data type for the columns.

Verify that you specify valid column details when you add columns to avoid unexpected run-time behavior. If you add a column that does not exist in the column family and create a data object read operation, the Data Integration Service returns a null value for the column at run time. If you do not specify a value for a column when you write data to a MapR-DB table, the Data Integration Service specifies a null value for the column at run time.

If the MapR-DB table has more than one column family, you can add column details for multiple column families when you create the data object. Select one column family at a time and add the columns details. The column family name is the prefix for all the columns in the column family for unique identification.

Search and Add Columns

When you create a data object, you can search the rows in a MapR-DB table to identify the column in the table and select the columns you want to add.

When you do not know the columns in a MapR-DB table, you can search the rows in the table to identify all the columns and the occurrence percentage of the column. You can infer if the column name is valid based on the number of times the column occurs in the table. For example, if column name eName occurs rarely while column name empName occurs in a majority of rows, you can infer the column name as empName.

When you search and add columns, you can specify the maximum number of rows to search and the occurrence percentage value for a column. If you specify the maximum number of rows as 100 and the column occurrence percent as 90, all columns that appear at least 90 times in 100 rows appear in the results. You can select the columns in the results to add the columns to the data object.

Get All Columns

Binary data or data that can be converted to a byte array can be stored in a MapR-DB column. You can read data from and write to MapR-DB tables in bytes.

When you create a data object, you can choose to get all the columns in a column family as a single stream of binary data.

After you create an HBase data object read operation, you can add the operation as a **Read** transformation in a mapping to read data in all the columns in the column family as a single stream of binary data.

After you create an HBase data object write operation, you can add the operation as a **Write** transformation in a mapping to write data in all the columns in the column family as a single column of binary data.

HBase Data Object Properties for MapR-DB

You can configure the HBase data object properties for MapR-DB when you create the data object.

General Properties

The following table describes the general properties that you configure for HBase data objects for MapR-DB:

Property	Description
Name	Name of the HBase data object.
Location	The project or folder in the Model repository where you want to store the HBase data object.
Connection	Name of the HBase connection that you created to connect to the MapR-DB table.

Add Column Properties

In the **Column Families** dialog box, select the column family to which you want to add the columns. The following table describes the column properties that you configure when you associate columns with column families:

Property	Description
Name	Name of the column in the column family.
Type	Data type of the column.
Precision	Precision of the data.
Scale	Scale of the data.

Search and Add Column Properties

The following table describes the column properties that you configure when you search for columns in column families and add the required columns:

Property	Description
Maximum rows to sample	Maximum number of rows in the MapR-DB table that you want to include while searching for columns. Default is 100.
Column occurrence percent	The threshold occurrence percentage of the column. A column appears in the results when the occurrence percentage value of the column meets or exceeds the threshold value. Default is 90.

HBase Data Object Read Operation Properties for MapR-DB

HBase data object read operation properties include run-time properties that apply to the MapR-DB table you add in the HBase data object.

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

The following table describes the advanced property for an HBase data object read operation for MapR-DB:

Property	Description
Date Time Format	Format of the columns of the date data type. Specify the date and time formats by using any of the Java date and time pattern strings. Configure this property based on the format of the source data. If the source data contains year values, set the year part in the format as <code>yyyy</code> to avoid data corruption.

HBase Data Object Write Operation Properties for MapR-DB

HBase data object write operation properties include run-time properties that apply to the MapR-DB table you add in the HBase data object.

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

The following table describes the advanced properties for an HBase data object write operation for MapR-DB:

Property	Description
Date Time Format	Format of the columns of the date data type. Specify the date and time formats by using any of the Java date and time pattern strings. Configure this property based on the format of the target data. If the target data contains year values, set the year part in the format as <code>yyyy</code> to avoid data corruption.
Auto Flush	This property is not applicable for MapR-DB.

Partitioning

You can configure partitioning for MapR-DB mappings that you run in the native environment or on the Blaze and Spark engines in the Hadoop environment. When a mapping that is enabled for partitioning contains a MapR-DB table in an HBase data object operation as a **Read** transformation or **Write** transformation, the Data Integration Service can use multiple threads to read and write data.

Some transformations do not support partitioning. When a mapping enabled for partitioning contains a transformation that does not support partitioning, the Data Integration Service uses one thread to run the transformation. The Data Integration Service can use multiple threads to run the remaining mapping pipeline stages.

You can configure dynamic partitioning for HBase data objects that contain a MapR-DB table. When you configure dynamic partitioning, the Data Integration Service determines the number of partitions to create at run time.

To configure dynamic partitioning, open the HBase data object read or write operation that contains the MapR-DB table, click the **Run-time** tab, and select **Dynamic** from the **Partition Type** list.

After you configure dynamic partitioning, you can configure the maximum parallelism.

Configuring Maximum Parallelism for MapR-DB Mappings

Configure maximum parallelism in one of the following ways for MapR-DB mappings that you want to run in the native environment or on the Blaze and Spark engines in the Hadoop environment:

Set a maximum parallelism value for a mapping in the Developer tool.

By default, the **Maximum Parallelism** property for each mapping is set to **Auto**. Each mapping uses the maximum parallelism value defined for the Data Integration Service.

You can change the maximum parallelism value in the mapping run-time properties to define a maximum value for a particular mapping. When you set the maximum parallelism to different integer values for the Data Integration Service and the mapping, the Data Integration Service uses the minimum value of the two.

Ask the administrators to set the maximum parallelism for the Data Integration Service to a value greater than 1 in the Administrator tool.

Maximum parallelism determines the maximum number of parallel threads that process a single pipeline stage. Administrators can increase the **Maximum Parallelism** property value based on the number of CPUs available on the nodes where mappings run.

Parameterization of HBase Data Objects for MapR-DB

You can parameterize the HBase connection and the HBase data object operation properties for MapR-DB.

You can parameterize the following data object read operation properties for HBase data objects:

- Connection in the run-time properties
- Filter condition in the query properties
- Date time format in the advanced properties

You can parameterize the following data object write operation properties for HBase data objects:

- Connection in the run-time properties
- Date time format in the advanced properties

Creating an HBase Data Object for MapR-DB

Create an HBase data object to specify the MapR-DB table that you want to access to read or write data.

1. Select a project or folder in the **Object Explorer** view.
2. Click **File > New > Data Object**.
3. Select **HBase Data Object** and click **Next**.
The **New HBase Data Object** dialog box appears.
4. Enter a name for the data object.
5. Click **Browse** next to the **Location** option and select the target project or folder.
6. Click **Browse** next to the **Connection** option and select an HBase connection from which you want to import the MapR-DB table.
7. From the **Available OS Profiles** list, select an operating system profile. You can use the **Available OS Profiles** to increase security and to isolate the design-time user environment when you import and preview metadata from a Hadoop cluster.

Note: The Developer tool displays the **Available OS Profiles** list only if the Metadata Access Service is enabled to use operating system profiles. The Metadata Access Service imports the metadata with the default operating system profile assigned to the user. You can change the operating system profile from the list of available operating system profiles.

8. To add a MapR-DB table to the data object, click **Browse** next to the **Resource** option.
If a default Metadata Access Service is not set, a message appears to configure the Metadata Access Service. Click **OK** and set one Metadata Access Service as default. After you set a default Metadata Access Service, the **Add sources to the data object** dialog box appears. If the Metadata Access Service does not exist, contact the Informatica administrator to create a new Metadata Access Service.
9. Navigate or search for the MapR-DB table that you want to add to the data object and click **OK**.
You can add one MapR-DB table to the data object.
10. Click **Next**. The **Column Families** dialog box appears.
11. Select a column family and specify the columns in it. Choose to add columns or get all columns.

- To manually add, or search and add columns to the column family, select the **Add Columns** option.
 - To read from or write all columns in the column family to a single binary column, select the **Get all columns** option.
12. Add the columns in the column family. Choose to add columns or search the column names in the column family and add the columns.
 - To specify the columns from the column family when you know the column name and data type, select the column family to which you want to add the columns and click **Add**. Configure the add properties.
 - To search columns in the column family and add them, click **Search and Add**. The **Search and Add** dialog box appears.
 13. Specify the following details in the **Search and Add** dialog box:
 - a. Specify the maximum rows in the MapR-DB table that you want to include in the search.
 - b. Specify the threshold value of the column occurrence percentage.
 - c. Click **Go**.
The column name and the occurrence percentage of the column in the table appear in the results.
 - d. Select the columns that you want to specify for the column family. Configure the add properties.
 14. Click **Next**.
The **Create Row** dialog box appears.
 15. Select the **Include Row ID** option to generate a row ID for the MapR-DB table.
 16. Specify the datatype, precision, and scale for the row ID and click **Next**.
The **Review Columns** dialog box appears. The column family name is the prefix for all the column names in that column family for unique identification. The default data type of the row ID is String.
 17. Review the columns in the column families and click **Finish**.
The data object appears under Data Objects in the project or folder in the **Object Explorer** view.

Creating an HBase Data Object Operation for MapR-DB

Create an HBase data object operation from an HBase data object that contains a MapR-DB table.

Before you create an HBase data object operation for MapR-DB, you must create an HBase data object with the MapR-DB table.

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

7. Click **Finish**.

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

CHAPTER 4

MapR-DB Mappings

This chapter includes the following topics:

- [MapR-DB Mappings Overview, 19](#)
- [Filtering MapR-DB Source Data, 19](#)

MapR-DB Mappings Overview

After you create an HBase data object operation, you can add the operation to a mapping to read data from or write data to MapR-DB.

To read data from a MapR-DB table, add an HBase data object read operation as a **Read** transformation in the mapping. To write data to a MapR-DB table, add an HBase data object write operation as a **Write** transformation in the mapping.

When you configure the data object columns, you can get data in all columns in a column family to a single column as binary data. Use the Data Processor transformation to convert the binary data into the required data types.

You can validate and run mappings in the native environment or Hadoop environment. When you run a mapping in the native environment, the Data Integration Service runs the mapping from the Developer tool. When you run a mapping on a Hadoop cluster, you can select the Blaze or Spark engine. The Data Integration Service pushes the mapping to the selected engine for processing.

You can deploy the mapping and run it in the selected environment. You can run standalone mappings or mappings that are part of a workflow.

Filtering MapR-DB Source Data

When you configure a mapping that reads data from a MapR-DB source, you can enter a filter expression to filter records read from the source.

You can select the mapping and add the filter expression in the **Query** tab in the **Properties** view. You can use any comparison operator in the filter expression. When you run the mapping, the Data Integration Service filters the source data based on the expressions.

Note: If you use the not equal, less than, or less than or equal to operators, and some columns do not meet the filter condition, the Data Integration Service returns a Null value for these columns. If you use the equal,

greater than, or greater than or equal to operators, and some columns do not meet the filter condition, the Data Integration Service does not return the rows associated with these columns.

Example

The following table lists the columns in the CF column family in a MapR-DB table. There are rows that have c1 and c2 columns, rows that have at least one of the columns, and rows that have neither of the columns.

Row	Column Value
1	column=CF__c1, value=john
1	column=CF__c2, value=jane
2	column=CF__c1, value=jane
3	column=CF__c2, value=jdoe
4	column=CF__c8, value=adam

Create an HBase data object called Name and add it to a mapping. Add the following filter expression:

```
Employee.Name__C1 != 'john' AND Employee.Name__C2 != 'jane'
```

The Data Integration Service returns rows that contain null values because the filter expression contains the not equal to operator.

The Data Integration Service returns the following output:

```
Row 2  
c1: jane
```

```
Row 3  
c2: jdoe
```

```
ROW: 4  
c1: null  
c2: null
```

APPENDIX A

HBase Data Types Reference

This appendix includes the following topics:

- [HBase Data Type Reference Overview, 21](#)
- [HBase and Transformation Data Types, 22](#)

HBase Data Type Reference Overview

When you import a MapR-DB table, Informatica Developer displays HBase data types because PowerExchange for MapR-DB uses the HBase API to connect to MapR-DB tables.

Informatica Developer uses the following data types in MapR-DB mappings:

- HBase native data types. HBase data types appear in the physical data object column properties.
Note: Although data is stored in binary format in MapR-DB tables, you can specify the data type associated with a column when you create the HBase data object.
- Transformation data types. Set of data types that appear in the transformations. They are internal data types based on ANSI SQL-92 generic data types, which the Data Integration Service uses to move data across platforms. Transformation data types appear in all transformations in a mapping.

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

HBase and Transformation Data Types

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

HBase Data Type	Transformation Data Type	Range and Description
Binary	Binary	1 to 104,857,600 bytes. You can read and write data of the Binary data type. You can use the user-defined functions to transform the binary data.
DateTime	Date/Time	Date and time formats are specified by using any of the Java date and time pattern strings. For example, The "EEE, d MMM yyyy HH:mm:ss z" pattern string is interpreted as Wed, 14 Dec 2013 12:08:56. The "yyyy.MM.dd HH:mm:ss z" pattern string is interpreted as 2013.12.14 12:08:56 PDT.
Double	Double	Precision 15
Float	Decimal	Precision 1 to 28
Integer	Integer	-2,147,483,648 to 2,147,483,647
Long	Bigint	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
Short	Integer	-2,147,483,648 to 2,147,483,647
String	String	1 to 104,857,600 characters

INDEX

C

columns

- add [12](#)
- get all [12](#)
- search and add [12](#)

creating

- HBase connections for MapR-DB [10](#)
- HBase data object for MapR-DB [16](#)
- HBase data object operation for MapR-DB [17](#)

D

data object

- column configuration [12](#)

data object column configuration

- add columns [12](#)
- get all columns [12](#)
- search and add columns [12](#)

F

filtering source data

- run-time processing [19](#)

H

HBase connections

- MapR-DB properties [9](#)

HBase connections for MapR-DB

- creating [10](#)
- overview [9](#)

HBase data object for MapR-DB

- add columns [12](#)
- creating [16](#)

HBase data object operation

- read properties for MapR-DB [14](#)
- write properties for MapR-DB [14](#)

HBase data object operation for MapR-DB

- creating [17](#)

HBase data objects for MapR-DB

- add column properties [13](#)

HBase data objects for MapR-DB (*continued*)

- general properties [13](#)
- get all columns [12](#)
- overview [11](#)
- search and add column properties [13](#)
- search and add columns [12](#)

M

MapR-DB data object

- advance properties [19](#)

MapR-DB mappings

- run-time environment [19](#)
- validation [19](#)

O

overview

- HBase connections for MapR-DB [9](#)
- HBase data objects for MapR-DB [11](#)
- MapR-DB mappings [19](#)
- PowerExchange for MapR-DB [7](#)

P

PowerExchange for MapR-DB

- configuration overview [8](#)
- implementation [7](#)
- overview [7](#)
- prerequisites [8](#)

PowerExchange for MapR-DB data types

- mapping to transformation data types [22](#)
- overview [21](#)

R

run-time processing

- filtering MapR-DB source data [19](#)

run-time properties for MapR-DB

- HBase data object read operation [14](#)
- HBase data object write operation [14](#)