Documentation – MongoDB + Tableau

Data Warehouses and Analytical Processing, Poznan University of Technology

Introduction

As a project completed during Data Warehouses and Analytical Processing course Roman Kaczorowski's and Jan Wielebinski's task was to find and test a connector which would allow to connect non relational MongoDB database with Tableau data visualization tool. Project's supervisors were Renata Kowalicka and Przemysław Szymil from Roche company.

As the result of the project group has:

- prepared datasets in MongoDB database,
- established connection between MongoDB and Tableau,
- prepared reports which use aggregation, filtering and calculation within Tableau,
- provided files and scripts necessary to recreate work done in the project,
- created documentation,
- prepared presentation.

MongoDB

MongoDB is an document-oriented database written in C++. It is classified as a NoSQL database, because it stores data in collections instead of tables. MongoDB uses dynamic schemas. There is no need of defining the structure, all you have to do is give the name and value of field. You can change structure of documents simply by adding new fields or deleting existing ones. Documents in a collection need unique set of fields. The data model looks like JSON objects. MongoDB doesn't support joins or transactions, however it provides atomic operations on a single document. Usually these document-level atomic operations are enough to solve problems that would require ACID transactions in a relational database.

MongoDB is developed by MongoDB Inc. and is free and open-source, published under a combination of the GNU Affero General Public License and the Apache License. Development began in October 2007 and the first public release was in February 2009. MongoDB can be downloaded from http://www.mongodb.org/downloads. So far it supports following platforms:

- OS X
- Linux
- Windows
- Solaris

Simba MongoDB ODBC Windows driver

Since MongoDB is a schema-less data source, the Simba connector defines schemas on the fly providing a low touch SQL window to MongoDB data. Further schema refinements are possible via the Schema Editor. High performance data access is achieved through Simba SQL Engine's built-in Collaborative Query Execution which ensures that filtering and aggregation workflows are passed directly to MongoDB rather than being processed within a client application or intermediary system.

The connector can be downloaded from Simba Technologies webpage at http://www.simba.com/drivers/mongodb-odbc-jdbc/. It supports all major platforms.

MongoDB (3.2.7) configuration for Windows

1. After installing MongoDB database, create *data* and *log* directories which will be used by the database.
2. Create *mongo.config* file containing below lines:

```
dbpath=<DATA DIRECTORY PATH>
logpath=<LOG DIRECTORY PATH>\mongo.log
diaglog=3
```

3. Start database server by the command line in *bin* directory of MongoDB by entering:

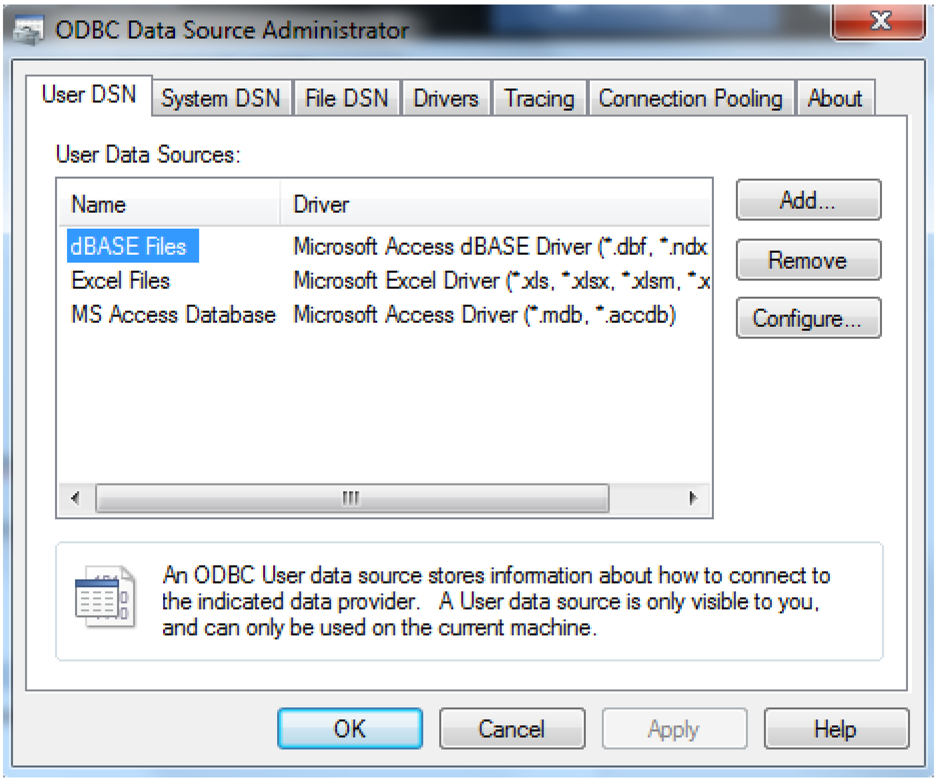
```
./mongod.exe --config="<CONFIG FILE PATH>"
```

4. Upload example file by the command line in *bin* directory of MongoDB by entering:

```
./mongoimport --db test --collection <NAME> --file <PATH>
```

Simba MongoDB ODBC Windows driver (2.0.1.1002) configuration for Windows

1. After installing Simba connector, please open *ODBC Administrator* via Windows start menu.

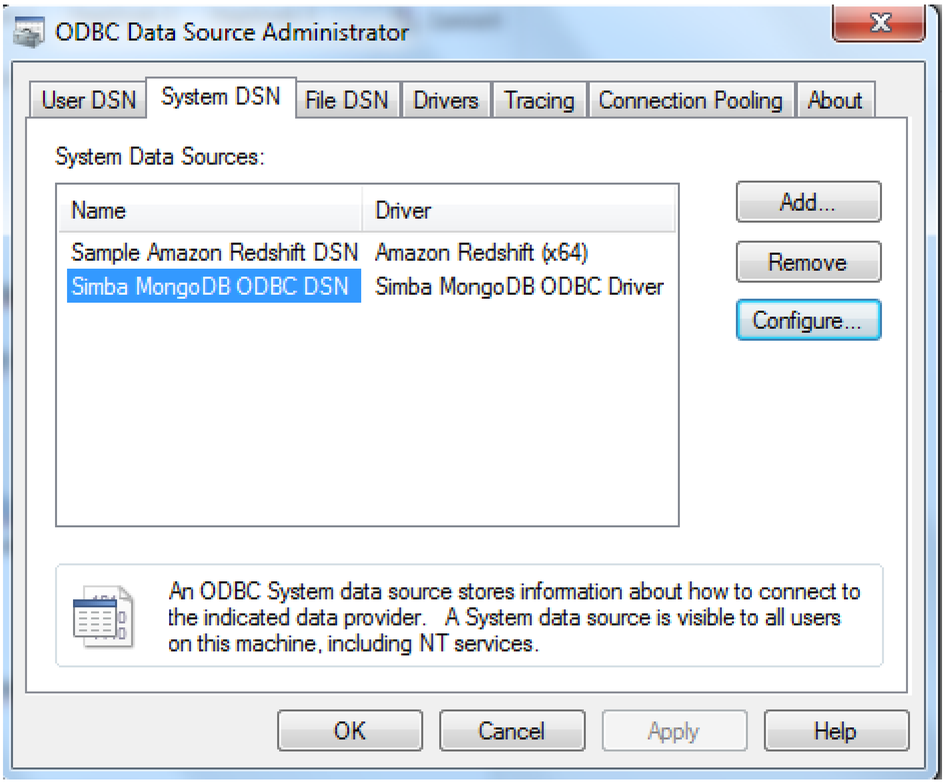


User Data Sources:

Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.x
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

2. Click on the *System DSN* tab.

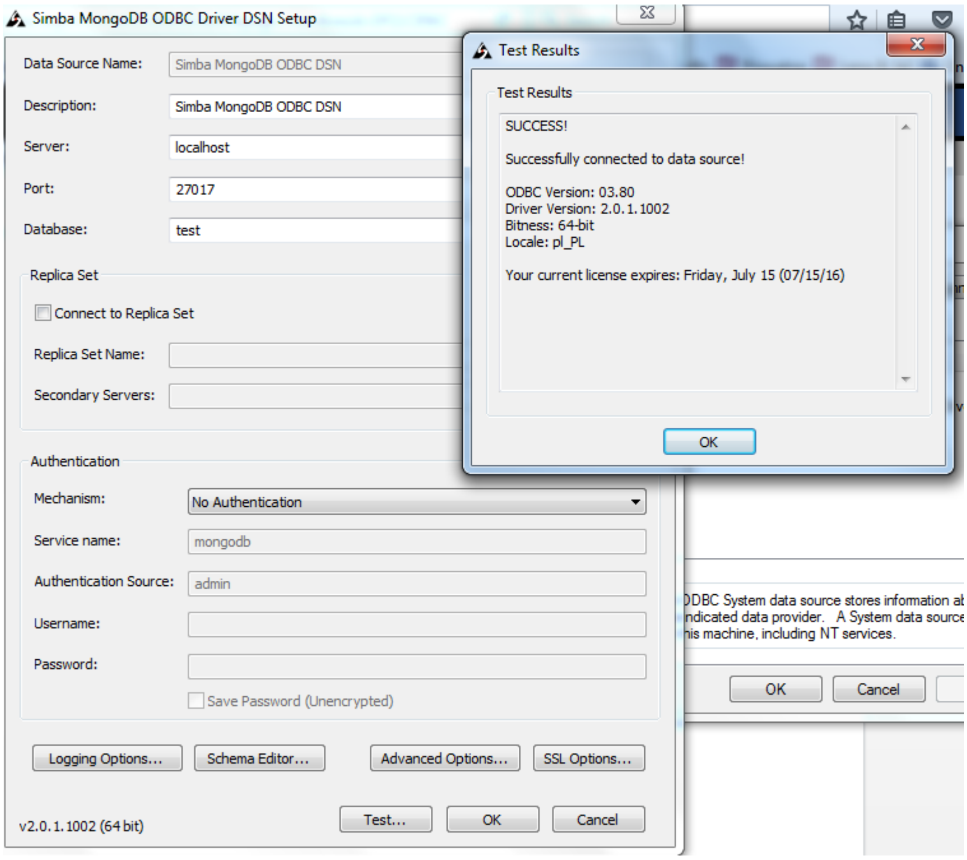


System Data Sources:

Name	Driver
Sample Amazon Redshift DSN	Amazon Redshift (x64)
Simba MongoDB ODBC DSN	Simba MongoDB ODBC Driver

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

3. Choose *Simba MongoDB DSN* and click on the *Configure* button.
4. In the Driver *DSN Setup* window, provide server address and name of the created MongoDB database.
5. Test your connection by clicking on the *Test...* button.



Data Source Name: Simba MongoDB ODBC DSN

Description: Simba MongoDB ODBC DSN

Server: localhost

Port: 27017

Database: test

□ Connect to Replica Set

Replica Set Name:

Secondary Servers:

Mechanism: No Authentication

Service name: mongodb

Authentication Source: admin

Username:

Password:

□ Save Password (Unencrypted)

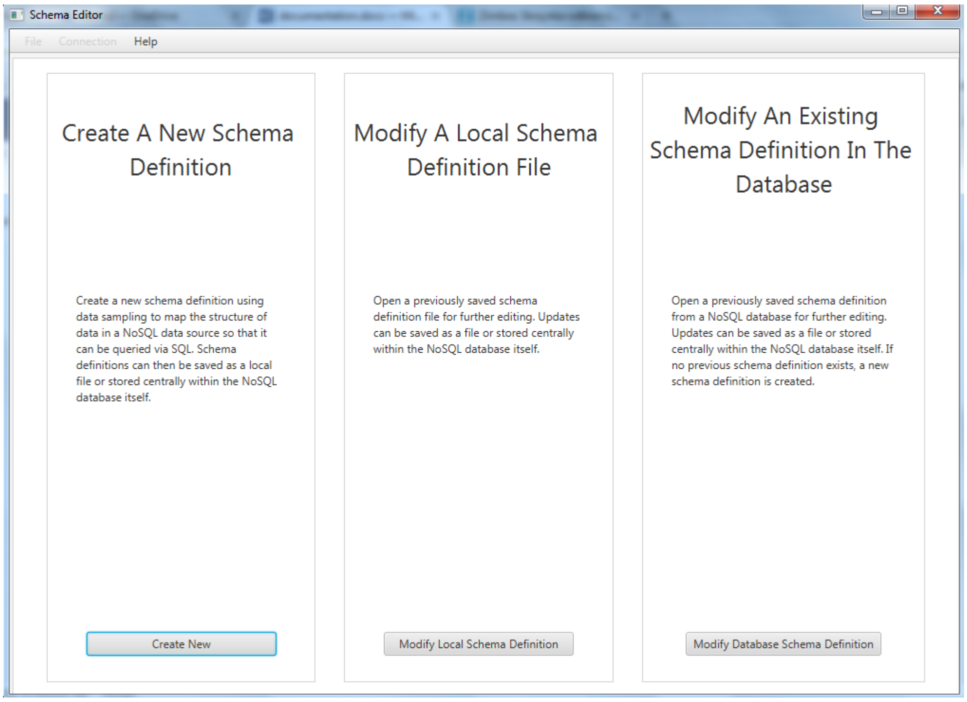
v2.0.1.1002 (64 bit)

Test Results

```
SUCCESS!

Successfully connected to data source!

ODBC Version: 03.80
Driver Version: 2.0.1.1002
Bitness: 64-bit
Locale: pl_PL

Your current license expires: Friday, July 15 (07/15/16)
```

Note: in case of the problem, please make sure that you have saved Simba product license in required directory. You should have received *.lic* file on the email address provided in the registration form when the connector was downloaded. After downloading the license file, it must be placed in the *lib* directory of the connector's installation folder.

6. In the *Driver DSN Setup* window, click on the *Schema editor* button. *Schema editor* window will appear.


Schema Editor
File
Connection
Help
Create New
Modify Local Schema Definition
Modify Database Schema Definition

Create A New Schema Definition

Create a new schema definition using data sampling to map the structure of data in a NoSQL data source so that it can be queried via SQL. Schema definitions can then be saved as a local file or stored centrally within the NoSQL database itself.

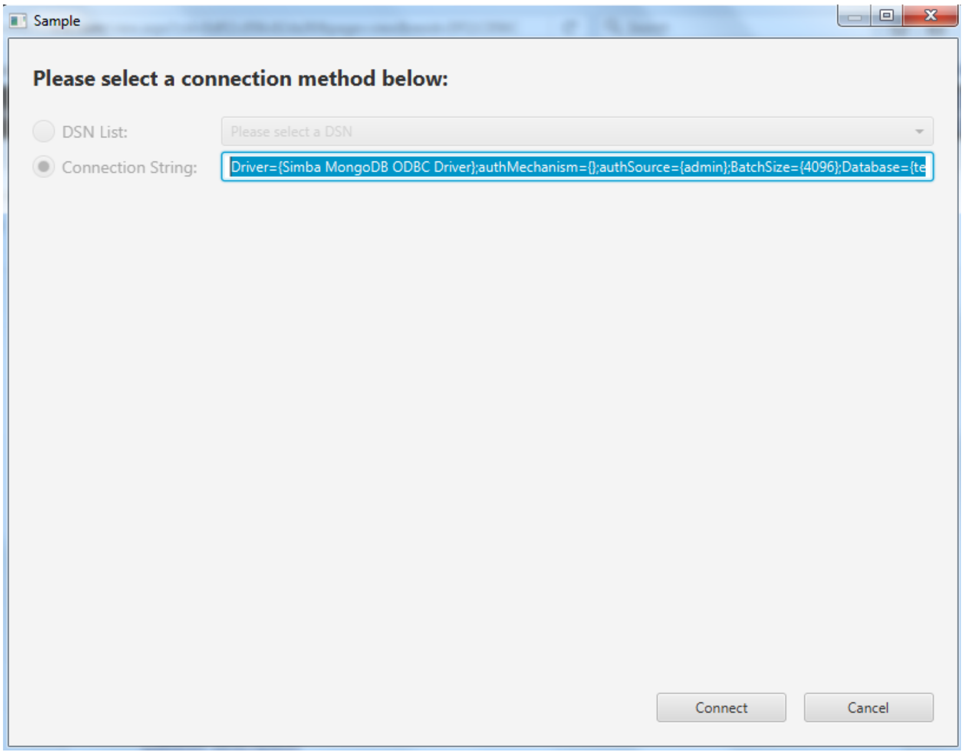
Modify A Local Schema Definition File

Open a previously saved schema definition file for further editing. Updates can be saved as a file or stored centrally within the NoSQL database itself.

Modify An Existing Schema Definition In The Database

Open a previously saved schema definition from a NoSQL database for further editing. Updates can be saved as a file or stored centrally within the NoSQL database itself. If no previous schema definition exists, a new schema definition is created.

7. Click on the *Create New* button to create new schema definition.
8. You can stick with default connection string and just click on the *Connect* button.

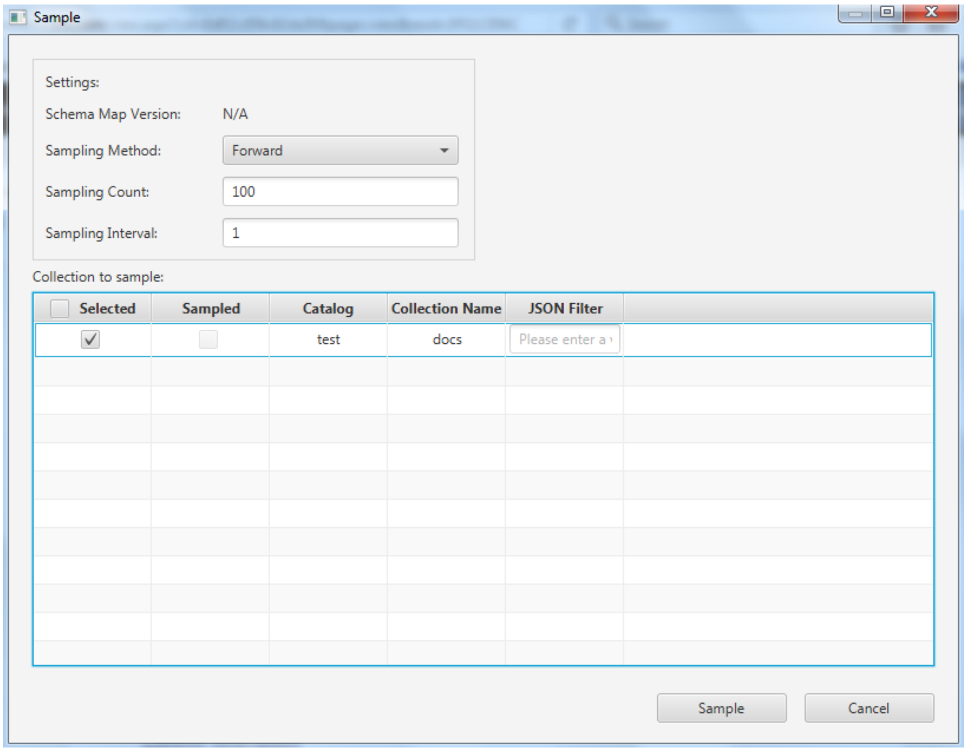


Please select a connection method below:

○ DSN List: Please select a DSN

● Connection String: Driver={Simba MongoDB ODBC Driver};authMechanism={};authSource={admin};BatchSize={4096};Database={te

9. Select proper collection, method and count of sampling and then click on the *Sample* button.



Sample

Settings:

Schema Map Version: N/A

Sampling Method: Forward

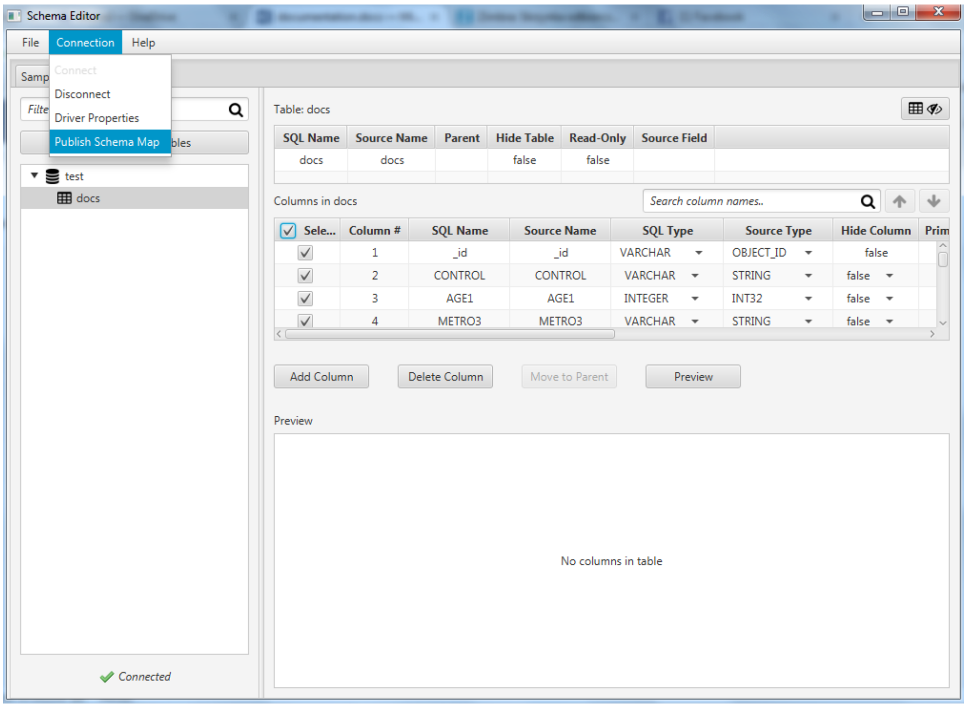
Sampling Count: 100

Sampling Interval: 1

Collection to sample:

□ Selected	Sampled	Catalog	Collection Name	JSON Filter
☑	□	test	docs	Please enter a

10. Prepare schema by selecting the columns that you want to analyze using *Add* or *Delete Column* buttons.
11. Click on the *Connection* menu in top bar, then choose *Publish Schema Map* option in order to save new schema on MongoDB server.



Schema Editor

Table: docs

SQL Name	Source Name	Parent	Hide Table	Read-Only	Source Field
docs	docs		false	false	

Columns in docs

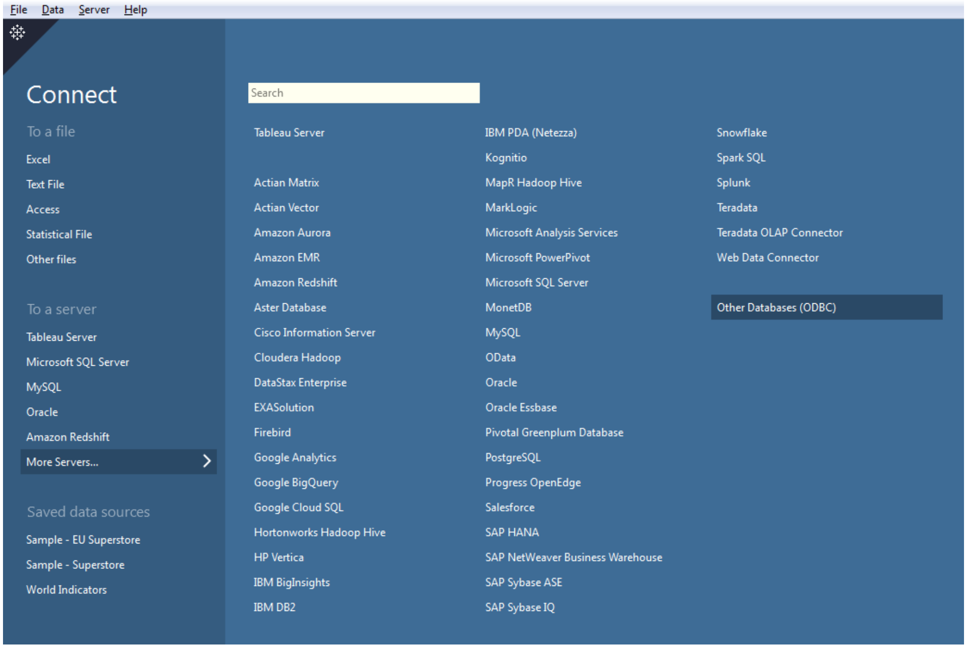
☑ Sele...	Column #	SQL Name	Source Name	SQL Type	Source Type	Hide Column	Prim
☑	1	_id	_id	VARCHAR	OBJECT_ID	false	
☑	2	CONTROL	CONTROL	VARCHAR	STRING	false	
☑	3	AGE1	AGE1	INTEGER	INT32	false	
☑	4	METRO3	METRO3	VARCHAR	STRING	false	

Preview

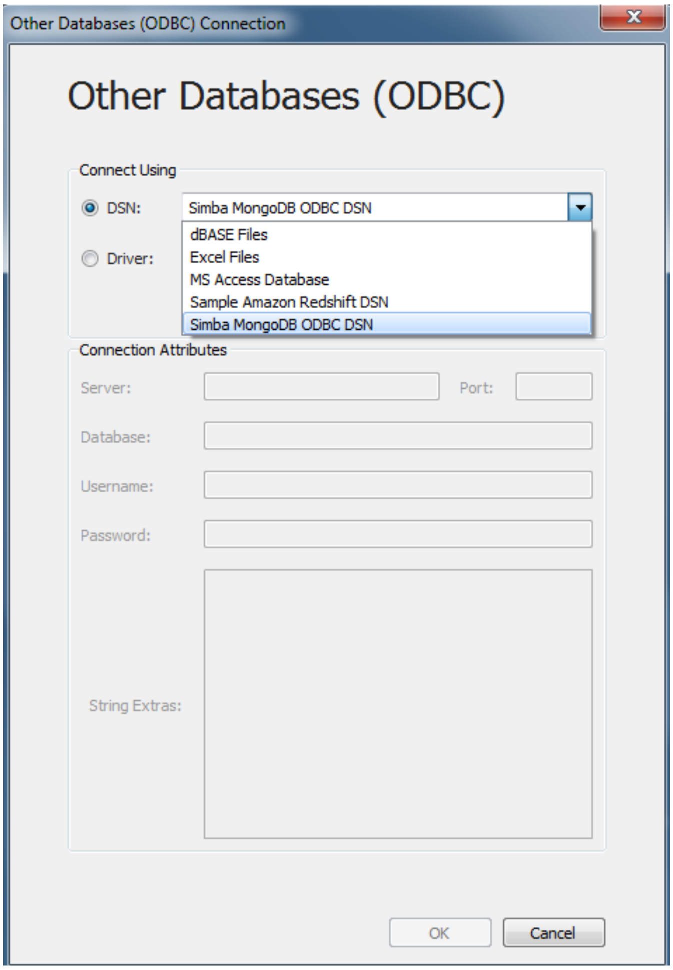
No columns in table

Tableau (9.3) configuration for Windows

1. After launching Tableau, select *More Servers…* option from *Connect to a server…* menu.
2. Select *Other Databases (ODBC)*.



3. In *Connect Using* field check *DSN* option and from the dropdown list select *Simba MongoDB ODBC DSN* and then click on *Connect* button. Finally, click on the OK button.


Other Databases (ODBC) Connection

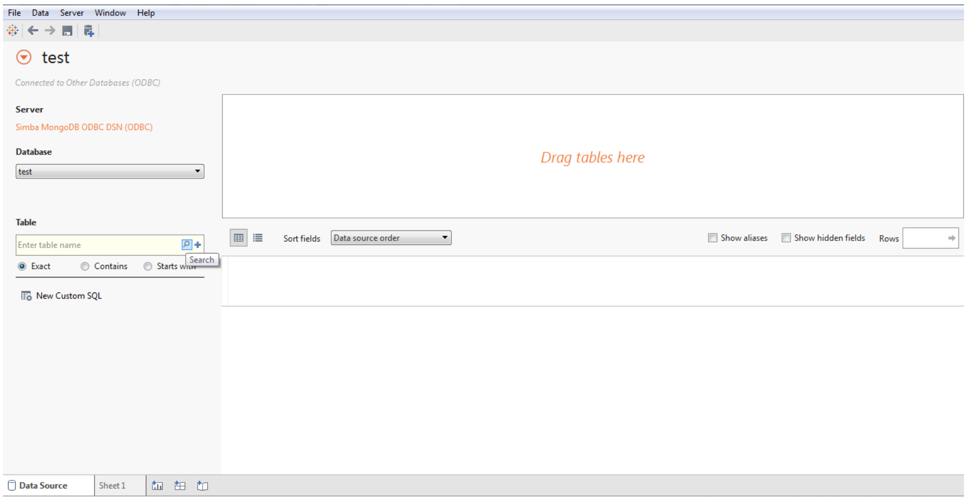
Other Databases (ODBC)

Connect Using

● DSN: Simba MongoDB ODBC DSN

○ Driver:

4. Select proper database, then click on the loupe icon to
5. Select required tables and enjoy building the report!



test

Connected to Other Databases (ODBC)

Server

Simba MongoDB ODBC DSN (ODBC)

Database

test

Table

Enter table name

● Exact ○ Contains ○ Starts with

Sort fields: Data source order

□ Show aliases □ Show hidden fields Rows

Dataset description

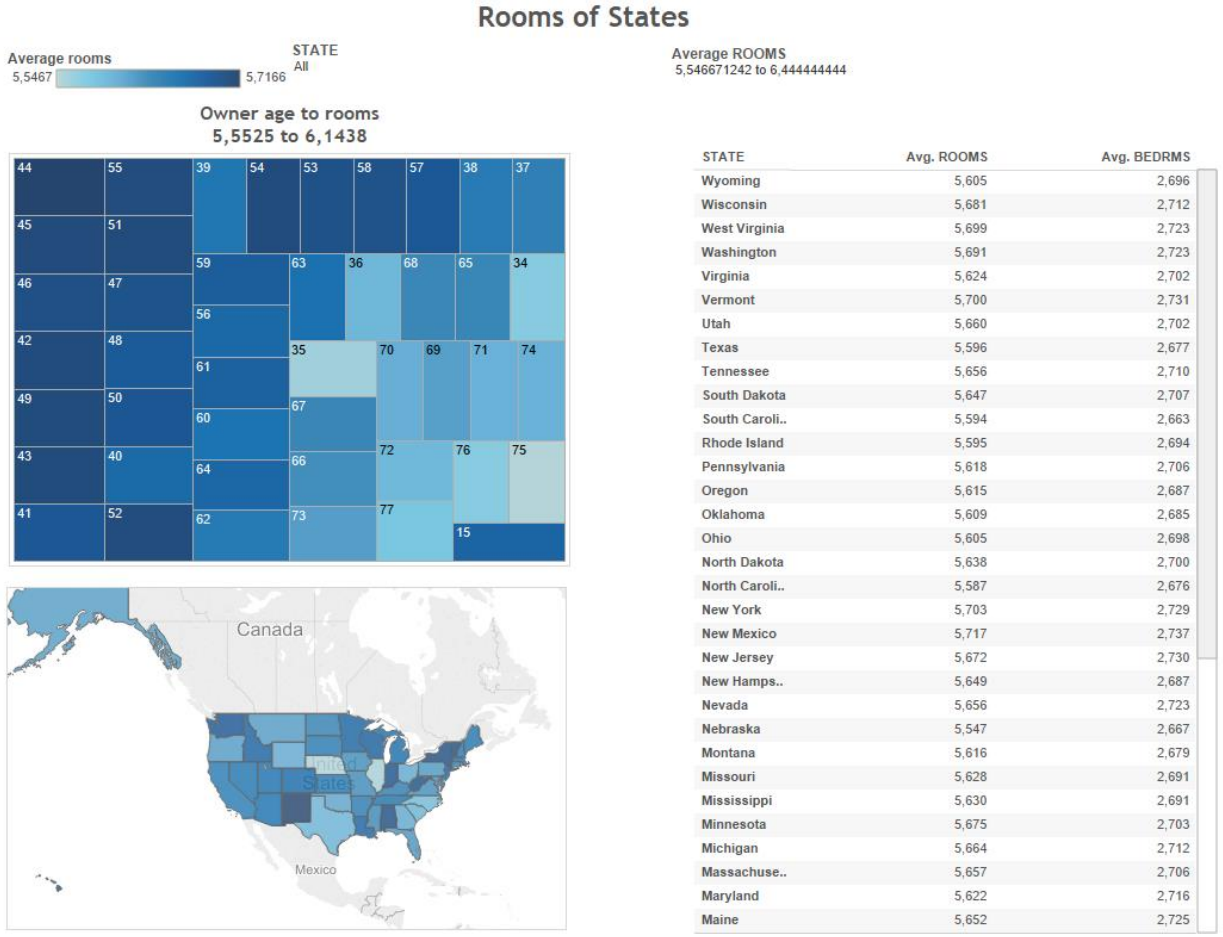
Dataset is called „Housing Affordability Data System (HADS)" and can be easily downloaded from https://catalog.data.gov/dataset/housing-affordability-data-system-hads . This set categorizes housing units by affordability, households by income, with respect to the Adjusted Median Income, Fair Market Rent (FMR) and poverty income. It also includes housing cost burden for owner and renter households.

The data files are available for public use, since they were derived from AHS public use files and the published income limits and FMRs. This dataset gives the community of housing analysts the opportunity to use a consistent set of affordability measures.

It has 100 attributes to describe property and contains 145532 rows.

Report description

Two dashboards were created to see how Tableau handles data from MongoDB. First is focused on rooms in house.



Rooms of States

Average rooms 5,5467 5,7166

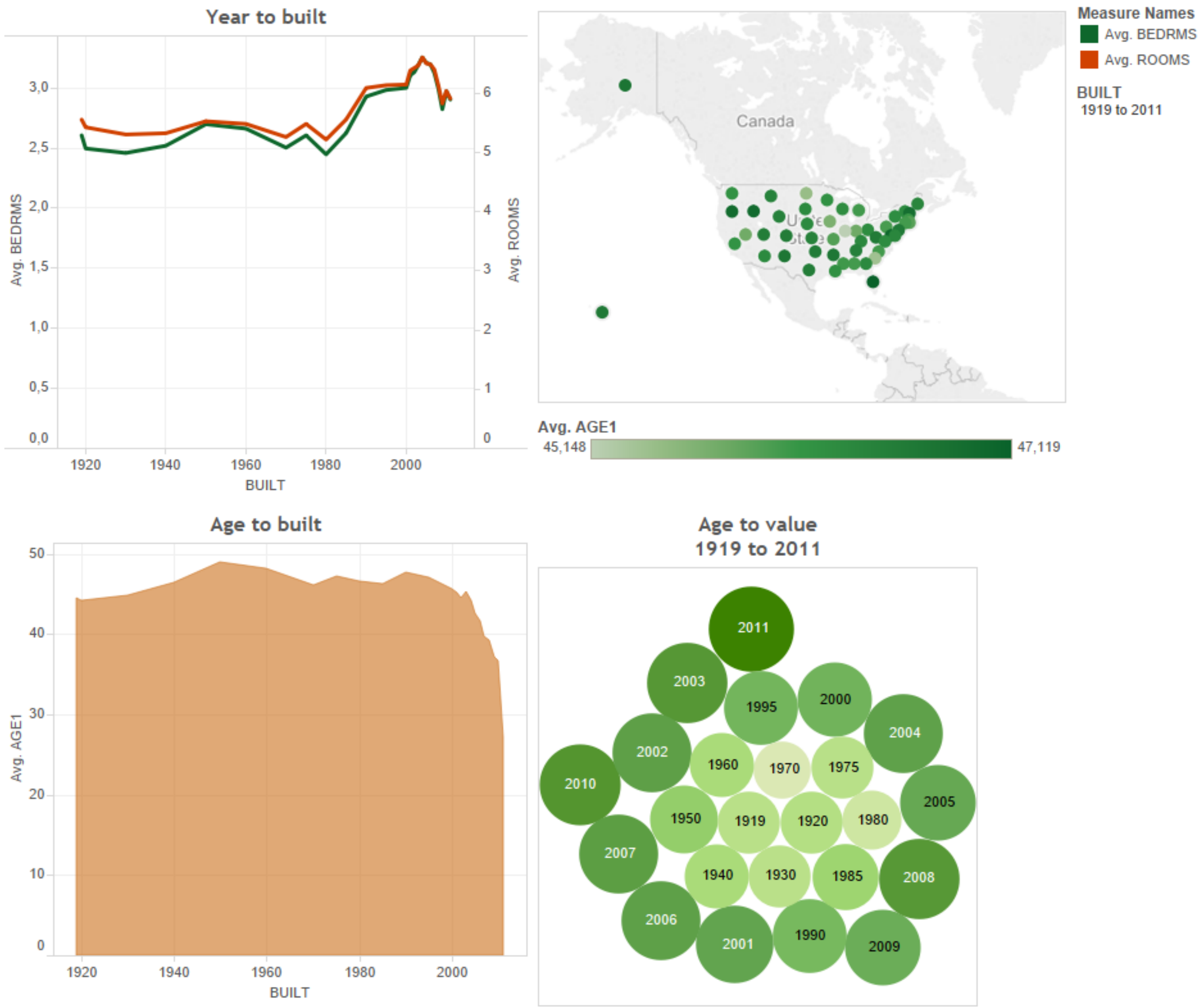
STATE All

Average ROOMS 5,546671242 to 6,444444444

Owner age to rooms 5,5525 to 6,1438

STATE	Avg. ROOMS	Avg. BEDRMS
Wyoming	5,605	2,696
Wisconsin	5,681	2,712
West Virginia	5,699	2,723
Washington	5,691	2,723
Virginia	5,624	2,702
Vermont	5,700	2,731
Utah	5,660	2,702
Texas	5,596	2,677
Tennessee	5,656	2,710
South Dakota	5,647	2,707
South Caroli..	5,594	2,663
Rhode Island	5,595	2,694
Pennsylvania	5,618	2,706
Oregon	5,615	2,687
Oklahoma	5,609	2,685
Ohio	5,605	2,698
North Dakota	5,638	2,700
North Caroli..	5,587	2,676
New York	5,703	2,729
New Mexico	5,717	2,737
New Jersey	5,672	2,730
New Hamps..	5,649	2,687
Nevada	5,656	2,723
Nebraska	5,547	2,667
Montana	5,616	2,679
Missouri	5,628	2,691
Mississippi	5,630	2,691
Minnesota	5,675	2,703
Michigan	5,664	2,712
Massachuse..	5,657	2,706
Maryland	5,622	2,716
Maine	5,652	2,725

The top left side is treemap chart showing relation between average number of bedrooms to the age of house owner. On the right side there is a list of all states with average bedrooms and rooms count. All dashboards components are related to each other and can be easily filtered by choosing states from dropdown menu or the average room amount can be selected by manipulating the slider. Charts are also interactive and can filter the data. For example if the state is being selected on the map, another diagram will show statistics related only to this state.



Second dashboard is focused on the year when the house was built. The first chart shows relation between number of rooms and year of construction. On the map, darker color corresponds to mature owners. The bubble chart is linked to property value. Lighter colors mean low average price. As in the previous dashboard, all components are related and filtering is also possible. It can be done by selecting the data on the charts or from the right side menu.