



# Creating a universe on Hive with Hortonworks HDP 2.0

Learn how to create an SAP BusinessObjects Universe on top of Apache Hive 2 using the Hortonworks HDP 2.0 distribution

**Author(s):** Ajay Singh (Hortonworks), JC Raveneau (SAP), Pierpaolo Vezzosi (SAP)  
**Company:** Hortonworks & SAP  
**Created on:** December 2013

## Contents

- 1 Introduction..... 2
  - 1.1 Applies to..... 2
  - 1.2 Summary ..... 3
  - 1.3 Audience & prerequisites ..... 3
  - 1.4 Structure of this document ..... 3
  - 1.5 Important note about support ..... 4
- 2 Finding and installing the Hortonworks software..... 4
  - 2.1 Find, install and start a Hortonworks HDP 2 server ..... 4
  - 2.2 Find, install and configure the Hortonworks ODBC middleware ..... 5
- 3 Creating a universe on top of Hortonworks Hive..... 8
  - 3.1 Creating the connection in IDT ..... 9
  - 3.2 Creating the data foundation in IDT ..... 10
  - 3.3 Creating the business layer in IDT ..... 14
  - 3.4 Publishing the universe ..... 18
- 4 Running a sample query ..... 19
- 5 Additional information..... 22

## 1 Introduction

Building on the strategy to be an open business intelligence platform capable of addressing most data sources, SAP BusinessObjects BI4 added the support for Apache Hive™ back in 2012 through the Apache Hive JDBC driver.

Since then, Apache™ Hadoop® became relevant as an enterprise ready big-data source thanks to the effort around commercial distributions such as Hortonworks Data Platform which also provide an ODBC driver for Hive.

In order to best leverage the latest innovations with your SAP BusinessObjects BI deployment, we offer here an option to leverage the capability of the platform to connect to any data source that offers an ODBC driver.

### 1.1 Why create a Universe on Hadoop ® to connect to Hive?

Hive was designed to be the data warehouse on Hadoop. As such it is a versatile and convenient way to get immediate value out of your Hadoop data with your existing SAP BusinessObjects BI4 platform and all BI clients that consume Universes (SAP Lumira included).

Once a Universe is created on Hive and published to the platform, users can consume it as any other Universe from any other data source.

While the default behavior of the Universe leverages the compatibility between SQL and the Hive Query Language (HQL), advanced Hive features can always be accessed via hand coding HQL in derived tables.

### 1.2 Best Practices for Universes on Hive

Hadoop is very good at storing and analyzing large volumes of data, thanks to HDFS™ and MapReduce. It is traditionally been used as a batch analytics and transformation platform, with query latency of over 30 seconds. As the usage and adoption of Hadoop has proliferated, enterprises are increasingly looking at their Hadoop infrastructure to support interactive queries. To this end, Hortonworks has made significant advancements and will be delivering an increasingly interactive user experience via Hive in the first half of 2014. More information and early access version of the offering can be found at <http://hortonworks.com/labs/stinger/>.

While the enhancements represent a key step forward for Hadoop, given the need to operate at petabyte scale the solution does not address highly concurrent (1000s) sub second response times. As such, Universe and BI report or BI Dashboard designers must understand the capabilities to best meet the user expectation. If high interactivity and / or concurrency is required, you should consider pairing Hadoop with SAP HANA: <http://www.sapbigdata.com/>

While creating a table in Hive, only relevant columns should be exposed. Files found in Hadoop will typically be raw and either include information irrelevant for the use case or empty columns. Limiting our Hive table to only relevant metadata is not expected to have a significant impact on performance but makes the process of creating the Universe easier.

You should also consider pre-processing the data where possible. As with any data warehouse, preparing a dataset through aggregations, cleansing or any other transforms will ensure this does not have to be done at query time.

The new ORC (optimized row column) file format delivered with Hive 0.11 is a key contributor to good performances. While creating your Hive tables it is recommended to use ORC as the backing file format. Details can be found here: [http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.0.2/ds\\_Hive/orcfile.html](http://docs.hortonworks.com/HDPDocuments/HDP2/HDP-2.0.0.2/ds_Hive/orcfile.html)

Experiment with scalability. Hadoop is in the growing stages as an enterprise data platform. Some SAP BusinessObjects BI deployments handle thousands of users with concurrency pushing into the 1000's. It's probably a good idea to limit the access to your Hadoop-based Universe to the smaller set of users that would benefit the most from it. In use cases where concurrency is required you should consider pairing Hadoop with SAP HANA.

Finally, where appropriate one should consider scheduling the reports. BI users should be educated to fully leverage the scheduling capabilities of the SAP BusinessObjects BI platform.

### 1.3 Applies to

SAP BusinessObjects BI 4.0, BI 4.1 and newer releases

Hortonworks Data Platform 2.0 with Hive 2

### 1.4 Summary

This document provides information on how to build an SAP BusinessObjects universe on top of the Hive 2 distribution of Hortonworks Data Platform 2.0 (HDP 2.) using the Hortonworks Hive ODBC driver

### 1.5 Audience & prerequisites

Readers of this document should be proficient in Universe design and in accessing Hive data stores.

We expect readers to have previously used the Information Design Tool to build universes and be comfortable with SQL and ODBC connectivity to a Hortonworks installation.

The document doesn't contain basic information on the usage of the client interfaces as we expect readers to be familiar with them.

To create the universe you are expected to have installed:

- The SAP BusinessObjects Information Design Tool (IDT)
- A Hortonworks Data Platform 2.0 system or Hortonworks Sandbox
- The Hortonworks Hive ODBC driver (on the machine where IDT is installed)

To query the universe you are expected to have installed:

- The Web Intelligence Rich Client (on the machine where IDT is installed, for local queries)
- The SAP BusinessObjects BI platform (on a server where all the client tools can connect to retrieve the universe and run the query)

### 1.6 Structure of this document

In this document we present a typical workflow which can be followed to create from scratch a universe on Hortonworks Data Platform including Hive and run a sample query on it with SAP BusinessObjects Web

Intelligence Rich client. We also provide information on how to run queries with the other clients available in SAP BusinessObjects BI.

In the typical workflow you are required to take the following steps:

1. Install and run an Hortonworks Data Platform 2.0 server
2. Install and configure the Hortonworks Hive ODBC Driver to connect to the server
3. Install the SAP BusinessObjects Information Design Tool and Web Intelligence Rich client
4. Create a universe on top of Hortonworks Data Platform with Information Design Tool
5. With a BI client tool, use the universe to run queries on Hortonworks Hive

## 1.7 Important note about support

This document shows how to create a universe on Hortonworks Data Platform using the SAP BusinessObjects Generic ODBC connection.

At the time of writing of this document, SAP provides support of this configuration via the 'Generic ODBC support' policy. If an issue is found with this configuration, a fix can be provided only if the same problem can be reproduced on the SAP reference configuration (today with Microsoft SQL Server). For future changes in the support policy, you can check the online Platform Availability Matrix which can be found at <http://service.sap.com/PAM>

## 2 Finding and installing the Hortonworks software

To run queries on a Hortonworks Data Platform you first have to install one and then install the client driver needed to connect to it. The detailed information on how to install and run the distribution is available on the Hortonworks sites, in this section we provide only a few basic steps to get you started with the solution.

### 2.1 Find, install and start a Hortonworks Data Platform 2.0

You can obtain the Hortonworks Data Platform 2 (HDP 2.0) from the following site:

<http://hortonworks.com/products/hdp-2/>

The HDP 2.0 version contains the Hive 0.12.0 server which is used in this document.

To quick start your tests you can download the Hortonworks Sandbox which is already installed and pre-configures on a virtual machine from this link:

<http://hortonworks.com/products/hortonworks-sandbox/#install>

For the examples of this document, you can install the Hortonworks Sandbox on the same physical Windows machine where the Information Design Tool is installed. The workflows described later have been performed in this configuration.

You can choose one of the various virtual machine systems provided; the test for this document was done using the VMware version with VMware Player 6.0.1

VMware player can be downloaded free of charge at this link:

[https://my.vmware.com/web/vmware/free#desktop\\_end\\_user\\_computing/vmware\\_player/6\\_0](https://my.vmware.com/web/vmware/free#desktop_end_user_computing/vmware_player/6_0)

#### Note

A test run with a previous version of VMware Player failed because of network issues, it is hence recommended to use the 6.0.1 version or a later one.

After downloading the Hortonworks Sandbox you should open it with VMware Player and set the correct settings:

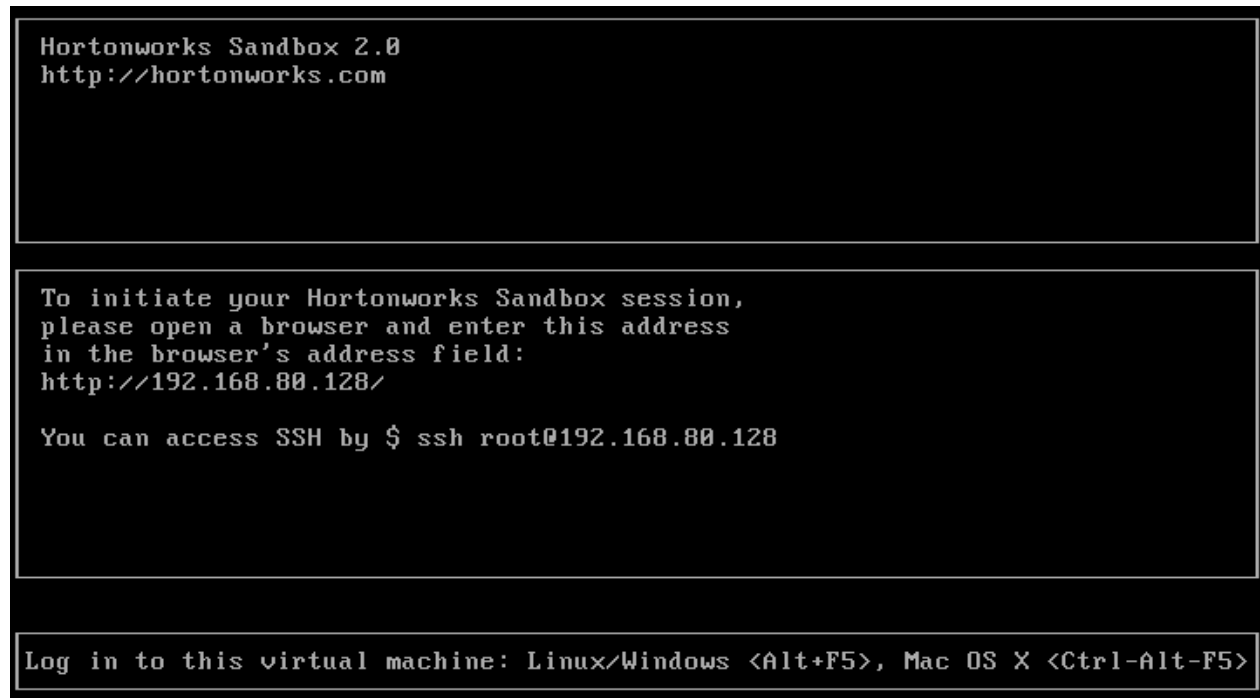
- Set the virtual machine memory to at least 4Gb
- Set the network to **Host-Only** (for running the tests with the universe on the same physical machine)

You also need to make sure that the VMware virtual network adapter which enables connections between the physical machine and the virtual machine is activated. To do so you should:

- Open the Windows **Network and Sharing Center**
- Go to **Change Adapter Settings**
- Make sure the **VMware Network Adapter VMnet1** is enabled

You can now start the virtual machine.

At the end of the boot process (if successful) you have a screen similar to the one shown in Figure 1.



```
Hortonworks Sandbox 2.0
http://hortonworks.com

To initiate your Hortonworks Sandbox session,
please open a browser and enter this address
in the browser's address field:
http://192.168.80.128/

You can access SSH by $ ssh root@192.168.80.128

Log in to this virtual machine: Linux/Windows <Alt+F5>, Mac OS X <Ctrl-Alt-F5>
```

Figure 1: IP address of the machine after a successful boot

From the screen you can see what the IP address of the virtual machine is. You can use that address in a browser on the physical host machine to check that the Hortonworks system is up and running.

By default, the user name to access HDP 2.0 is 'sandbox' and there is no password.

#### Note

Based on your network security policy, firewall settings and proxy settings, you might not be able to connect to the virtual machine with a browser. You can check if the machine is responding with a **PING** or by running a test via the ODBC manager, as discussed in the next section.

The Hortonworks system is now available; the next step is to connect to Hive using the ODBC middleware.

## 2.2 Find, install and configure the Hortonworks Hive ODBC Driver

The Hortonworks ODBC middleware can be obtained from the following site:

[http://hortonworks.com/products/hdp-2/#add\\_ons](http://hortonworks.com/products/hdp-2/#add_ons)

You have to install the Windows 32bit ODBC driver on the machine where you run Information Design Tool.

If you want to connect to Hortonworks Hive from the BI platform server (e.g for usage by Web Intelligence online, SAP Lumira, Design Studio, Dashboards, Predictive Analysis, Explorer, Crystal Reports Enterprise) then you have to install on the server the 64bit ODBC middleware (and only the 64bit version).

For this tutorial, using only a client machine, we use only the 32bit ODBC driver.

#### Note

On Windows 64bit machines you have both the 32bit and 64bit ODBC managers.

The 32bit driver manager can be launched from the following path:

**C:\Windows\SysWOW64\odbcad32.exe**

The 64bit driver manager from this path: **C:\Windows\System32\odbcad32.exe**

Figure 2 shows graphically the correct ODBC deployments on a test client machine which contains the Hortonworks Sandbox as well. This deployment has been used to create the samples of this document. In figure 2 (and in figure 3), the “UNX” box represents the universe being created. The universe connects to the HDP system via the ODBC driver.

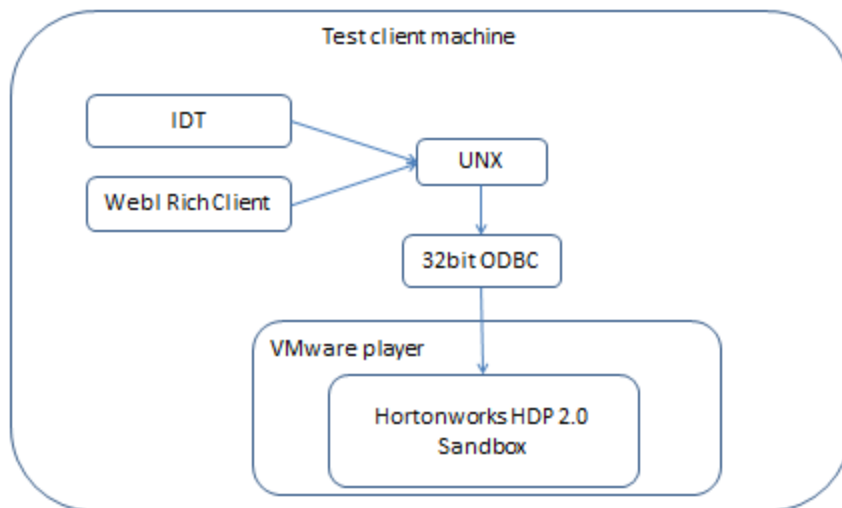
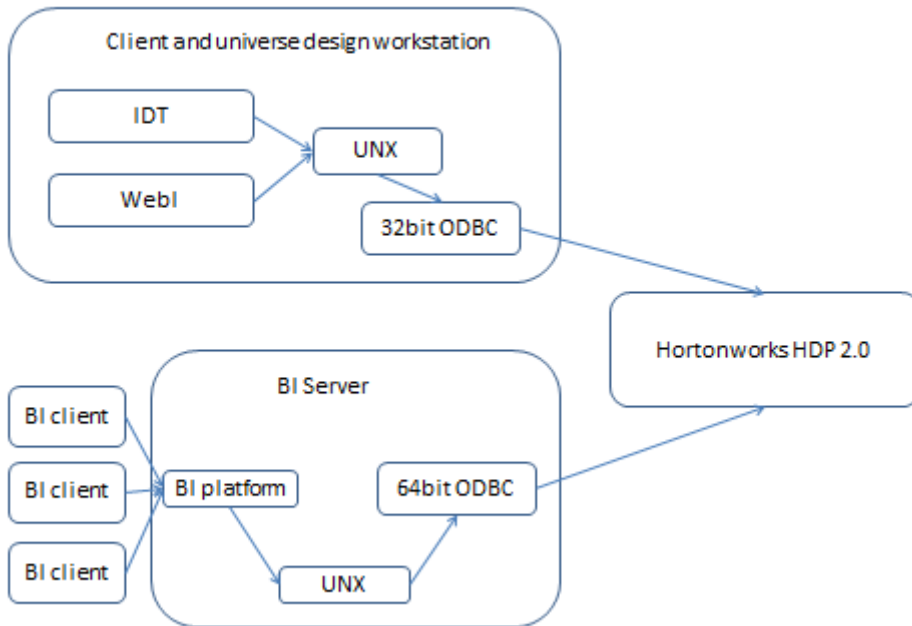


Figure 2: Sample client test deployment

Figure 3 shows the usual deployment with a full HDP 2.0 system and a BI platform



**Figure 3: A real-life deployment**

On the machine with Information Design Tool, after you download the ODBC driver installation file, you execute it and, at the end of the setup, you find a new entry in the Windows 32bit ODBC manager called "Hortonworks Hive ODBC driver".

You can now create a new system DSN to connect to the newly installed Hortonworks Sandbox.

A sample ODBC configuration is displayed in Figure 4

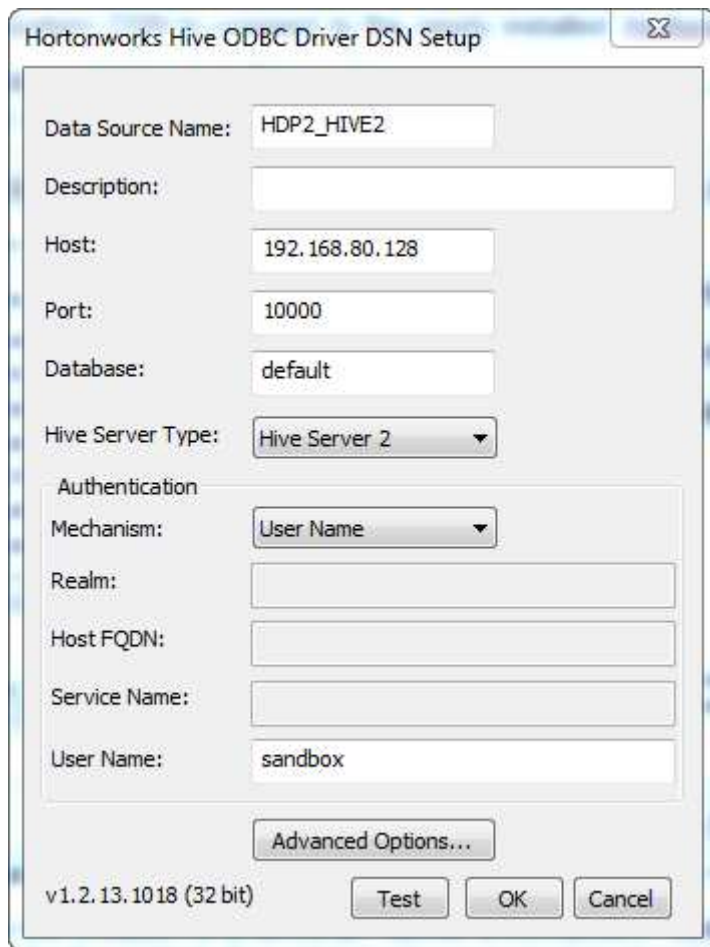


Figure 4: ODBC configuration

To create a correct DSN you have to :

- Provide a **Data source name**. This name is then used in IDT to create the connection. In this document we set the name to “**HDP2\_HIVE2**”
- Set the ‘**Host**’ field to the IP address displayed in the HDP 2.0 screen
- Set the **Hive Server Type** to **Hive Server 2**
- Set the **Authentication Mechanism** to **User name**
- Set the **User Name** as “**sandbox**” if you are connecting to the sandbox system

All the other parameters can be left as they are.

If you install the 64bit ODBC driver on the BI server machine, you have to make sure that the **Data source name** value is the same as the one used on the designer workstation.

You can now click the Test button to check if the middleware is correctly configured. If the test is successful you can go to the next step and start creating a universe.

### 3 Creating a universe on top of Hive

Your HDP 2.0 system or the Hortonworks Sandbox are now up and running and it is possible to connect to the Hive server on it using the ODBC driver, you are ready to build your first universe.

There are four main steps in the creation of a universe: define a connection, build a data foundation on it, create a business layer and, finally, publish the universe so that client tools can consume it.

You can launch the Information Design Tool (IDT) and get started with the first step.



### 3.1 Creating the connection in IDT

In your local IDT project you have to create a new relational connection on the HDP 2.0 Hive server with the following steps:

- Select the project and folder you want to use and then choose the **New • Relational Connection** command.
- In the popup window provide a name for the new connection, in this test we use 'HIVE HDP2' and click next
- Choose the **Generic ODBC3 datasource** connectivity as shown in Figure 5 and click **Next**

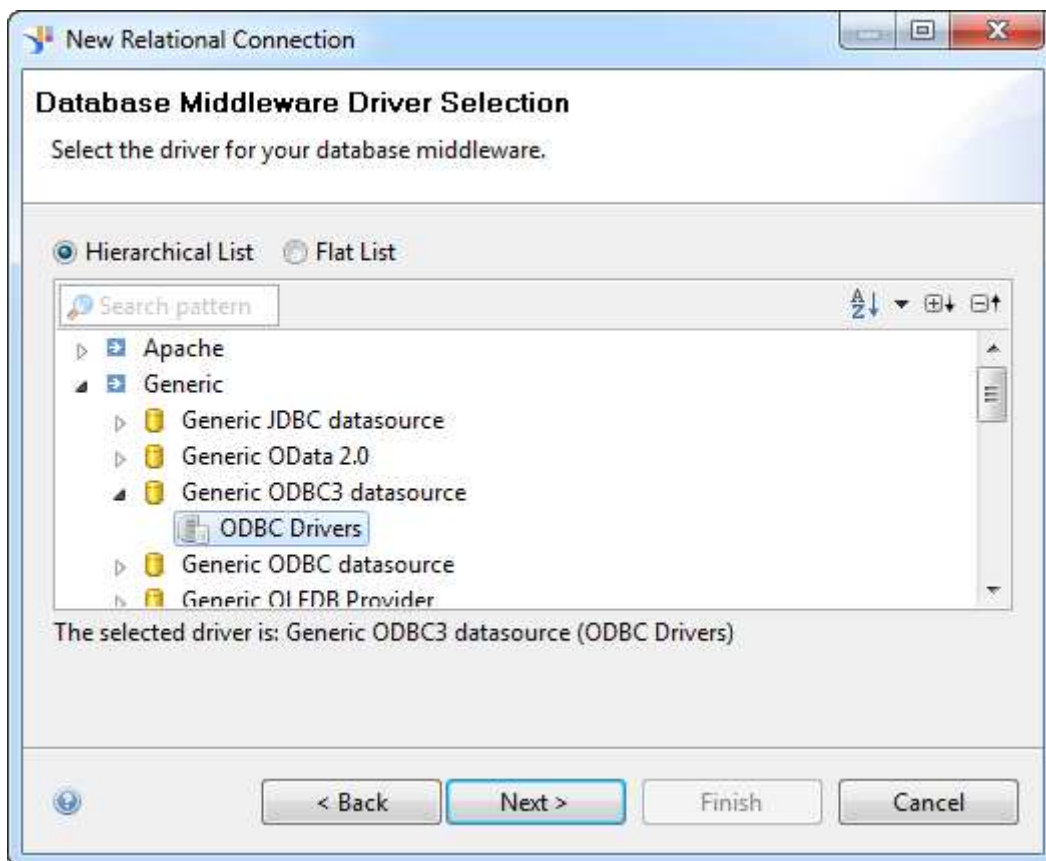


Figure 5: Choice of the ODBC 3 driver

- You then set the correct data source name, user name and password information to connect to your HDP 2.0 system as show in Figure 6. The Data Source Name references the Data Source Name value defined in Figure 4.

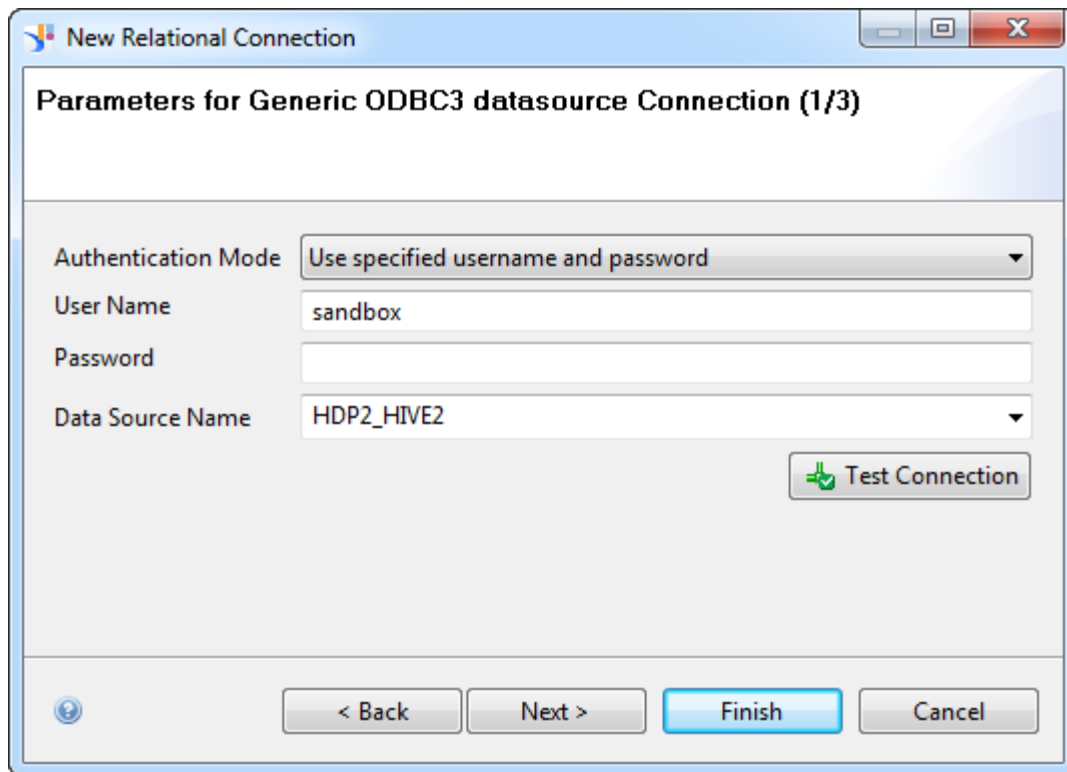


Figure 6: The IDT connection to Hortonworks

- You should test the connection to make sure that it works fine
- Click on the Finish button (there is no need to complete the following optional steps)

The connection is now defined and can be found in the local project as shown in Figure 7.

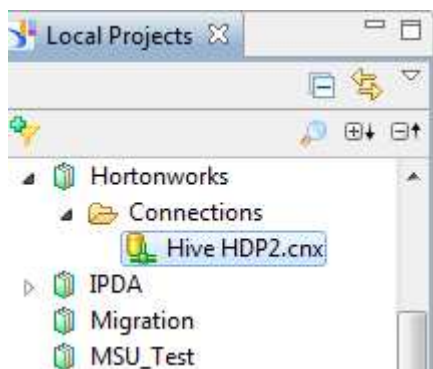


Figure 7: The connection in the local IDT project

When the connection is available, the next step is to build a data foundation onto it.

### 3.2 Creating the data foundation in IDT

You are now going to build a data foundation pointing to the HDP 2.0 server via the connection you just defined.

To create the data foundation you can follow the steps below:

- Right click in the same project of the connection and select **New ▪ Data Foundation**
- Give a name to the data foundation, in this example we call it **“Hive HDP2 DF”** and click next

- Select the **Single Source** type (for the time being it is not possible to build multi-source data foundations using the generic ODBC driver)
- In the connection list select the newly created connection to HDP 2.0 as shown in Figure 8. If the connection doesn't appear here then it means that you are not creating the data foundation in the correct project. Make sure you are working in the project where the connection is defined.

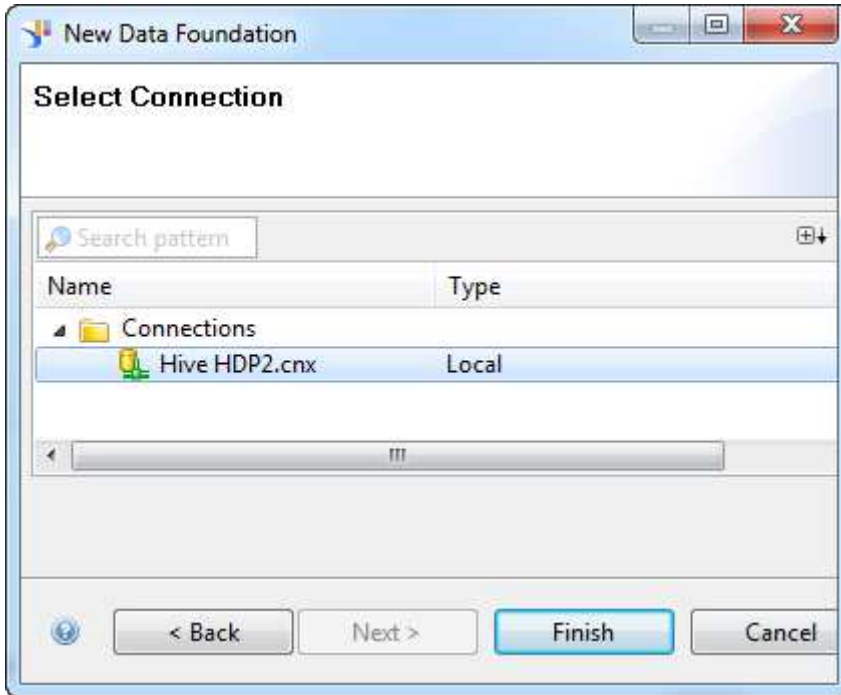


Figure 8: Select the connection to HDP 2

- Click finish

The data foundation now opens in its editor and you can start adding tables into it from the Hive 2 server.

In the Hortonworks Sandbox there are two tables with sample data "**sample\_07**" and "**sample\_08**". Those tables contain information about average salaries and number of workers having a certain job description respectively in 2007 and 2008. We are going to build a simple universe letting you query information from those tables.

The Hive server requires catalog, table and column names to be wrapped in quotes, the following steps show how to build a simple data foundation taking into account this requirement.

- In the data foundation editor you should expand the connection catalog to show the two sample tables under the "**default**" catalog.
- You can now drag and drop those two tables from the connection into the data foundation editor space as shown in Figure 9.

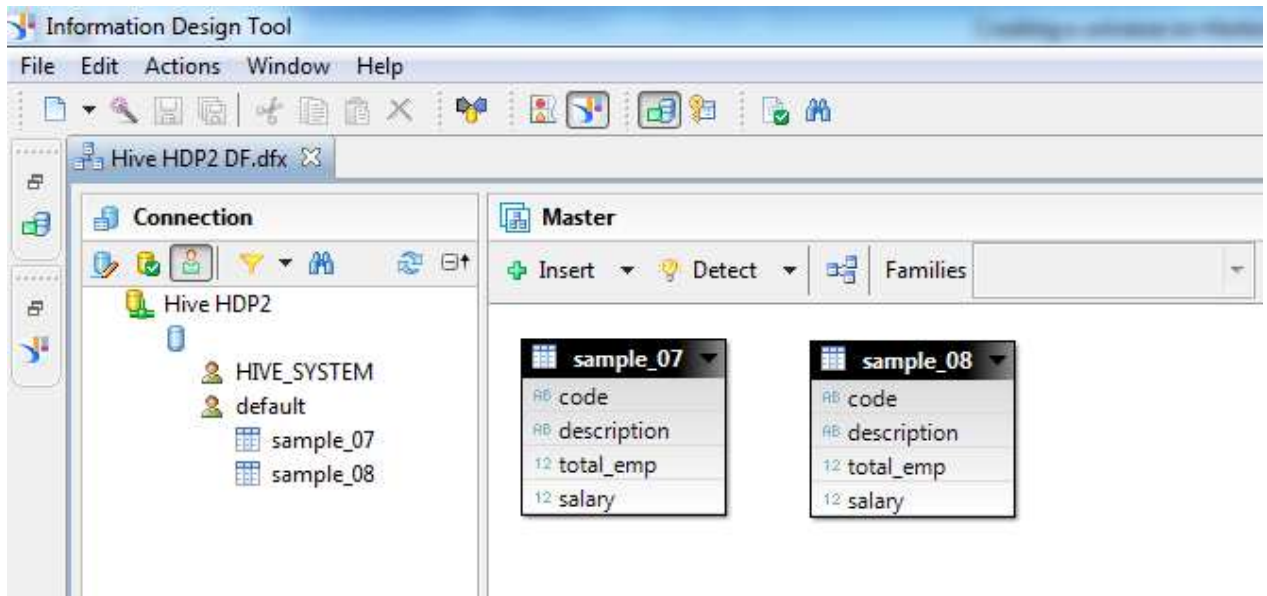


Figure 9: The two sample tables in the data foundation

- To set the double quotes around the catalog name you have to select both tables, right click on them and choose the **Change Qualifier/Owner...** menu command as shown in Figure 10

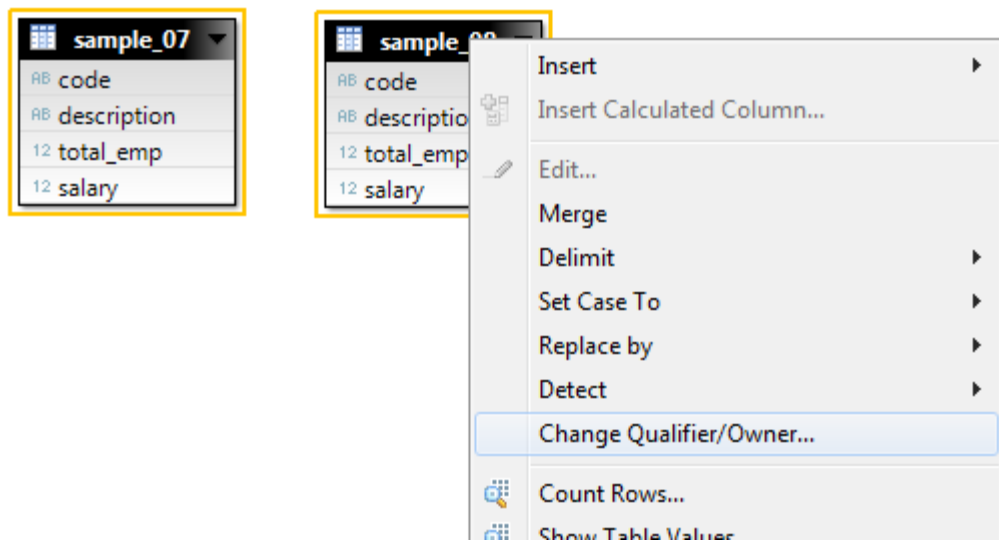


Figure 10: The command to set the correct catalog name format

- In the popup window you have to check the **Delimit** box next to the owner name as shown in Figure 11. Then click ok. The Delimit option wraps the owner name between double quotes. When connecting to HDP 2.0 the owner name and the catalog name coincide.



Figure 11: Wrapping the catalog in double quotes

- To set the double quotes around the table and column names you have to select the tables again, right click on them and choose the **Delimit** command. In the drop down list you should select the **Yes (Tables and Columns)** option as visible in Figure 12

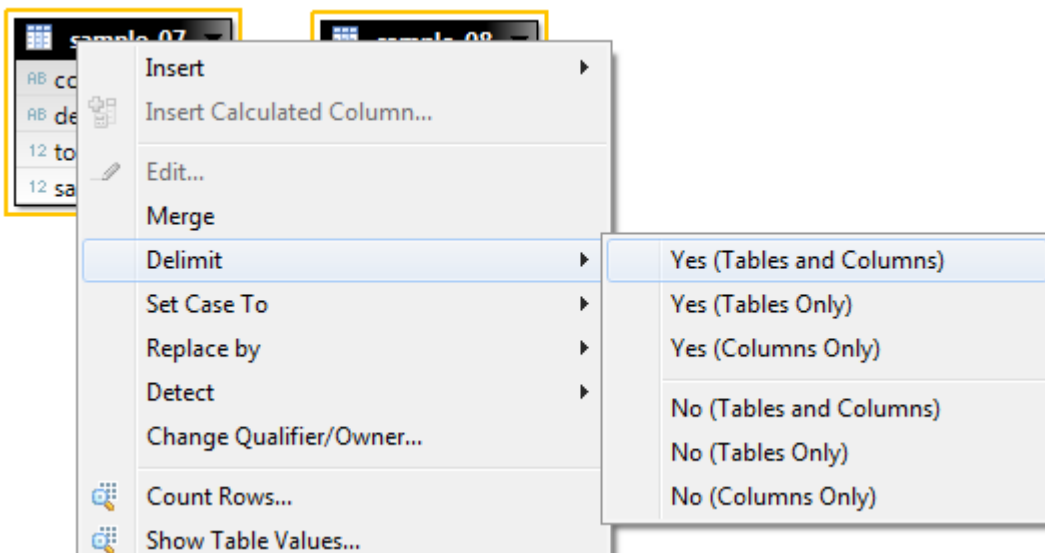


Figure 12: Putting double quotes around the table and column names

- To complete the sample data foundation you can create a join between the two tables by dragging the **"sample\_07"."code"** column on top of the **"sample\_08"."code"** column. The final data foundation should look like the one in Figure 13.

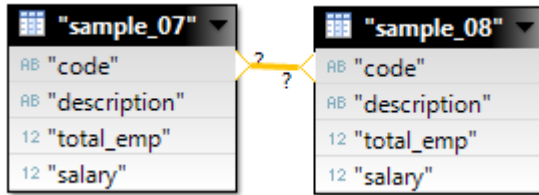


Figure 13: the resulting data foundation

The two question marks on the join line are just a reminder that no cardinality has been set. This is not a problem in this simple test.

You can now save the data foundation and go to the next section to create a business layer.

### 3.3 Creating the business layer in IDT

You are now going to create a simple business layer which can be used to retrieve the 2007 and 2008 salaries and calculate the salary increase for each job title.

To create a new business layer you should click into your local project and follow the steps below:

- Right click in the project (or in a project folder) and select the command **New ▪ Business Layer**
- In the following window select to build the business layer on top of a **Relational Data Foundation** and click **Next**
- You then set a name for the business layer, in this example we are going to call it **“Hive HDP 2 BL”**
- In the list of data foundations select the one you just created (named **“Hive HDP 2 DF.dfx”** in our example)
- As shown in Figure 14, make sure to uncheck the **Automatically create folders and objects** option, then click **Finish**.

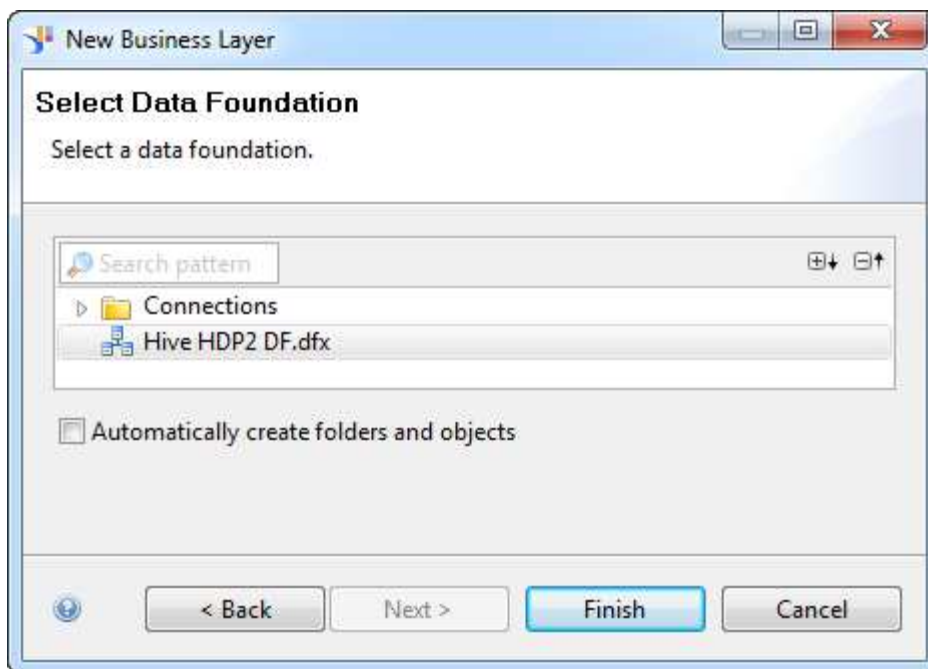


Figure 14: Selecting the data foundation and unchecking the automatic detection

The new empty business layer has been created and opens in its editor. You can now start adding the business objects which are going to be surfaced to the end users in the BI client tools.

- In the business layer editor right click on the name of the layer and select the **New • Folder** command as shown in Figure 15.

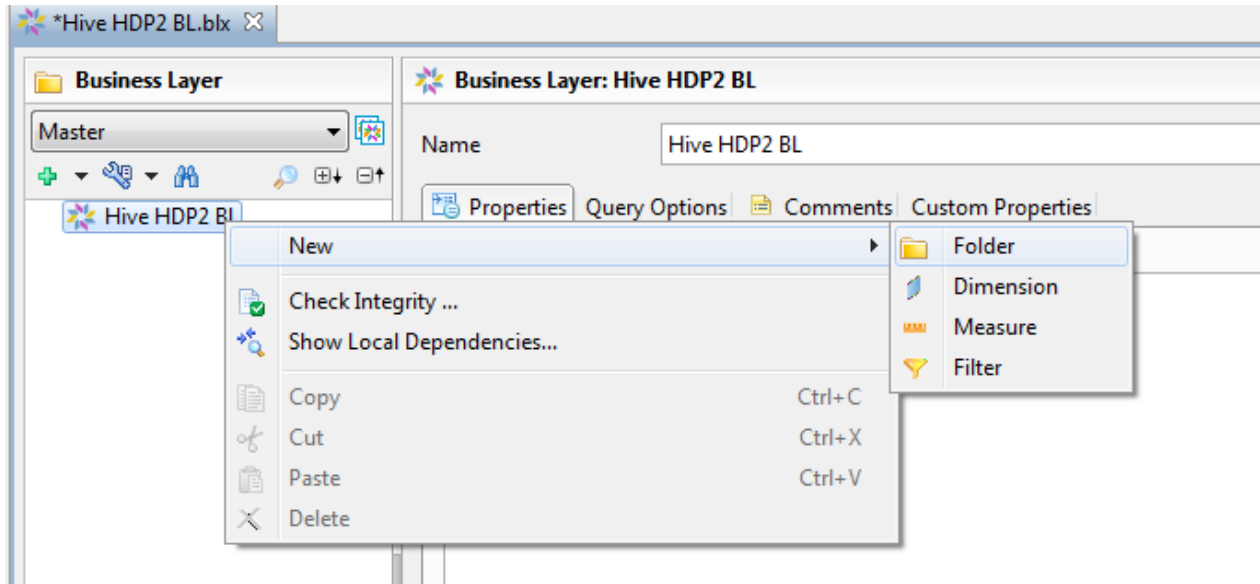


Figure 15: defining an object folder

- You then provide a name for the newly created folder, in our example we call it **“Information”**
- Now you can select all the columns in the **“table\_07”** appearing in the lower space of the screen which shows the data foundation. After selecting you can drop them into the **“Information”** folder by dragging them with the mouse as shown in Figure 16.

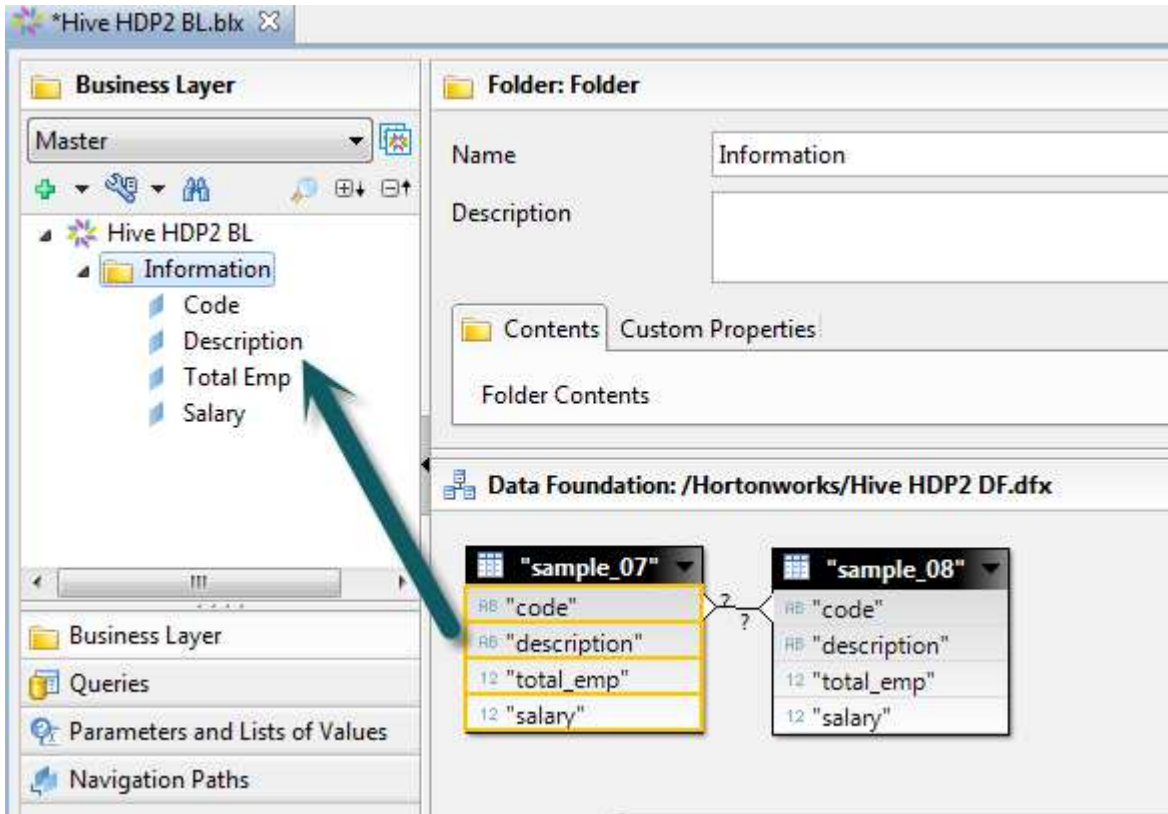


Figure 16: Create some objects by drag and drop

- Four dimension objects have been created, one for each table field
- You should now select the **Total Emp** object and rename it to **Total Emp 2007**. Similarly rename the **Salary** object to **Salary 2007**
- You can now drag from the “**sample\_08**” table the fields “**total\_emp**” and “**salary**” into the object folder. You then rename the created objects as **Total Emp 2008** and **Salary 2008**

The universe already contains a few objects which can be used to run queries on the source. You can add an additional measure which provides the salary increase between 2007 and 2008. Follow the steps below to create the measure:

- Right click in the **Information** folder and select the **New ▪ Measure** command
- In the measure editor provide a name for it, in this example we call it **Salary increase 2008/2007**
- In the measure Select statement you can put the following definition: **sum("default"."sample\_08"."salary"- "default"."sample\_07"."salary")**
- This formula returns the difference between the 2008 and 2007 salaries and aggregates the differences with a sum to calculate a global salary variation when asked.
- The final business layer should look like the one shown in Figure 17

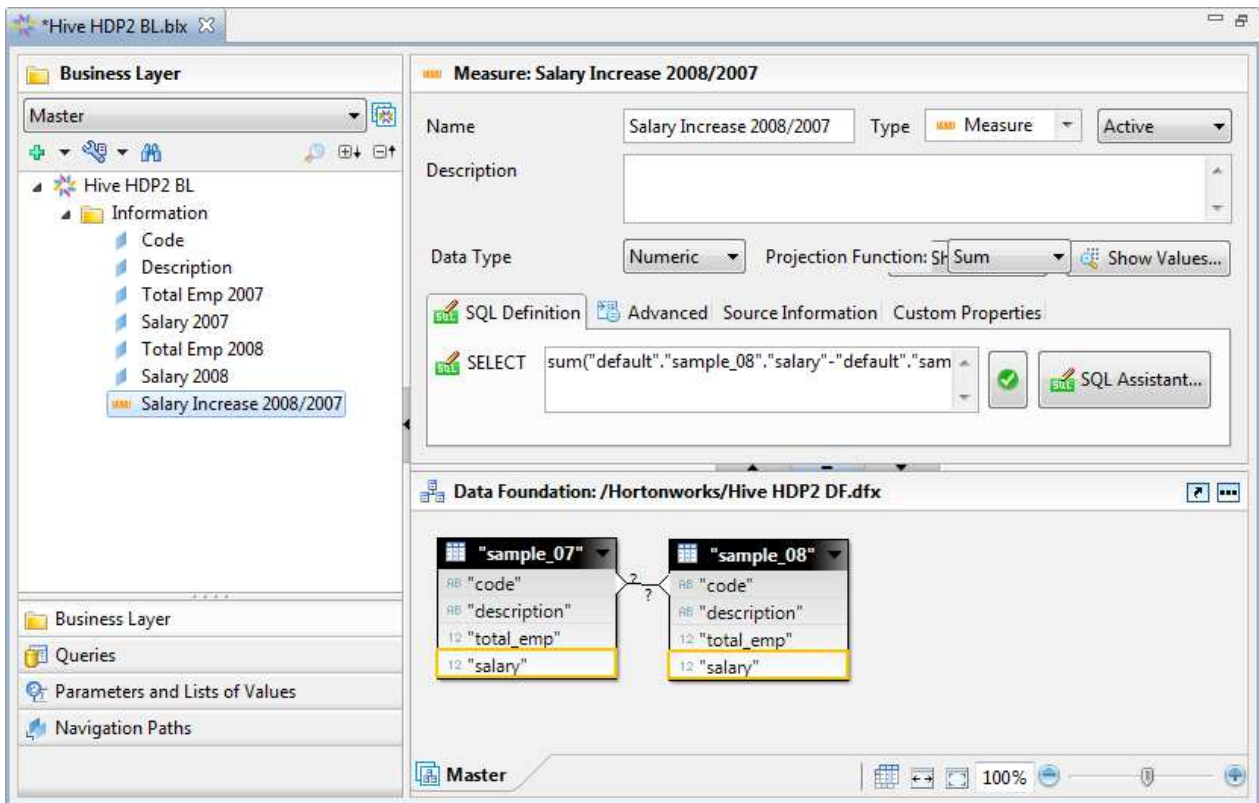



Figure 17: the final business layer

You can now check if the universe works correctly by running a couple of queries within IDT.

- In the main business layer editor open the **Queries** subpanel (on the bottom left in Figure 17).
- Create a new query by clicking on the new query button (  ) and ask e.g. for the job description, the average salaries in 2007 and 2008 and the salary increase. Hitting the refresh button you have the results as shown in Figure 18



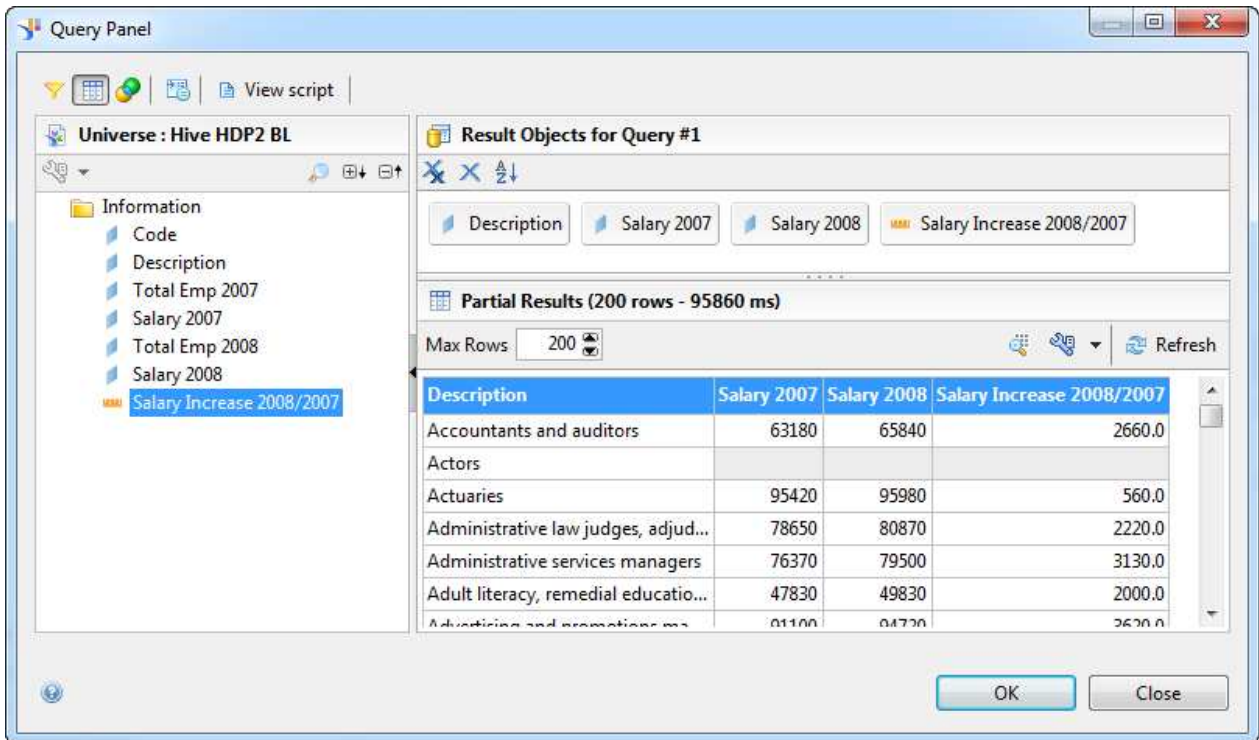

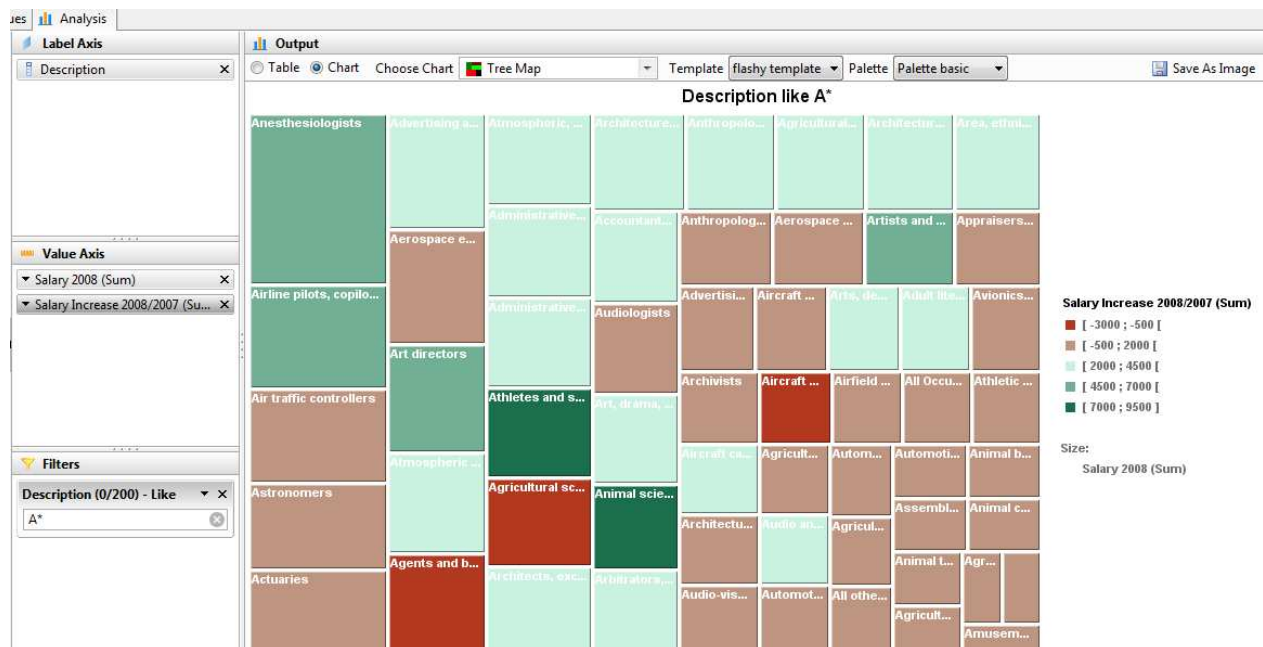


Figure 18: a sample query

- Clicking on the **Advanced preview** button (  ) and then on the **Analysis** tab you can immediately get a graphic representation such as the one in Figure 19 showing the largest salaries (size) and largest increases (increase in green, decrease in red).



- You can create now another test query with only the **Salary Increase 2008/2007** measure. The single returned value is the sum of all increases across all job descriptions (shown in Figure 20).

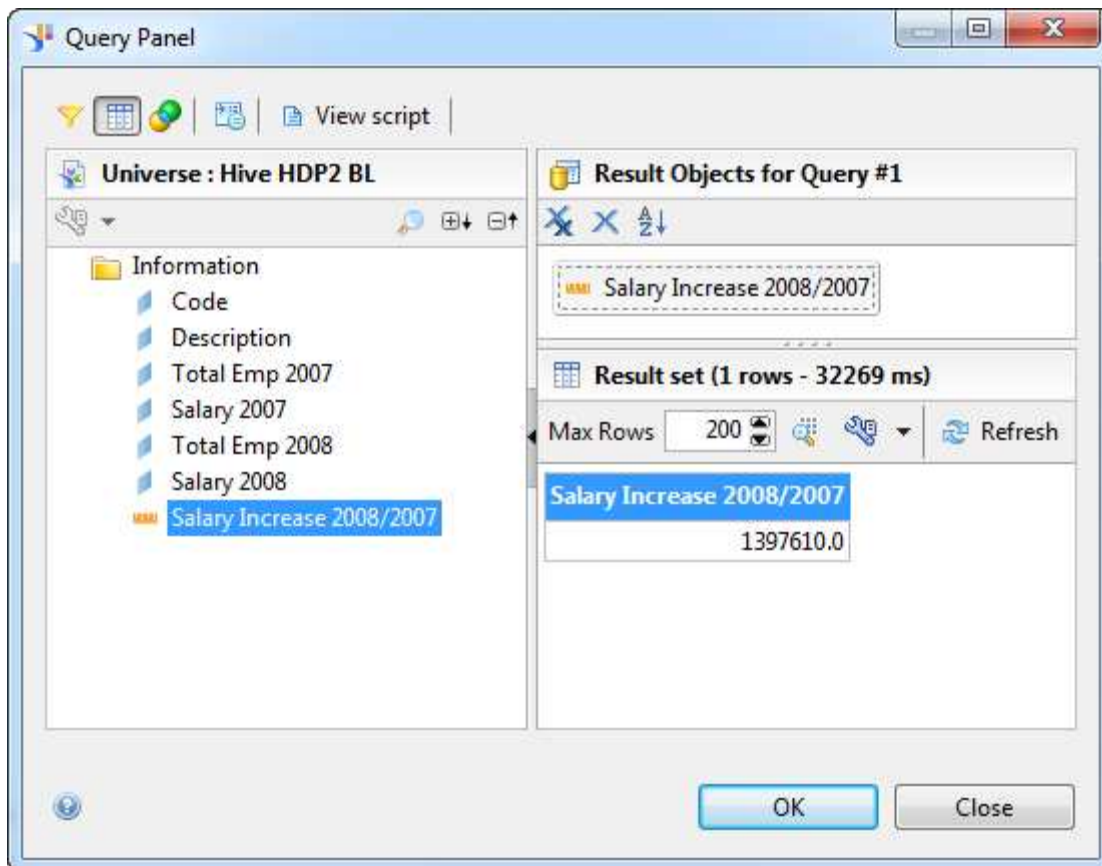


Figure 20: Query aggregating the salary increase across all job descriptions

The business layer is now completed. You can save it and go to the next step to make it available for all client tools.

### 3.4 Publishing the universe

After completing a business layer you have to publish it into the universe format (a file of format UNX) to make it available for consumption to client tools.

The universe can be published either locally or on a server.


You can publish the universe locally on the machine where is IDT. In this case the universe file can be used only by Web Intelligence Rich Client. You can send the file by mail to other users with Web Intelligence Rich Client. In order to use the universe on their machines they need to define a local 32bit ODBC DSN with the same name as the one chosen in the universe connection.

To publish the universe locally follow the next steps:

- In the project, select the business layer and right click on it
- Select the command **Publish ▸ To a local folder...** and click next to keep the default behavior
- By default the universe file is saved in the directory where Web Intelligence Rich Client searches the available universes

You can also publish the universe onto a BI server. In this case the universe is accessible to all client tools connecting to the server (Web Intelligence, Crystal Reports Enterprise, Dashboards, Design Studio, SAP Lumira, Predictive Analysis, Explorer). On the server you have to make sure that there is a 64bit ODBC DSN pointing to the correct HDP 2.0 system and having the same name of the ODBC DSN used in the connection in the Information Design Tool.

To publish the universe on the server you have first to put the connection on the server and then link the business layer to it. You can follow this steps:

- In the local project select the connection to HDP 2.0
- Right click on it and select **Publish Connection to a Repository**
- Select a BI server (or enter a new one) and choose a folder for the connection
- Accept the proposal to create a local connection shortcut
- In the local project, the connection shortcut appears as a new file with the **.CNS** extension. Make sure that on the server you have a 64bit ODBC connection to the HDP 2.0system with the same DSN name used to define the connection in IDT.
- Open the data foundation and change the connection so that the new .CNS file is used. The connection can be changed with the  button as shown in Figure 21.

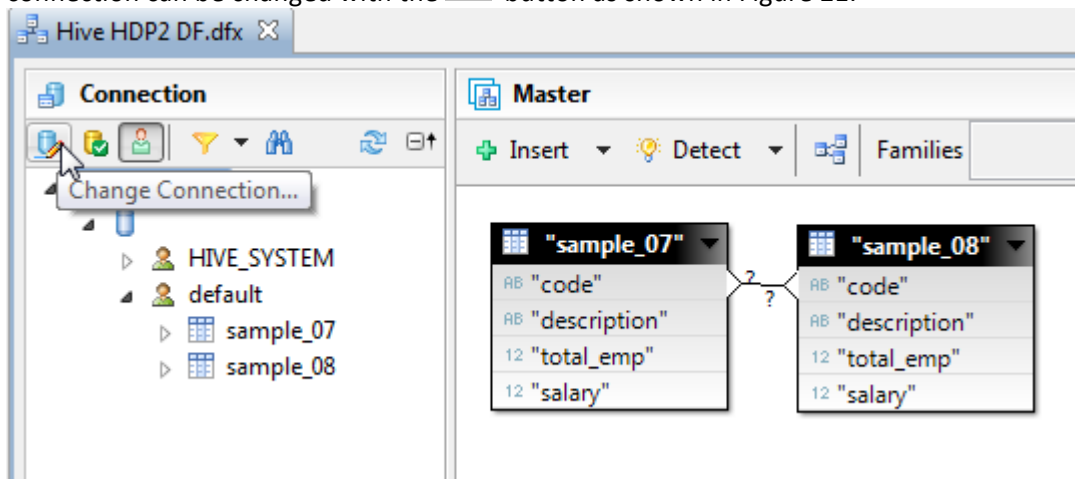


Figure 21: changing connections

- Save the data foundation
- In the local project select the business layer, right click on it and select **Publish - To a Repository** command.
- Select a BI server a folder and save the universe there.

The universe is now ready to be consumed by the client tools.

#### 4 Running a sample query

To provide a simple example, you can now try and run a query with Web Intelligence Rich client on the universe published locally.

- Launch Web Intelligence Rich client on the same machine where you have IDT
- On the initial page click on the Universes tab and select the newly published universe ("Hive\_HDP2\_BL.unx" in our example)
- The query panel opens; you can drag and drop a few objects as shown in Figure 22. Then run the query

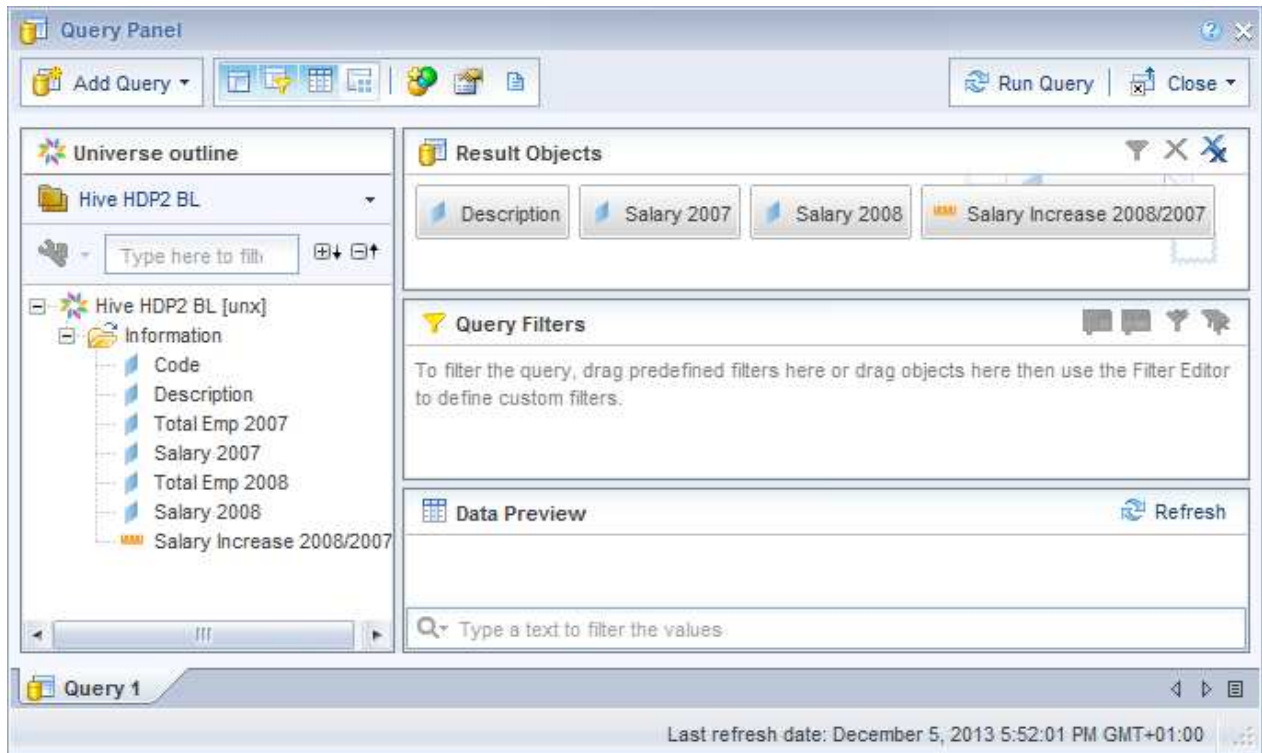


Figure 22: A sample query in Web Intelligence Rich Client

- Once the query is completed the data is returned in the client interface as a table (shown in Figure 23)

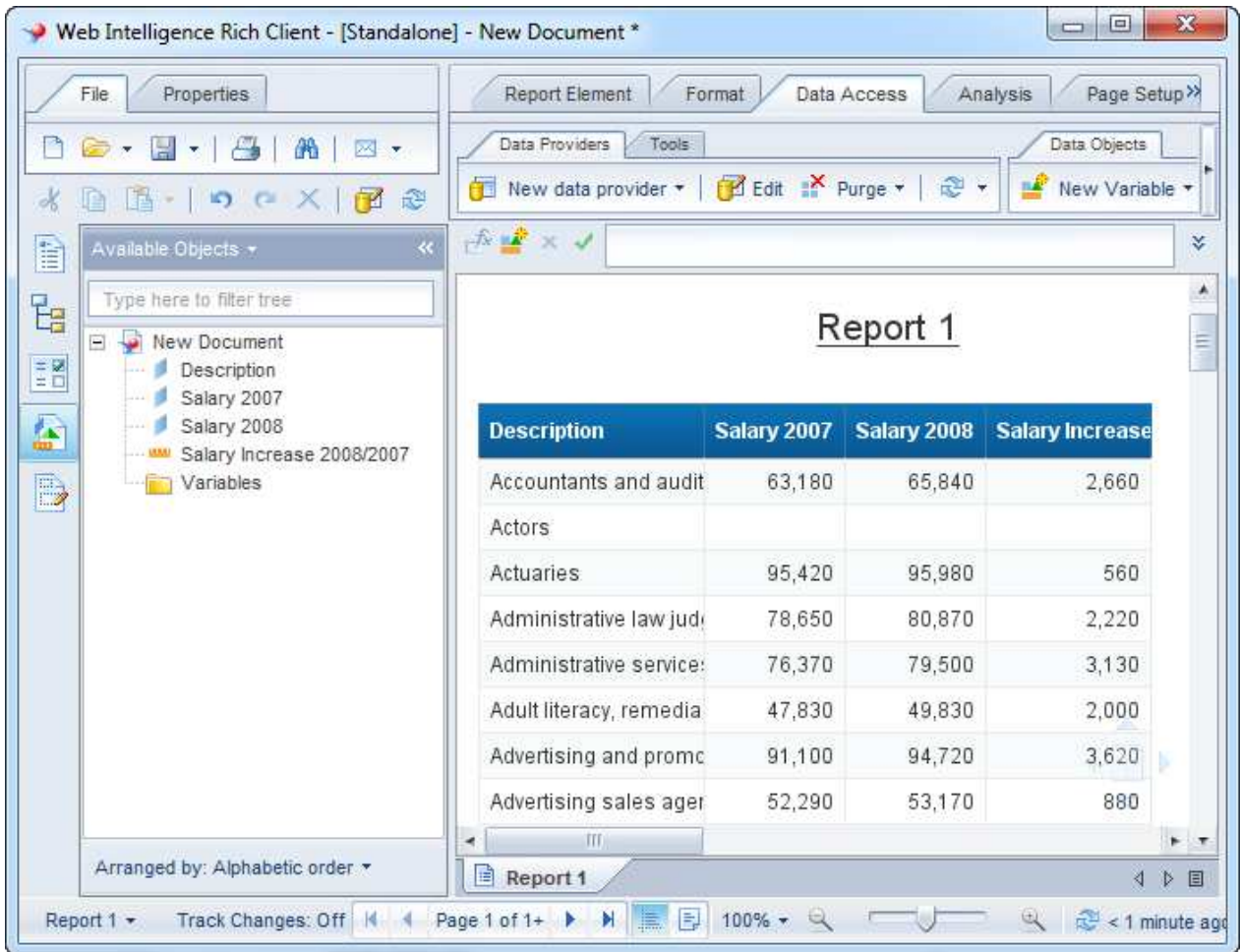


Figure 23: the initial Web Intelligence output

- You can now modify the visualization or add calculations using the Web Intelligence functionalities. In the example in Figure 24 you can see a selection of jobs with the highest salary and highest increase in 2008.

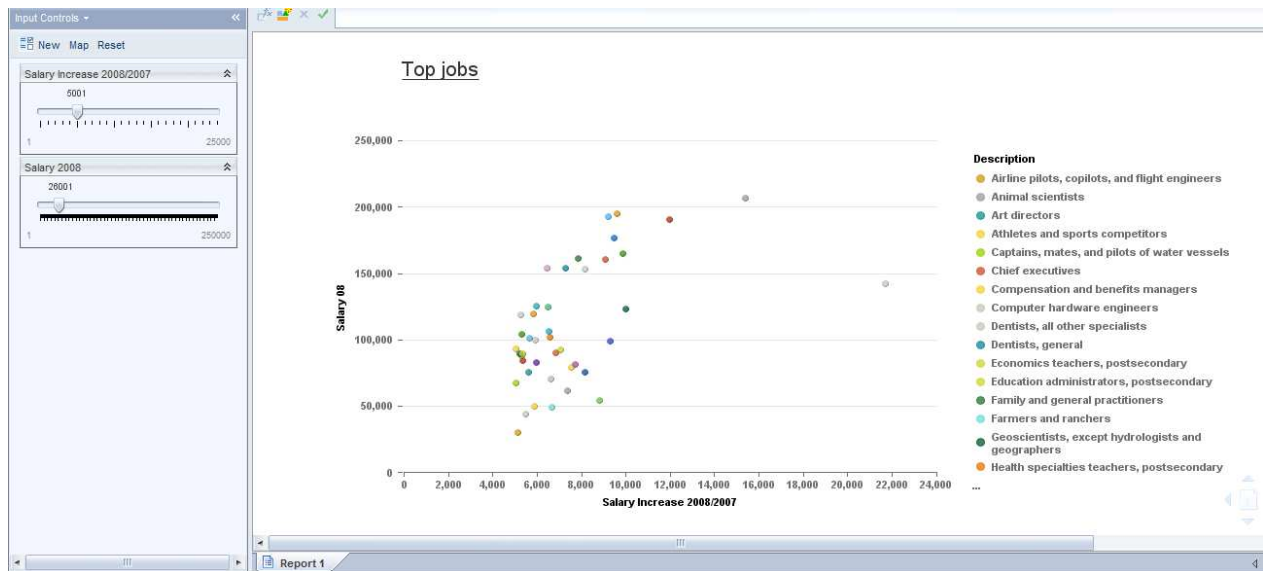


Figure 24: Sample on-report analysis

If you want to connect to the HDP 2.0 with other tools you have to publish the universe on the BI server. With that configuration you can retrieve the data in SAP Lumira, Predictive Analysis, Design Studio, Dashboards, Crystal Reports for Enterprise, Explorer.

## 5 Additional information

Information on Hortonworks' solutions can be found at <http://hortonworks.com/>

Information on SAP Business Objects solutions can be found at: <http://help.sap.com/bobi>

Detailed workflows with videos for creating a universe are available at: <http://scn.sap.com/docs/DOC-8461>