



# How to connect to MongoDB from Denodo

Revision 20210301

## NOTE

This document is confidential and proprietary of **Denodo Technologies**.  
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2024  
Denodo Technologies Proprietary and Confidential

## Goal

This document describes how to access MongoDB from the Denodo Platform.

## Content

MongoDB is an open source NoSQL database that uses a document-oriented database model which supports various forms of data. Instead of using tables and rows as in relational databases, the MongoDB architecture is made up of collections and documents. It provides high performance, high availability, and automatic scaling

The Denodo MongoDB Custom Wrapper available at the Denodo Support Site allows users to connect to any mongoDB database using the collections as a data source for Virtual DataPort.

## Connecting to MongoDB from the Denodo Platform

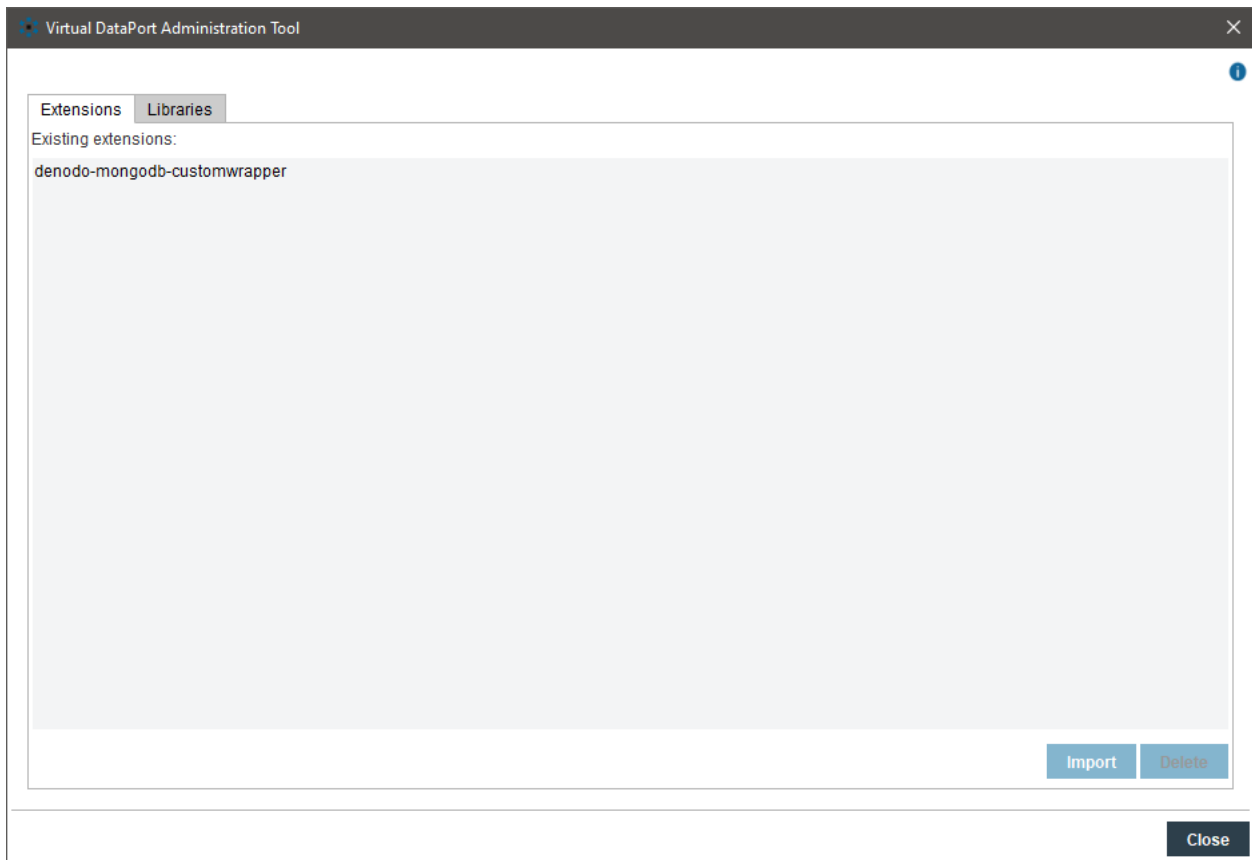
### Importing the custom wrapper into Virtual DataPort

The Denodo MongoDB Custom Wrapper component is available to download for Denodo support users from the “Downloads > Denodo Connects” section of the [Denodo Support Site](#).

In order to use the Denodo MongoDB Custom Wrapper in Virtual DataPort, configure the Virtual DataPort Administration Tool to import the extension.

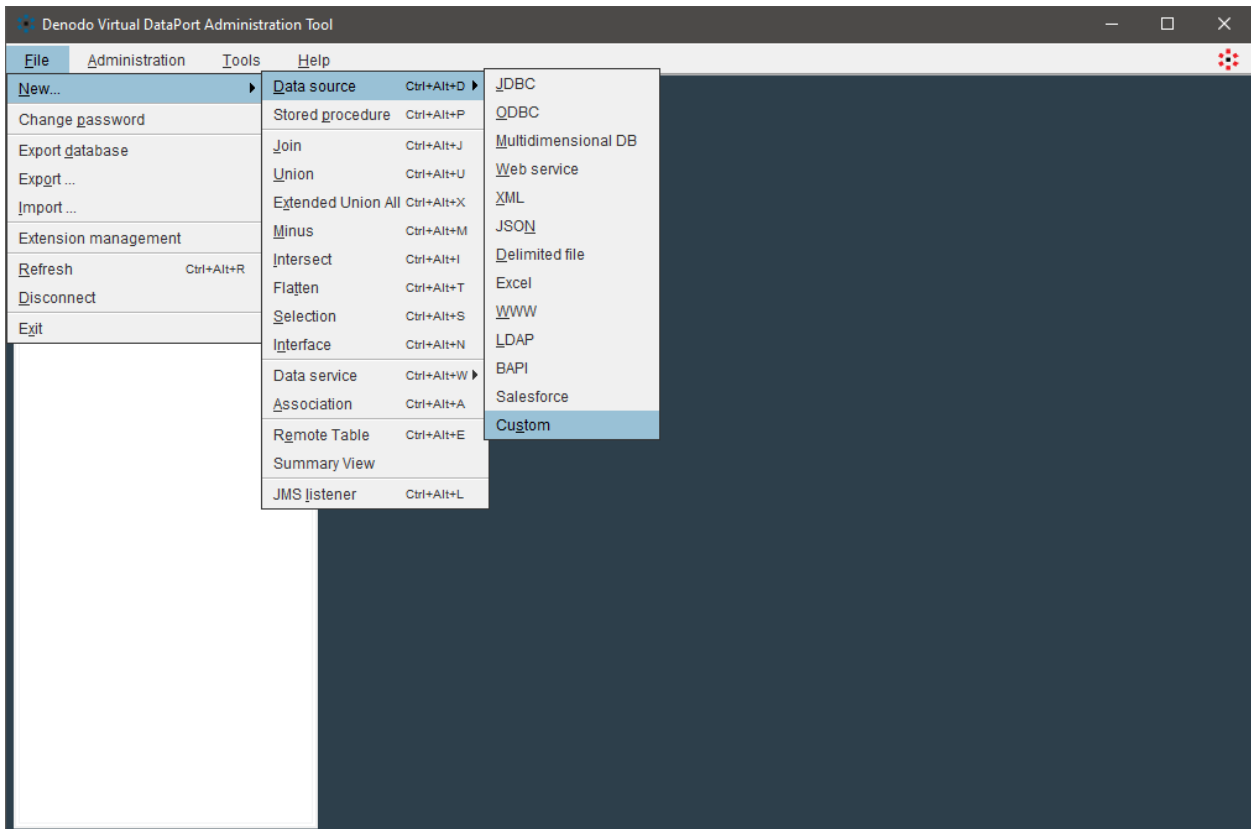
From the denodo-mongodb-customwrapper distribution, select the denodomongodb-customwrapper- $\{version\}$ -jar-with-dependencies.jar file and upload it to Virtual DataPort. To do so, from the Virtual DataPart Administration tool:

Go to “File > Extension management” and create a new item selecting the jar file.

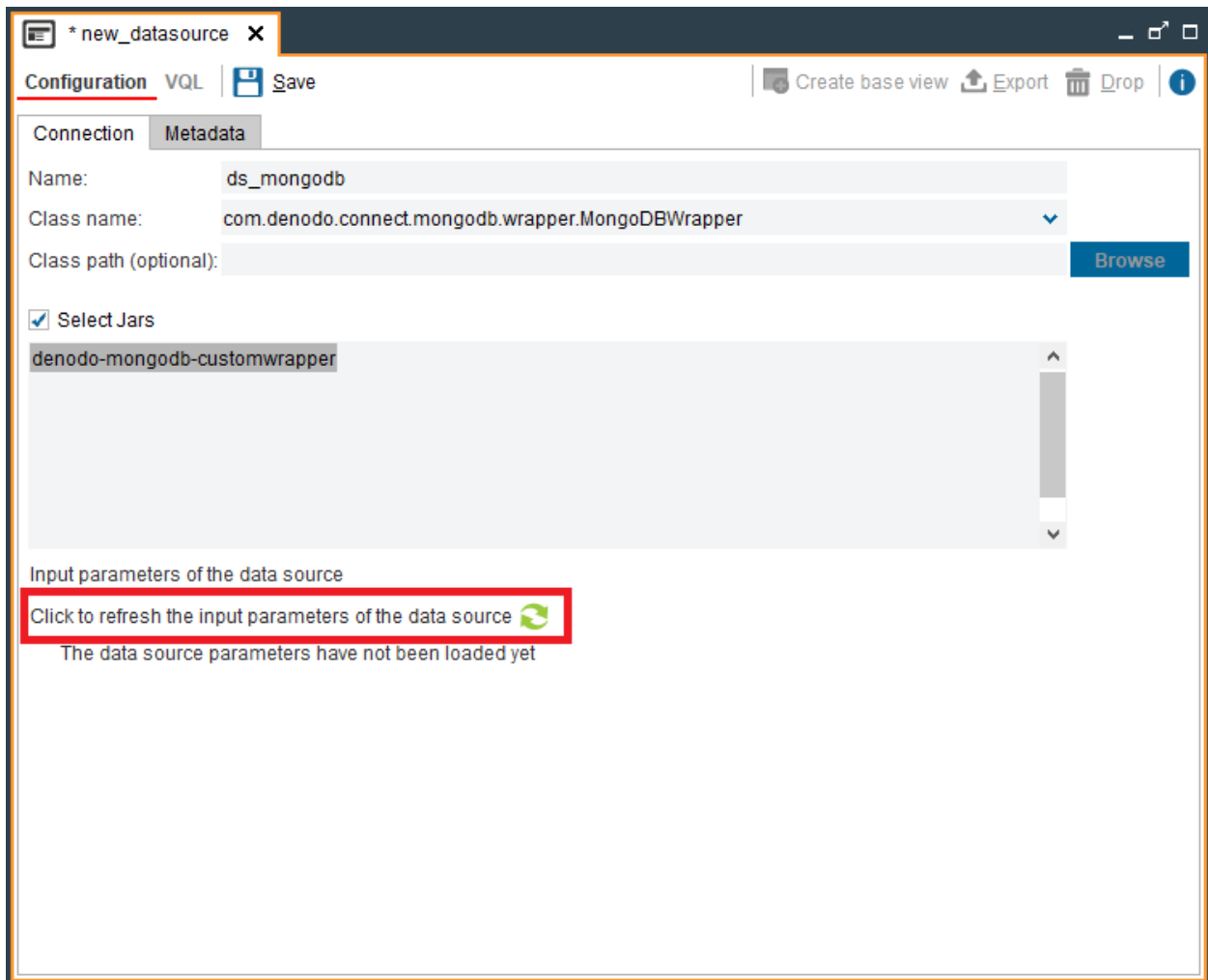


## Creating a Data Source

- From the Virtual DataPort Administration tool, create a new data source by selecting "File > New > Data source > Custom". This will open the wizard to create a connection to a data source with a custom wrapper.



- Specify the wrapper's Class name as `com.denodo.connect.mongodb.wrapper.MongoDBWrapper`. Also check 'Select Jars' and select the jar file of the custom wrapper.



- Click on the button to refresh the input parameters of the data source.
- Specify the input parameters of the data source:
  - **Host:** Name of the computer or IP address where MongoDB is running, default is 127.0.0.1.
  - **Port:** Port number to connect to MongoDB, default is 27017.
  - **Pass-through session credentials:** Enabling this checkbox allows Virtual DataPort to execute the queries using the credentials of the user logged into Virtual DataPort, so User and Password fields would be only used for introspection.
  - **User/Password:** Username and password to connect to MongoDB, if the authentication is enabled.
  - **Database:** Database name where the data is located (if you want to use a different database for authentication please check the “Authenticate against a different database” section).
  - **Connection String:** this parameter allows more options in the connection between the driver and mongodb. This is the pattern:

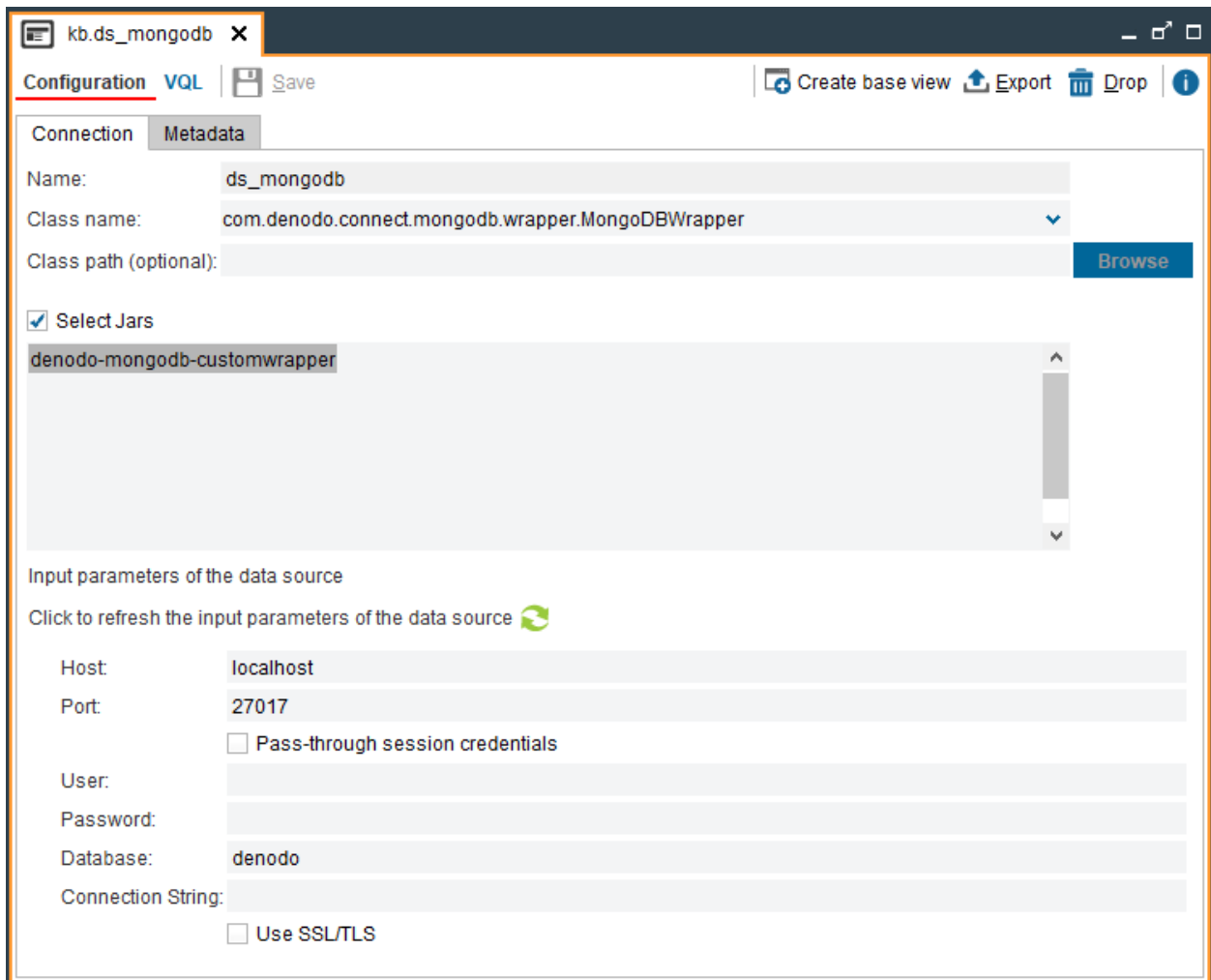
[mongodb://]host1[:port1][,host2[:port2],...[,hostN[:portN]]]/database[?]

Options]

The prefix `mongodb://` is optional, the database is mandatory and user and password are not written in this field, but in the User and Password Parameters.

Make a note that, if you introduce this parameter, Host, Port and Database parameters should be empty.

- **Use SSL/TLS:** Enabling this checkbox will make the wrapper use SSL/TLS. When using a Connection String, this can also be specified by means of the `"ssl=true"` option.




- Click on "Save".

### Authenticate against a different database

In MongoDB it is possible to authenticate against a database but read the data from a different one. When the authentication is done in a different database, you have to add the option `authsource` in the Connection String parameter and leave the parameters Host, Port and Database empty as in the screenshot below.

Input parameters of the data source

Click to refresh the input parameters of the data source 

Host:

Port:

Pass-through session credentials

User:

Password:

Database:

Connection String:

Use SSL/TLS

More details about *authsource* can be found in the Connection Options of the Connection String URI [here](#).

### Creating a Base view

Once the data source is created, create base views for that particular source. In order to do that click on the "Create Base View" option.

- Specify the following to create a base view.
  - **Collection:** Collection name that we will import as a table in Virtual DataPort.

**There are also two parameters that are mutually exclusive:**

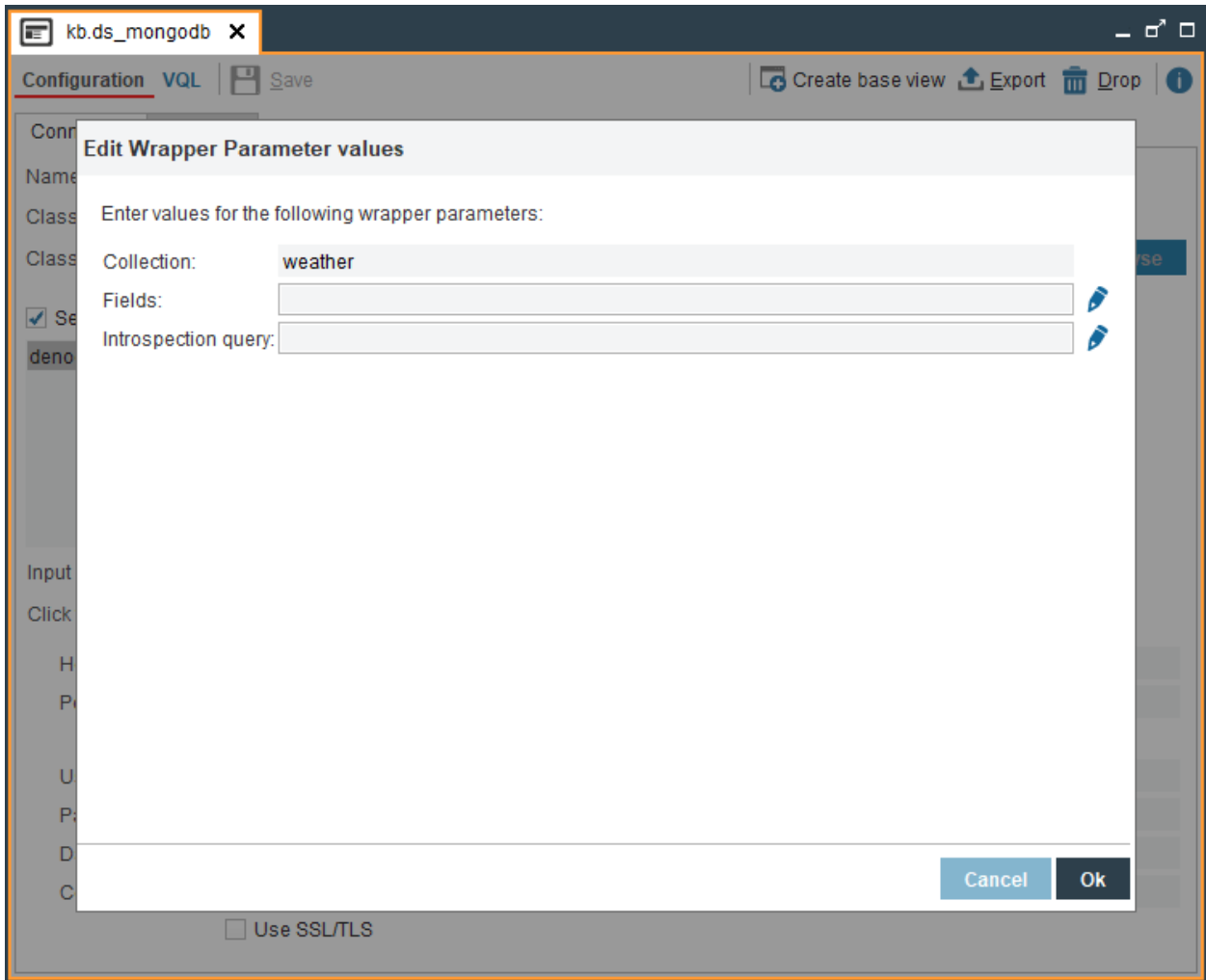
- **Fields:** The fields we would like to import as columns in VDP. We must keep the syntax.

field1[:type1][,field2[:type2],...].

Type, when specified, should be one of the constants in `java.sql.Types` (note these are SQL standard types).

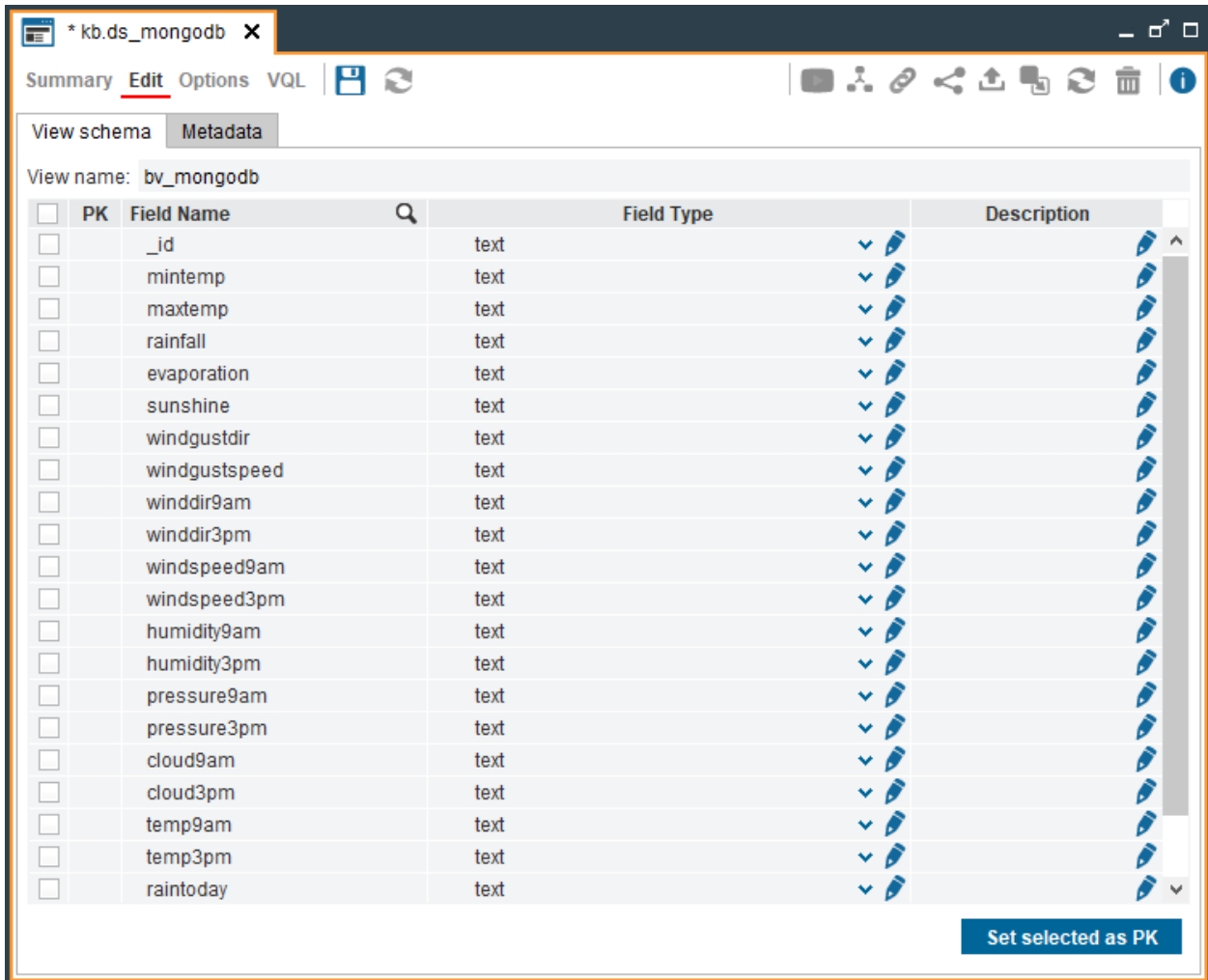
- **Introspection query:** Documents retrieved by this query will be analyzed to reveal their fields and build the view schema. An empty query selects all documents in the collection, so, if the collection has a lot of documents, the creation of the base view can take a long time. In addition, it is recommendable to check the types of the new view inferred from the introspection. This query requires MongoDB syntax.

The below image is an example of creating a base view over the MongoDB datasource.



- Click 'Ok' to create a base view.





- Now, the base view is ready for the execution and to be combined with the rest of the sources.

## References

Virtual DataPort Administration Guide: [Custom Data Sources](#)  
Denodo MongoDB Custom Wrapper - [User Manual](#)