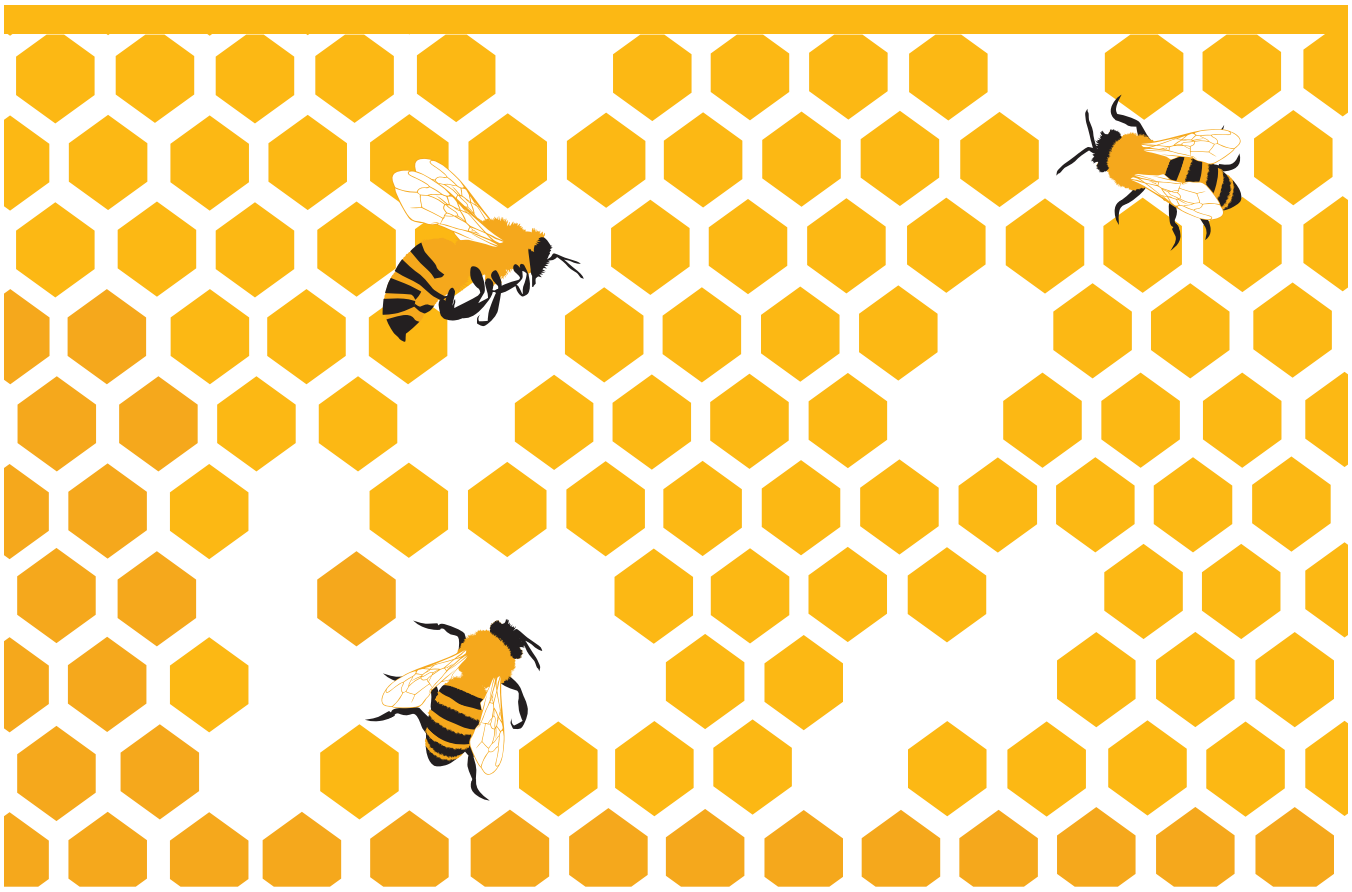


Set Up Hortonworks Hadoop with SAP® Sybase® IQ

TABLE OF CONTENTS

INTRODUCTION

Welcome to setting up Hadoop and Hive with SAP Sybase IQ!

Hadoop is an open-source framework designed to handle big data. It allows large amounts of data to be distributed across clusters of computers. Then, when retrieving the data, a MapReduce algorithm is implemented to divide up the work across the clusters (map) and then recombining the data (reduce).

Hive is an infrastructure which can be used on top of Hadoop. It provides the ability to access data without using MapReduce or code files. It provides HiveQL, which is similar to SQL, which allows for querying data in a Hadoop table.

In previous versions of IQ, the best way to integrate Hadoop with an IQ database was to use user-defined functions and Java code. While this can be useful in certain cases and is still supported, connecting to a Hive server on top of the Hadoop data can be much simpler.

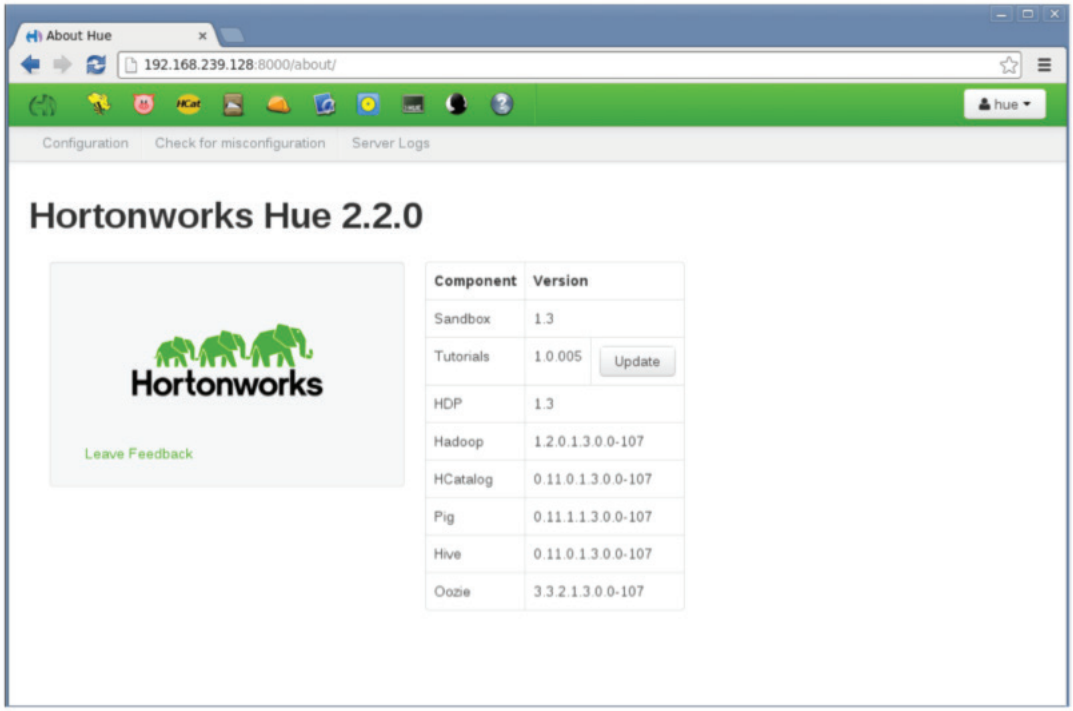
This guide is intended to provide directions on the setup required to use IQ with the Hortonworks distribution of the Hive server and HiveQL. This is not a guide to help with IQ installation or setup. For instructions on IQ setup, please check the SAP website.

INSTALL HADOOP ENVIRONMENT

Hortonworks Sandbox is a distribution of the Hadoop environment that installs as a standalone virtual machine.

Download the product and follow the installation instructions at:

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



Note: This installation requires an installation of VMware Player, Virtualbox or equivalent software. Download links are available on the Hortonworks website.

SET UP IQ ENVIRONMENT

unixODBC

unixODBC is a driver manager that can be used to connect to data sources using specified drivers. We will use this driver manager to open the Hive server data source in IQ, instead of the default driver manager in a typical IQ install.

Install the latest unixODBC driver manager by downloading the .tar.gz file from the following website:

http://www.unixodbc.org/download.html

Unzip and untar as specified on the website.

Then, open a terminal in the unixODBC-2.3.1 folder (or corresponding folder), and run the following commands:

```
./configure
make
makeinstall
```

The command should finish without obvious errors. When installed correctly, you should be able to run the isql or the odbcinst commands.

SAP Sybase IQ creates a soft link in its own directory to the driver manager that it will use, when connecting to data sources. By default, this is directed to the SQL Anywhere driver, so the connection expects a SQL Anywhere or IQ server at all times. SAP Sybase IQ is build using SQL Anywhere connectivity, so using a SQL Anywhere driver is expected.We will direct this to the newly installed unixODBC driver manager to allow us to connect to any ODBC driver.

In the terminal, execute the following commands to view all driver manager shared object libraries.

```
cd $SYBASE/IQ-16_0/lib64
ls -l
```

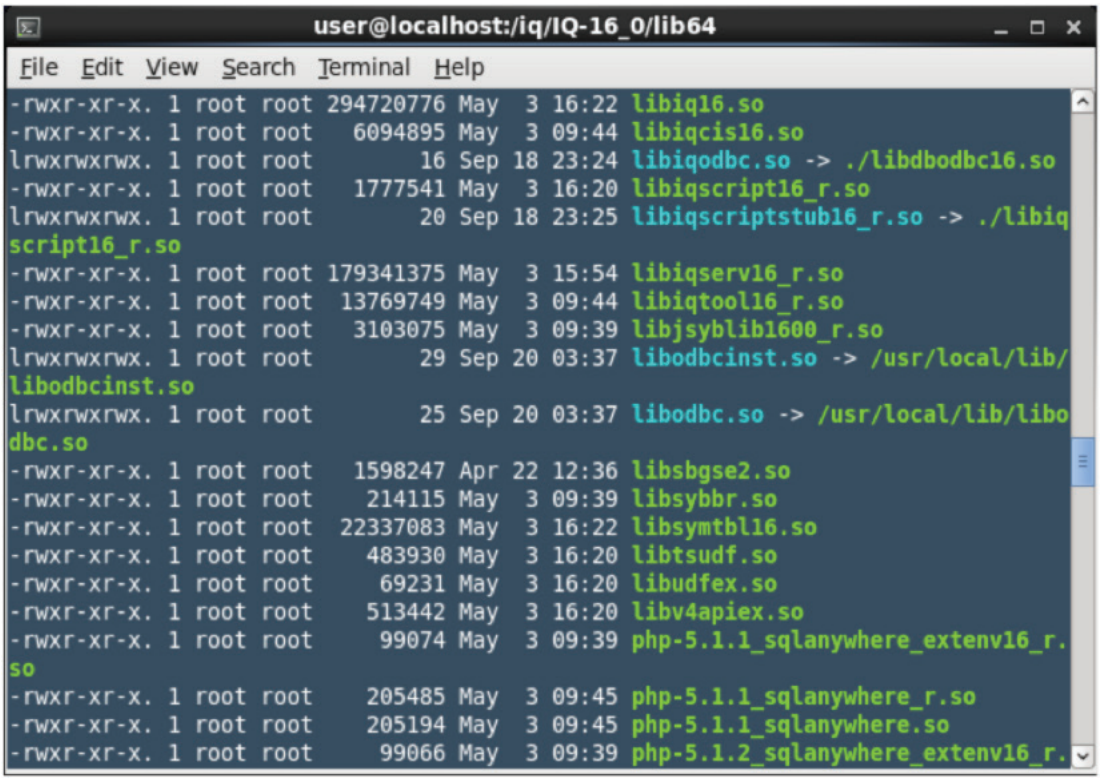
This will display all library files in the folder. IQ uses the libodbc.so and libodbcinst.so files to find the driver manager.

If a libodbc.so link already exists, remove it. Then, we will create links to the library files installed by the unixODBC installation. Ensure that the files exist at the /usr/local/lib location before running this command.

Note: If you cannot find the libodbc.so and libodbcinst.so, run “locate libodbc.so”, which will specify the file location.

```
rm libodbc.so
ln –s /usr/local/lib/libodbc.so libodbc.so
ln –s /usr/local/lib/libodbcinst.so libodbcinst.so
ls –l
```

Now, a list should appear of all files. Check for the appropriate shared object files and ensure they link to the right place. The list should appear like the list on the next page:



Now the unixODBC driver manager will be used.

Install Hortonworks ODBC Driver

In the Linux environment with IQ installed, a Hortonworks ODBC driver is required to use to access the Hive server. This can be downloaded at the following link: http://hortonworks.com/products/hdp/hdp-1-3/#add_ons

Save the .tar.gz file.

Unzip it using the following command:

```
tar –zxvf hive-odbc-native.1.2.13.1018.tar.gz
```

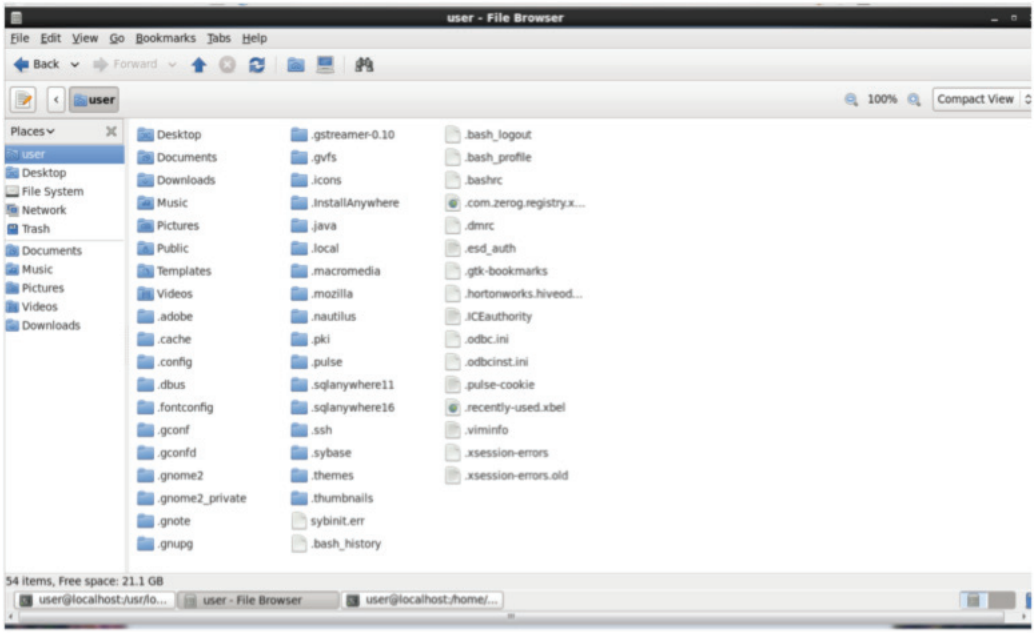
Install the .rpm file in the resulting folder, ensuring you choose the correct one (32 or 64-bit based on your IQ installation):

```
rpm –ivh hive-odbc-native-1.2.13.1018-1.el6.x86_64.rpm
```

DSN

To access connections, you can store attributes to the connection in a DSN. Typically, this connection data is stored in an .odbc.ini file. An .odbcinst.ini file is used to store driver information, including the newly installed driver. The Hortonworks distribution also includes a .hortonworks.hiveodbc.ini file which stores necessary additional information on the connection to the driver.

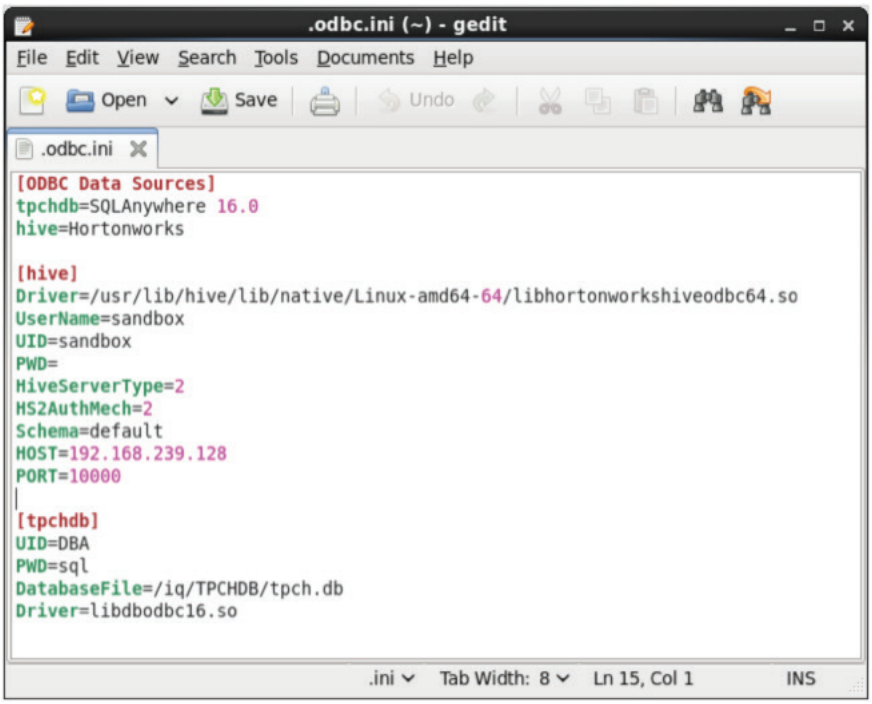
Open your home directory. Check if .odbc.ini or .odbcinst.ini files exist, by checking the folder with hidden files shown (ctrl-h):



We will now update the existing files or create new ones, with the required DSN data for the Hive ODBC driver.

Note: The Hive ODBC installation comes with default configuration files, which can be used, but may have newline errors. To avoid introducing errors and for simplicity, we will create our own. If you wish to use the provided files, ensure that no extra newline characters are introduced.

The .odbc.ini file should be configured as follows:



The name of data source we are creating, in this case “hive”, should be specified with corresponding driver name (in .odbcinst.ini), under [ODBC Data Sources]

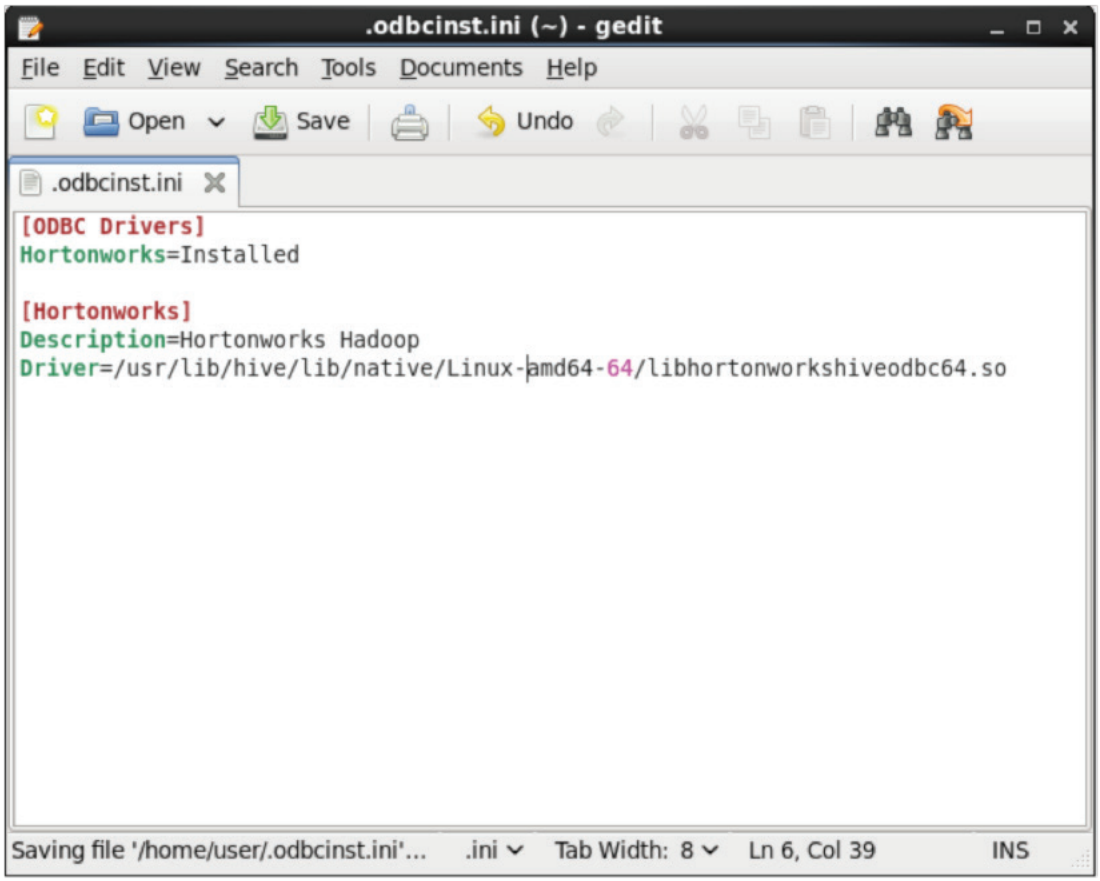
Under [hive], the attributes for the data source should be specified.

The HOST will be the IP address of the Hive server, as specified in the Hortonworks VM (the IP address used to access the web interface). 10000 is the default Hive listening port.

Ensure that HS2AuthMech is specified if the HiveServerType is 2, to avoid connection issues.

The Driver must contain a path to the actual driver .so file.

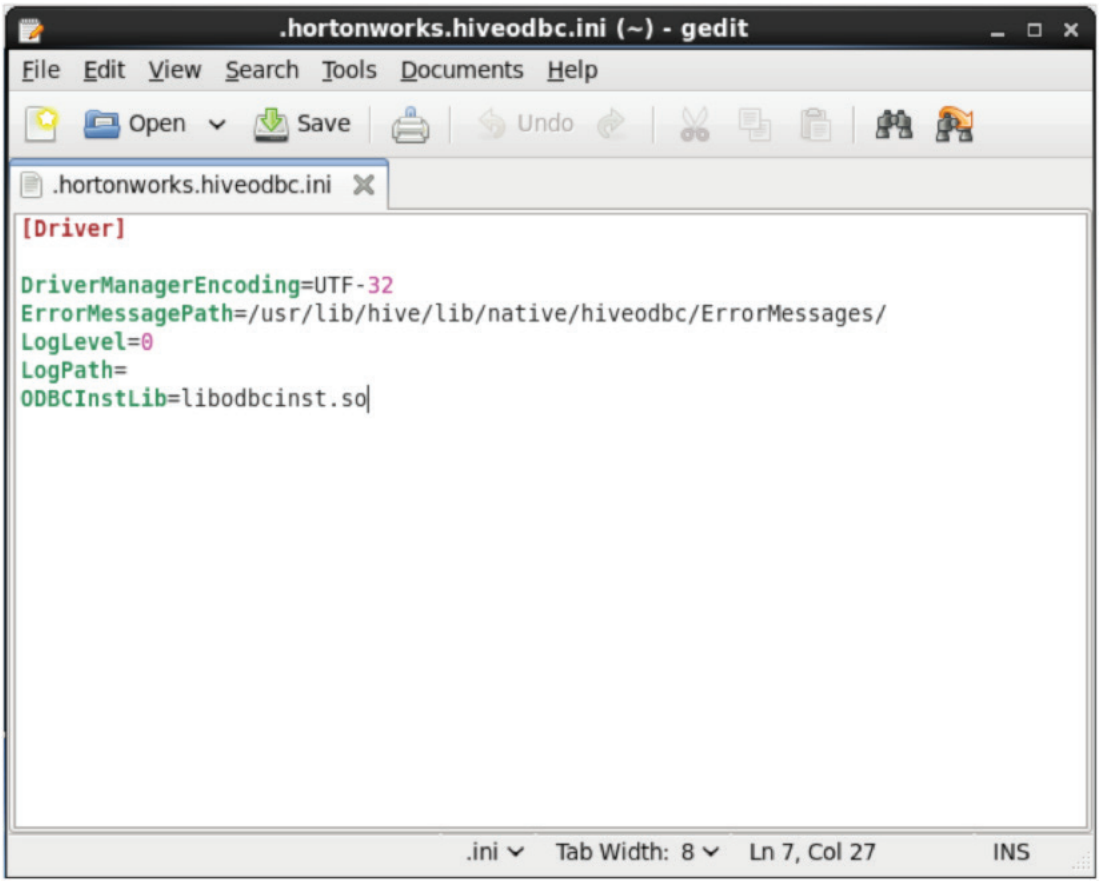
The .odbcinst.ini file should be configured as follows:



The driver field must contain the path to the actual .so file, as in the .odbc.ini file.

Each entry under [ODBC Drivers] is the name of the driver, as referenced in the .odbc.ini file. In this case, the driver is named Hortonworks.

The .hortonworks.hiveodbc.ini file should be configured as follows:



The ErrorMessagePath is the actual path to the error messages folder, which is created when the driver is installed. This is used to return accurate errors when accessing the server.

CONNECT USING IQ

Now that the environment and DSN are set up, we can use them to access the Hive server.

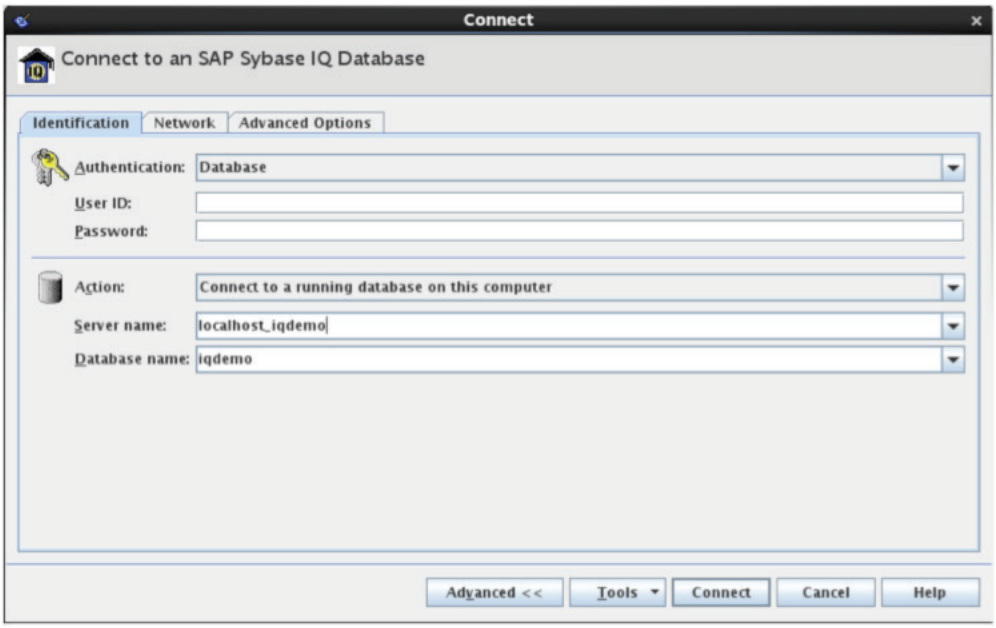
We will access the Hive server from an existing IQ server. To begin, we will start an IQ server. Open a Linux terminal in the directory with the desired database and configuration files. Run a command similar to the follow, with the correct variable names:

```
start_iq @iqdemo.cfg iqdemo.db
```

Now we will open Interactive SQL and connect to that running server:

```
dbisql
```

In the prompt, connect to the server which was started above.



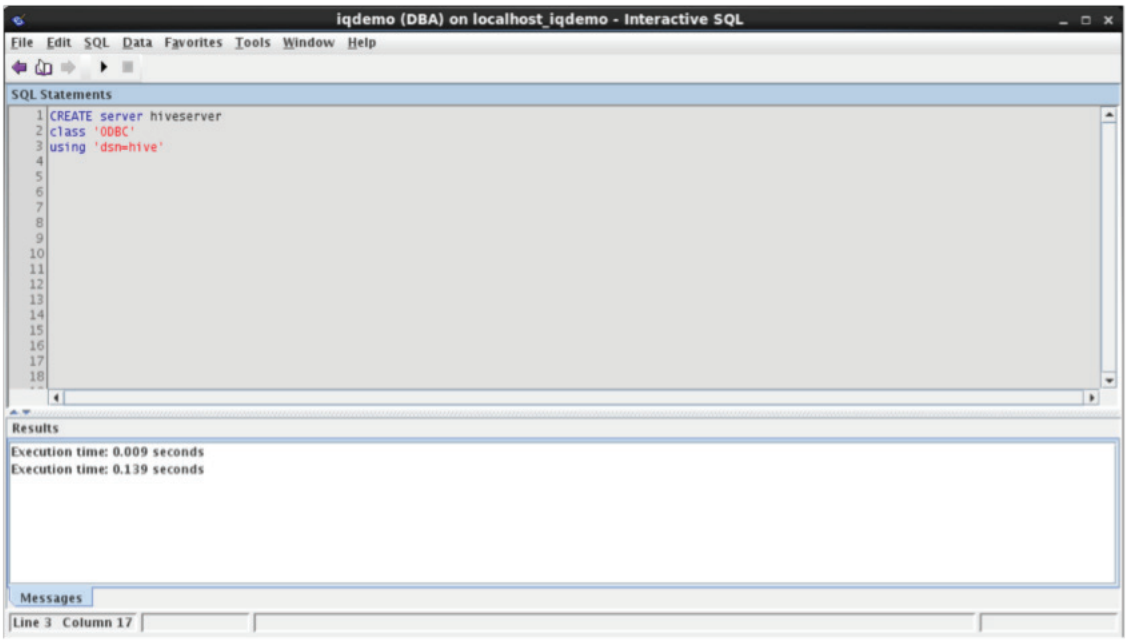
Note: You can connect to the Hive server directly, using the “Connect with an ODBC Data Source” option, if you want to access the Hive server as a server, without using proxy tables. This, however, is not supported and does not allow you to access both SA data and Hive data simultaneously.

Create Remote Server

Now, in the SQL Statements window, we will create a remote server.

Run the following command, choosing a server name (hiveserver) and specifying the DSN as created above.

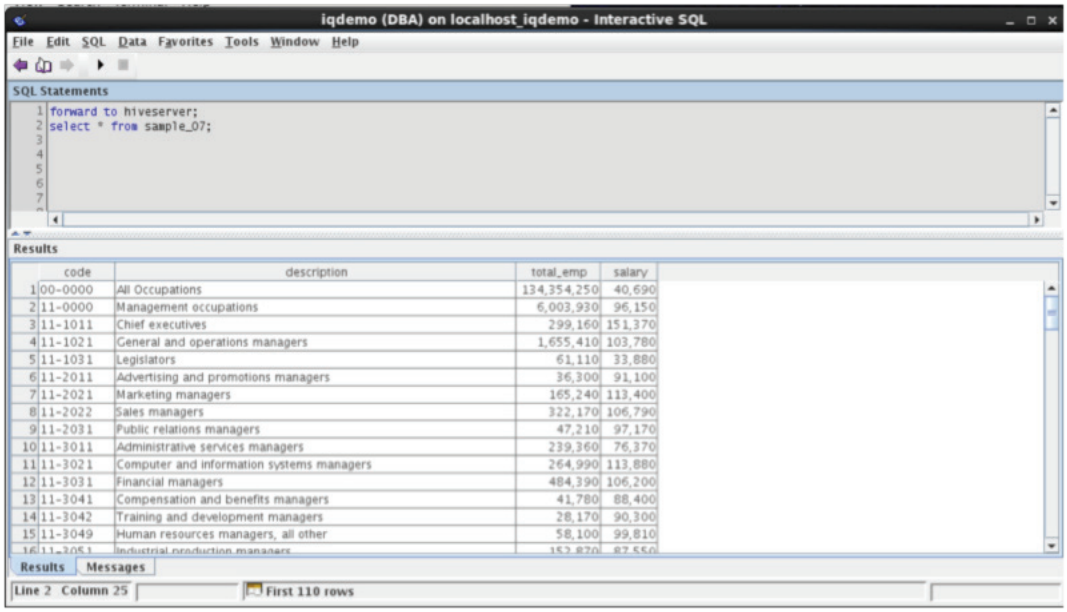
```
CREATE server hiveserver
class ‘ODBC’
using ‘dsn=hive’
```



Access Hadoop from Server

Data can be retrieved from the newly connected Hive server using normal SQL statements, provided that those statements also exist in HiveQL. Select statements can be run with the following commands:

```
Forward to hiveserver;
Select * from sample_07;
```



Note: Existing data, in the Hadoop tables, can also be accessed by accessing the IP address specified in the Hortonworks VM in a web browser.

JOIN also works in the same way.

Access Hadoop from Proxy Table

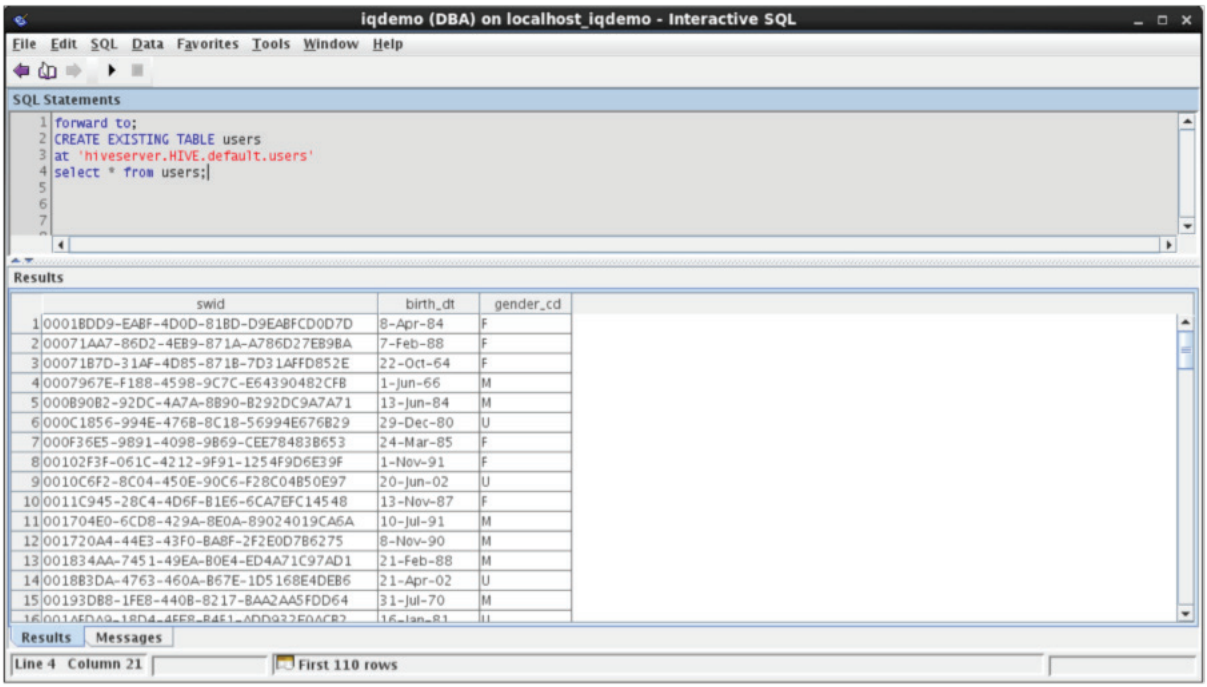
A proxy table can be created to access the data as well. Execute the following command:

```
Forward to;
Create existing table users
At ‘hiveserver.HIVE.default.users’
```

Note: The connection string ‘hiveserver.HIVE.default.users’ is generated by ‘servername.databasename.username.tablename’.

Now the users table is a proxy to the users table on the Hive server. You can access data from that table from IQ:

```
Select * from users;
```



Create Table

To create a table from IQ, run the following command:

```
Forward to;
Create table hive_t (bee int, nest int)
At ‘hiveserver.HIVE.default.hive_t’
```

Then you can access the table both from IQ and from the Hive server directly, with a proxy table already created.

WWW.SAP.COM