

Big SQL on Hadoop

Connecting to the IBM Big SQL Server and running SQL queries.

Contents

CONNECTING TO THE IBM BIG SQL SERVER AND RUNNING SQL QUERIES..... 3

1.1 GETTING STARTED 4

1.2 MANAGING THE BIG SQL STATUS..... 8

1.3 CONNECTING TO BIG SQL USING JSqSH 12

1.4 USING THE BIGINSIGHTS HOME TO RUN BIG SQL QUERIES 23

SUMMARY 30

Connecting to the IBM Big SQL Server and running SQL queries

For this exercise, you must have completed the docker image setup instructions in the *IBM Big Insights Quick Start Edition for Non-Production Environment (Docker image)* document that accompanies the docker image download. The file is titled `BigInsights_QSE_v4001_readme.pdf`.

IBM Big SQL is a component of the IBM InfoSphere BigInsights product. Before you can start working with your Hadoop data using Big SQL, you need to be able to connect to Big SQL. There are different methods for connecting to the IBM Big SQL Server. You will get to see how to some of these methods in this exercise.

First you will learn how to start, stop, and manage the Big SQL Server using two different methods: the command line and the Ambari tool. Then you will connect to Big SQL and run queries. The first method that you will use for Big SQL is JSqsh, pronounced “jay-skwish”. JSqsh is an open source CLI for JDBC applications such as Big SQL. This means that you can also use JSqsh for other JDBC applications as well. Another option to use with Big SQL is through the BigInsights console. There is a web based, Big SQL console where you can run your queries and get the results.

After completing this hands-on lab, you should be able to:

- Manage the Big SQL Server using the command line
- Manage the Big SQL Server using Ambari
- Connect to Big SQL using JSqsh to run Big SQL queries
- Use the BigInsights Home to run Big SQL queries

Allow 90 minutes to complete this section of lab.

Throughout this lab you will be using the following account login information:

When to use:	Username	Password
Log in from the command-line to accept the licenses	root	password
Log in to manage the Big SQL operations and services	bigsql	bigsql
Apache Ambari	admin	admin

1.1 Getting Started

It is assumed that you have downloaded the docker image and completed the setup and configuration requirements that match the environment of your particular operating system. Remember to complete the tasks that include configuring the image for Big SQL.

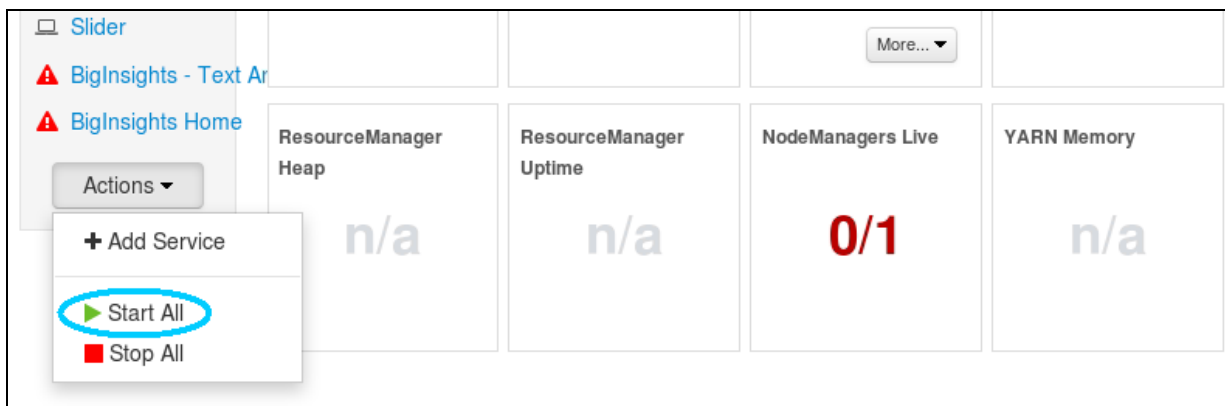
First, you want to start the BigInsights components.

- ___1. Open a Web browser and navigate to **<http://rvm.svl.ibm.com:8080>**, and sign in using the Ambari user id and password specified at the beginning of this document.

Notice that most of the BigInsights components listed at the left are in a Stopped state as indicated by the red, triangular warning icon.



- ___2. Scroll down if necessary, click the **Actions** button, and then click **Start All**.

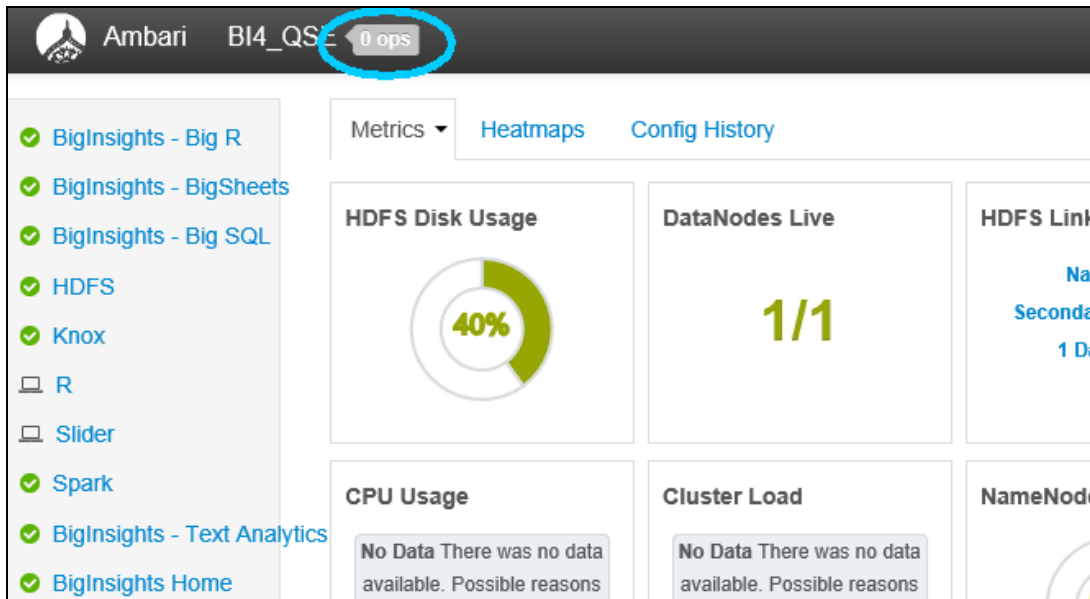


- ___3. Click **OK** on the *Confirmation* window.

Clicking *Stop All* would stop the components in a similar manner.

Note: Be sure to allow ample time for all the components to start. The first time you start the components and services, it may take approximately 30 minutes or even longer, depending on the physical resources on your machine.

- ___4. Periodically examine the background operations indicator at the top of the screen. It should show that 1 operation is running. When it updates to 0 ops as shown below, the Start All script is complete and the components should all be running as indicated by the green check mark icons on the left.



Note: If your icons still show red warning signs after the startup, it may be that the Ambari interface did not refresh properly, even though the details in the background operations show 100% and display a successful message. Feel free to click the Admin button at the top right of the window, and then click Sign out. You will be presented with the Ambari login screen. Log back in using the credentials at the beginning of this document and the component list should be updated with the correct, green check-mark icons.

- ___5. On your desktop, double-click the **Boot2Docker Start** shortcut that was created for you during your installation and configuration of the docker image.



The terminal window opens and will let you act with your environment, whether it is Windows, Mac, etc., as a Linux operating system.

```

MINGW32:/c/Users/byersj

Or run: `eval "$(<boot2docker shellinit)>"`

IP address of docker VM:
192.168.59.103

setting environment variables ...
Writing C:\Users\byersj\.boot2docker\certs\boot2docker-vm\ca.pem
Writing C:\Users\byersj\.boot2docker\certs\boot2docker-vm\cert.pem
Writing C:\Users\byersj\.boot2docker\certs\boot2docker-vm\key.pem
export DOCKER_HOST=tcp://192.168.59.103:2376
export DOCKER_CERT_PATH='C:\Users\byersj\.boot2docker\certs\boot2docker-vm'
export DOCKER_TLS_VERIFY=1

You can now use 'docker' directly, or 'boot2docker ssh' to log into the VM.
Welcome to Git (version 1.9.5-preview20150319)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

byersj@UOTTCHIN15 ~
$

```

To reference Windows locations, you might type `cd "/c/Program Files"`. To reference the Linux locations in the docker image, you need to run the `run.sh` script that is located in the directory where you unzipped your image files.

- __6. Change directories to the location where you unzipped your image and view the files in that directory. In the example shown in this document, the image was unzipped to the `/c/vm_images/Products/BIGSQL_W_DOCKER/` directory, so you would change directories to that location. Remember to use YOUR `.tar` file location in the following statement.

```
cd /c/vm_images/Products/BIGSQL_W_DOCKER
```

```
ls
```

```

byersj@UOTTCHIN15 /c/Program Files
$ cd /c/vm_images/Products/BIGSQL_W_DOCKER

byersj@UOTTCHIN15 /c/vm_images/Products/BIGSQL_W_DOCKER
$ ls
bi_v401_valueadd-bigsq1.tar  run.sh

byersj@UOTTCHIN15 /c/vm_images/Products/BIGSQL_W_DOCKER
$

```

__7. Run the run.sh script. Type:

```
./run.sh
```

The docker shell will open and you will be connected as the root user.

```
$ ./run.sh
Checking if the container exists for this image
Container biginsight_cont exists, checking the status of the container... If you
want to create new container, run the script with cleanup flag:
run.sh --cleanup
Container biginsight_cont in active state, connecting to the active container...

QSE docker container is active, opening a docker shell
Using python /usr/bin/python2.6
Starting ambari-server
Ambari Server running with 'root' privileges.
Organizing resource files at /var/lib/ambari-server/resources...
Server PID at: /var/run/ambari-server/ambari-server.pid
Server out at: /var/log/ambari-server/ambari-server.out
Server log at: /var/log/ambari-server/ambari-server.log
Waiting for server start.....
Ambari Server 'start' completed successfully.
Verifying Python version compatibility...
Using python /usr/bin/python2.6
Checking for previously running Ambari Agent...
ERROR: ambari-agent already running
Check /var/run/ambari-agent/ambari-agent.pid for PID.
[root@rvm /]#
```

Notice that you are logged in at the top of the file system at the / directory.

Now that are components are started you may move on to the next section.

1.2 Managing the Big SQL status


There is a method to manage just the Big SQL component. In this section, you will see how you can manage just the Big SQL server using both the command line, as well as the Ambari application.

There are two ways to start up Big SQL: through the terminal or by using Ambari. Big SQL should already be started, so you will first stop it using the terminal, and then you will use Ambari to actually start it.

Big SQL should still be running from your docker image configuration. You will check the status to make sure it is running. Then, you will stop it using the command line, and then start it using Ambari.

- __1. Once the terminal has been opened and you are connected, you will change to the bigsql user and navigate to the directory containing the bigsql commands. Type:

```
su bigsql
cd $BIGSQL_HOME/bin
ls
```

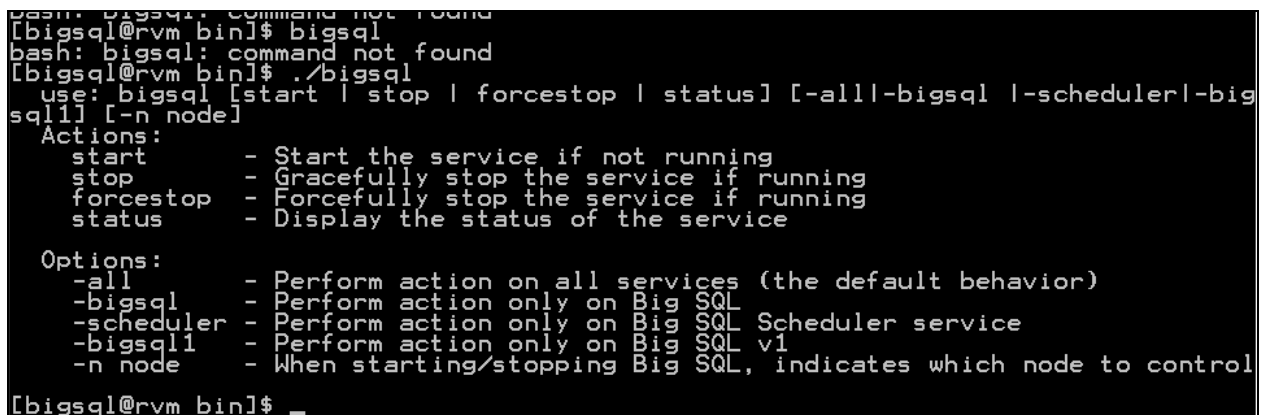


```
[root@rvm /]# su bigsql
[bigsql@rvm /]$ cd $BIGSQL_HOME/bin
[bigsql@rvm bin]$ ls
bigsql  bigsql-msg  bigsql-sched-ha.sh
```

Notice the bigsql file. It is the main command you will use to manage Big SQL.

- __2. At the prompt, type:

```
./bigsql
```



```
[bigsql@rvm bin]$ ./bigsql
bash: ./bigsql: command not found
[bigsql@rvm bin]$ ./bigsql
use: bigsql [start | stop | forcestop | status] [-all|-bigsql |-scheduler|-bigsql1] [-n node]
Actions:
  start      - Start the service if not running
  stop       - Gracefully stop the service if running
  forcestop  - Forcefully stop the service if running
  status     - Display the status of the service
Options:
  -all        - Perform action on all services (the default behavior)
  -bigsql     - Perform action only on Big SQL
  -scheduler  - Perform action only on Big SQL Scheduler service
  -bigsql1    - Perform action only on Big SQL v1
  -n node     - When starting/stopping Big SQL, indicates which node to control
[bigsql@rvm bin]$
```

Notice the various actions and options you can use with the bigsql command.

Typing `./bigsql start` will start the service, `./bigsql stop` will stop the service, and so forth.

- __3. Check the status of Big SQL. Type:

```
./bigsql status
```

```
[bigsql@rvm bin]$ ./bigsql status
SERVICE                HOSTNAME                NODE    PID  STATUS
Big SQL Scheduler       rvm.svl.ibm.com        -       27199 Available
Big SQL Master          rvm.svl.ibm.com        0       3723 Available
Big SQL Worker          rvm.svl.ibm.com        1      21356 Available
Big SQL v1              -                      -       - Not installed
[bigsql@rvm bin]$
```

If Big SQL is running it will display show a status of *Available*. Otherwise it will display a status of *Not running*. Notice that it is Available, since you started it in the previous section.

- __4. You want to stop the service and examine the status for the service when it is not running. Type:

```
./bigsql stop
```

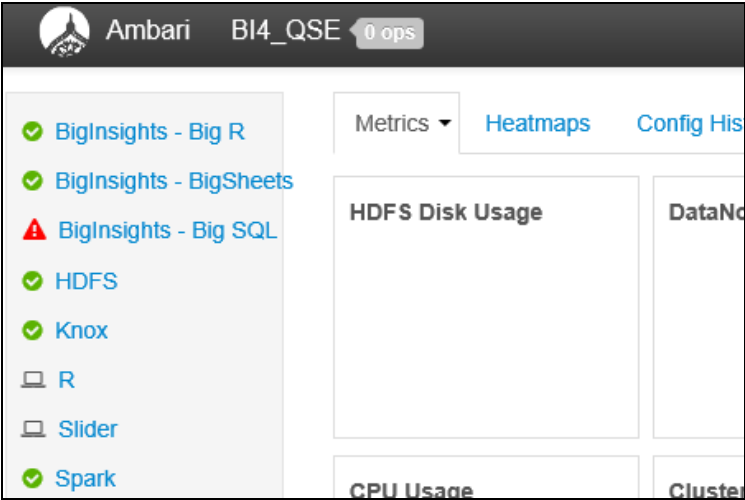
```
./bigsql status
```

```
[bigsql@rvm bin]$ ./bigsql stop
Stopping Big SQL v1      : SKIPPED (not installed)
Stopping Big SQL        : OK
Stopping Big SQL Scheduler : OK
[bigsql@rvm bin]$ ./bigsql status
SERVICE                HOSTNAME                NODE    PID  STATUS
Big SQL Scheduler       rvm.svl.ibm.com        -       - Not running
Big SQL Master          rvm.svl.ibm.com        0       - Not running
Big SQL Worker          rvm.svl.ibm.com        1       - Not running
Big SQL v1              -                      -       - Not installed
[bigsql@rvm bin]$
```

Notice that the status is now *Not running*.

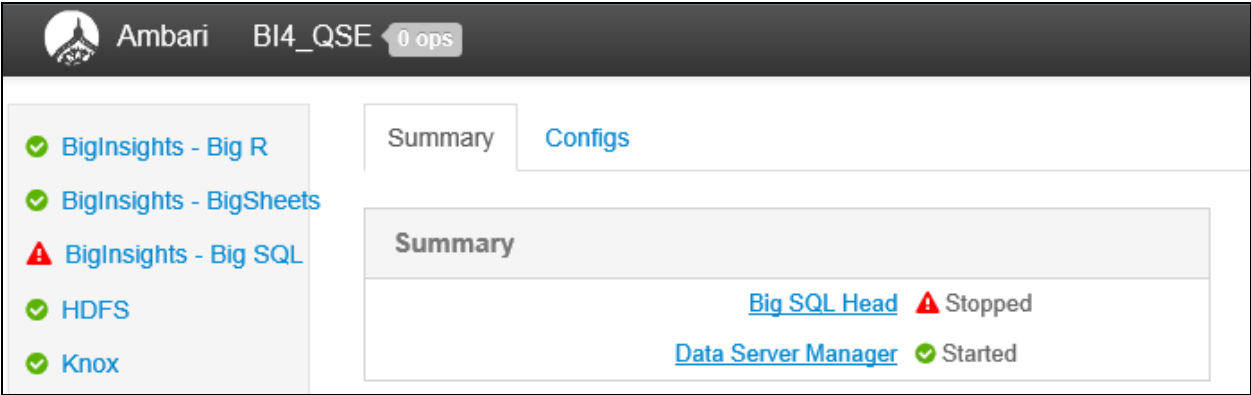
Next, let us actually start the Big SQL using the Ambari tool.

- __5. Restore the Web browser that is open to Ambari.

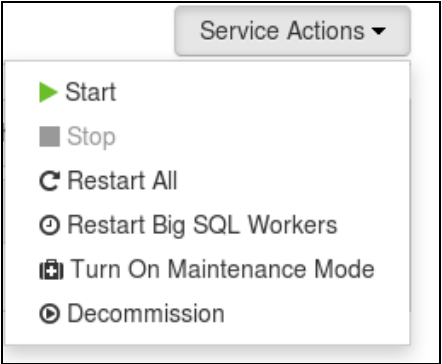


Notice that the Big SQL component is no longer running.

- __6. In Ambari, click the **Services** tab, and from the components list on the left, click **BigInsights - Big SQL**.



- __7. Click the **Service Actions** button at the top right of the window and notice the options for the Big SQL service.



- __8. You want to start the Big SQL service, so click **Start**, and then click **OK** on the Confirmation window.

After a short time, the service starts.

- __9. Once the service has started, restore the terminal window and type:

```
./bigsql status
```

```
[bigsql@rvm bin]$ ./bigsql status
SERVICE                                HOSTNAME                                NODE    PID STATUS
Big SQL Scheduler                       rvm.svl.ibm.com                        -       30888 Available
Big SQL Master                         rvm.svl.ibm.com                        0        7487 Available
Big SQL Worker                         rvm.svl.ibm.com                        1      25855 Available
Big SQL v1                             -                                     -         - Not installed
[bigsql@rvm bin]$
```

- __10. At the command prompt, type exit to exit the bigsql user authentication and work as root again.

```
[bigsql@rvm bin]$ exit
exit
[root@rvm /]#
```

- __11. At the command prompt, type exit to exit the docker shell.

You should still be in the docker container, but now positioned in your home operating system file system.

```
[root@rvm /]# exit
exit
byersj@VOTTCHIN15 /c/vm_images/Products/BIGSQL_W_DOCKER
$
```

Leave the window open for the next section.

1.3 Connecting to Big SQL using JSqsh

In this section, you will work with Big SQL using JSqsh. You will set up a Big SQL connection, configure the repository, and run a few simple queries to show the interaction with JSqsh.

- __1. In the open docker container window, at the prompt, type:

```
boot2docker ssh
```

```
docker exec -it biginsight_cont bash
```



```
byers.j@VOTTCHIN15 /c/vm_images/Products/BIGSQL_W_DOCKER
$ boot2docker ssh
##
## ## ##
## ## ## ## ##
=====
      o
     / \
    /   \
   /     \
  /       \
 /         \
/           \
[boot2docker]
Boot2Docker version 1.7.1, build master : c202798 - Wed Jul 15 00:16:02 UTC 2015
Docker version 1.7.1, build 786b29d
docker@boot2docker:~$ docker exec -it biginsight_cont bash
[root@rvn /]#
```

- __2. Navigate to the jsqsh directory. Type the following:

```
cd /usr/ibmpacks/common-utils/jsqsh/2.14/bin
```

- __3. Create a Big SQL connection for JSqsh. Type the following:

```
./jsqsh --setup
```

- __4. Start the connection wizard. Type in the letter **c**.

JSQSH SETUP WIZARD

Welcome to the jsqsh setup wizard! This wizard provides a (crude) menu driven interface for managing several jsqsh configuration files. These files are all located in \$HOME/.jsqsh, and the name of the file being edited by a given screen will be indicated on the title of the screen

Note that many wizard screens require a relative large console screen size, so you may want to resize your screen now.

(C)onnection management wizard

The connection management wizard allows you to define named connections using any JDBC driver that jsqsh recognizes. Once defined, jsqsh only needs the connection name in order to establish a JDBC connection

(D)river management wizard

The driver management wizard allows you to introduce new JDBC drivers to jsqsh, or to edit the definition of an existing driver. The most common activity here is to provide the classpath for a given JDBC driver

Choose (Q)uit, (C)onnection wizard, or (D)river wizard: c_

- __5. Notice the available connection for bigsql. In this lab, you will be working with the bigsql database on port 51000.

JSQSH CONNECTION WIZARD - (edits \$HOME/.jsqsh/connections.xml)
The following connections are currently defined:

	Name	Driver	Host	Port
1	bigsql	db2	rvm.svl.ibm.com	51000

Enter a connection number above to edit the connection, or:
(B)ack, (Q)uit, or (A)dd connection:

- __6. Select the Big SQL driver. Type the number **1** and then press **Enter**.
- __7. You can see that there are a number of variables already defined. Make sure that the values to your variables are the same as shown in the screen capture. For the password variable, specify the password so that you do not need to provide it every time you connect to the Big SQL connection. Select the password variable. Type in the number **5**.

```

JSQSH CONNECTION WIZARD - (edits $HOME/.jsqsh/connections.xml)
The following configuration properties are supported by this driver.
    Connection name : bigsql
    Driver : IBM Data Server (DB2, Informix, Big SQL)
    JDBC URL : jdbc:db2://${server}:${port}/${db}
Connection URL Variables
-----
1          db : BIGSQL
2          port : 51000
3          server : rvm.svl.ibm.com
4          user : bigsql
5          password : *****
6          Autoconnect : false
JDBC Driver Properties
-----
None
Enter a number to change a given configuration property, or
(T)est, (D)elele, (B)ack, (Q)uit, Add (P)roperty, or (S)ave: 5_

```

- __8. When prompted, enter in the password: **bigsql** and press **Enter**. You will see asterisks in place of the password:

```

Please enter a new value:
password: *****_

```

- __9. Make sure you have entered in the correct password by performing the *Test* operation. Type the letter **t**, to run the test.

```

Enter a number to change a given configuration property, or
(T)est, (D)elele, (B)ack, (Q)uit, Add (P)roperty, or (S)ave: t
Attempting connection...
Succeeded!
Hit enter to continue:_

```

- __10. If the connection test was not successful, make sure you update the password variable and rerun the test. If the test was successful, press **Enter** as directed, and then save the connection profile. Type in the letters **s** to save and **y** to confirm.

```

Enter a number to change a given configuration property, or
(T)est, (D)elele, (B)ack, (Q)uit, Add (P)roperty, or (S)ave: s
Are you sure (Y/N)? y_

```

- __11. Quit the JSqsh connection wizard by typing in the letter **q**.

- __12. Exit JSqsh by typing **quit**.

You are restored to the root user located in the /bin subdirectory.

- __13. Restart JSqsh specifying the **bigsql** connection that you just set up.

```
./jsqsh bigsql
```

Because you had saved the password in the connection profile, you will not be prompted to provide the password again.

```
[root@rvm bin]# ./jsqsh bigsql
JSqsh Release 2.14, Copyright (C) 2007-2015, Scott C. Gray
Type \help for available help topics. Using JLine.
[rvm.svl.ibm.com][bigsql] 1> _
```

- __14. To get help from JSqsh, type:

```
\help
```

```
[rvm.svl.ibm.com][bigsql] 1> \help
Available help categories. Use "\help <category>" to display topics within that
category
+-----+-----+
| Category | Description |
+-----+-----+
| commands | Help on all available commands |
| vars      | Help on all available configuration variables |
| topics    | General help topics for jsqsh |
+-----+-----+
[rvm.svl.ibm.com][bigsql] 1> _
```

- __15. To see the list of commands to use within JSqsh, type:

```
\help commands
```

Notice that the jsqsh more operator is automatically invoked for this command. You can press the space bar to scroll through the output, or press *q* to quit and go to the end of the output.

- __16. To see the schemas, type in:

```
\show schemas
```

```
[rvm.svl.ibm.com][bigsql] 1> \show schemas
```

TABLE_SCHEM	TABLE_CATALOG
BIGSQL	[NULL]
DEFAULT	[NULL]
NULLID	[NULL]
SQLJ	[NULL]
SYSCAT	[NULL]
SYSFUN	[NULL]
SYSHADOOP	[NULL]
SYSIBM	[NULL]
SYSIBMADM	[NULL]
SYSIBMINTERNAL	[NULL]
SYSIBMTS	[NULL]
SYSPROC	[NULL]
SYSPUBLIC	[NULL]
SYSSTAT	[NULL]
SYSTOOLS	[NULL]

```
[rvm.svl.ibm.com][bigsql] 1>
```

- __17. To display essential information about all available tables, type in:

```
\show tables -e
```

Notice that no tables exist for the bigsql schema. Next you want to create a table.

- __18. Create a simple Hadoop table using Big SQL. Copy and paste or type in the following:

```
create hadoop table test1 (col1 int, col2 varchar(5));
```

Because you did not specify a schema name for the table, it was created under your default schema, which is your username, bigsql. The statement above is equivalent to:

```
create hadoop table bigsql.test1 (col1 int, col2 varchar(5));
```

- __19. Check that the table has been created. Type:

```
\tables
```

```
[rvm.svl.ibm.com][bigsql] 1> \tables
```

TABLE_SCHEM	TABLE_NAME	TABLE_TYPE
BIGSQL	TEST1	TABLE

- __20. To display just user tables (avoid views and system tables) type:

```
\tables -T user
```

With this command, you may see tables from other users as well but you may not have the privilege to query them.

- __21. To see just your tables, namely the bigsql schema tables, type:

```
\tables -s BIGSQL
```

Pay attention that the login name provided here is in uppercase. That is because the system changes all names to uppercase. The search is case sensitive, so if you do not provide the login name in uppercase, you will not see what you are expecting. The screenshot shows that if you queried with the lowercase, you will not see the *test1* table. When you query using uppercase BIADMIN, you see the *test1* table.

```
[rvm.svl.ibm.com][bigsql] 1> \tables -s bigsql
+-----+-----+-----+
| TABLE_SCHEM | TABLE_NAME | TABLE_TYPE |
+-----+-----+-----+
[rvm.svl.ibm.com][bigsql] 1> \tables -s BIGSQL
+-----+-----+-----+
| TABLE_SCHEM | TABLE_NAME | TABLE_TYPE |
+-----+-----+-----+
| BIGSQL       | TEST1       | TABLE      |
+-----+-----+-----+
[rvm.svl.ibm.com][bigsql] 1> _
```

- __22. Insert a row into the test1 table. Type:

```
insert into test1 values (1, 'one');
```

```
[rvm.svl.ibm.com][bigsql] 1> insert into test1 values (1, 'one');
1 row affected (total: 1m1.76s)
[rvm.svl.ibm.com][bigsql] 1> _
```

It is important to remind you that the INSERT statement should only be used for testing purposes. The INSERT operation is not parallelized on the cluster; so therefore, it is very inefficient for loading large amounts of data. It is recommended that you use one of the bulk loading operators that you will see in the next exercise. Those operators are parallelized and optimized so your queries will yield much better performance in a production environment.

- __23. Look at the *test1* table metadata, type:

```
\describe BIGSQL.TEST1
```

```
[rvmsv1.ibm.com][bigsql] 1> \describe BIGSQL.TEST1
```

TABLE_SCHEM	COLUMN_NAME	TYPE_NAME	COLUMN_SIZE	DECIMAL_DIGITS	IS_NULLABLE
BIGSQL	COL1	INTEGER	10	0	YES
BIGSQL	COL2	VARCHAR	5	[NULL]	YES

```
[rvmsv1.ibm.com][bigsql] 1>
```

Notice again that you have use uppercase for the schema and the table names because those values are folded to upper case in the system catalog tables.

- __24. To see the inserted row, type in:

```
select * from test1;
```

```
[rvmsv1.ibm.com][bigsql] 1> select * from test1;
+-----+-----+
| COL1 | COL2 |
+-----+-----+
| 1 | one |
+-----+-----+
1 row in results(first row: 8.66s; total: 8.68s)
[rvmsv1.ibm.com][bigsql] 1>
```

In queries, you do not need to uppercase any of the names.

- __25. Issue a query that restricts the number of rows returned to 5. For example, select the first 5 rows from the syscat.tables:

```
select tabschema, tabname from syscat.tables fetch first 5 rows only;
```

```
[rvmsv1.ibm.com][bigsql] 1> select tabschema, tabname from syscat.tables fetch
first 5 rows only;
+-----+-----+
| TABSCHEMA | TABNAME |
+-----+-----+
| BIGSQL | TEST1 |
| SYSCAT | ATTRIBUTES |
| SYSCAT | AUDITPOLICIES |
| SYSCAT | AUDITUSE |
| SYSCAT | BUFFERPOOLDBPARTITIONS |
+-----+-----+
5 rows in results(first row: 0.7s; total: 0.9s)
[rvmsv1.ibm.com][bigsql] 1>
```

Restricting the number of rows returned by a query is good for development when working with large volume of data.

- __26. JSqsh has a few useful commands that should be considered. You can review the history of the commands recently executed in the JSqsh shell. Type:

```
\history
```

If you notice more commands listed than you typed into the prompt, remember that you are using an image that was prepared and tested for you. It is possible that you will find those commands in your history output.

- __27. To recall a query from the *history*, for example, to recall statement 6, type in **!6**. This will bring the query to the current command line. Then you just need to add a ; (semi-colon) to the final line and press **Enter** to run the statement.

- __28. To recall a previously executed statement, type in **!!** (two exclamation points, without spaces). Then add a ; (semi-colon) at the end to run the statement.

- __29. JSqsh also has the ability to pipe outputs to an external program. Pipe the output of the next statement to the *more* operator.

```
select tabschema, tablename from syscat.tables  
  
go | more
```

Note that because the first line did not have a semi-colon at the end, the statement did not run. That is because the default terminator for Big SQL is a semi-colon. The *go* command on the second line is actually what triggers Big SQL to run the statement. In fact, under the covers, the semi-colon at the end is the short cut for the JSqsh *go* command.

- __30. Experiment with JSqsh's ability to redirect output to a local file rather than the console display. Enter the following lines on the command shell, adjusting the path information as needed for your environment.

```
select tabschema, colname, colno, typename, length  
from syscat.columns  
where tabschema = USER and tablename= 'TEST1'  
go > /tmp/test1.out
```

You want to view the output by opening a second terminal window.

- __31. On your desktop, double-click the **Boot2Docker Start** shortcut window, navigate to the directory containing the `run.sh` script, and start the `run.sh`. Type:

```
cd /c/vm_images/Products/BIGSQL_W_DOCKER
```

(Remember to substitute the local directory for your environment in the previous command.)

```
./run.sh
```

You will be logged in as root and located at the top / directory.

- __32. Next navigate to the output directory from your JSqsh statement.

```
cd /tmp
```

```
more test1.out
```

```
[root@rvn tmp]# more test1.out
+-----+
| TABSCHEMA | COLNAME | COLNO | TYPENAME | LENGTH |
+-----+
| BIGSQL    | COL1    | 0     | INTEGER  | 4       |
| BIGSQL    | COL2    | 1     | VARCHAR  | 5       |
+-----+
```

In a production environment, you are likely to have your SQL statements in script files. Maintaining SQL script files can be quite handy for repeating executing various queries. Next, you will create a new file with multiple SQL queries.

- __33. Create a new `test1.sql` file. Type:

```
vi test1.sql
```

- __34. Press **i** to insert text, and then type or paste the following into the `test1.sql` file.

```
select tabschema, tabname from syscat.tables fetch first 5 rows only;
```

```
select tabschema, colname, colno, typename, length
```

```
from syscat.columns
```

```
fetch first 10 rows only;
```

__35. Save the file and exit. Type:

```
:wq
```

__36. Invoke the SQL script (test1.sql). Using the same command line, type or paste in:

```
/usr/ibmpacks/common-utils/jsqsh/2.14/bin/jsqsh bigsql < ./test1.sql
```

```
[root@rvml tmp]# /usr/ibmpacks/common-utils/jsqsh/2.14/bin/jsqsh bigsql < ./test1
.sql
JSqsh Release 2.14, Copyright (C) 2007-2015, Scott C. Gray
Type \help for available help topics. Using JLine.
[rvml.svl.ibm.com][bigsql] 1> select tabschema, tabname from syscat.tables fetch
first 5 rows only;
+-----+-----+
| TABSCHEMA | TABNAME |
+-----+-----+
| BIGSQL    | TEST1   |
| SYSCAT    | ATTRIBUTES |
| SYSCAT    | AUDITPOLICIES |
| SYSCAT    | AUDITUSE |
| SYSCAT    | BUFFERPOOLDBPARTITIONS |
+-----+-----+
5 rows in results(first row: 0.31s; total: 0.35s)
[rvml.svl.ibm.com][bigsql] 1>
[rvml.svl.ibm.com][bigsql] 2> select tabschema, colname, colno, typename, length
[rvml.svl.ibm.com][bigsql] 3> from syscat.columns
[rvml.svl.ibm.com][bigsql] 4> fetch first 10 rows only;
+-----+-----+-----+-----+-----+
| TABSCHEMA | COLNAME | COLNO | TYPENAME | LENGTH |
+-----+-----+-----+-----+-----+
| SYSIBM   | NAME    | 0     | VARCHAR  | 128    |
| SYSIBM   | CREATOR | 1     | VARCHAR  | 128    |
| SYSIBM   | TYPE    | 2     | CHARACTER | 1      |
| SYSIBM   | CTIME   | 3     | TIMESTAMP | 10     |
| SYSIBM   | REMARKS | 4     | VARCHAR  | 254    |
| SYSIBM   | PACKED_DESC | 5     | BLOB     | 133169152 |
| SYSIBM   | VIEW_DESC | 6     | BLOB     | 4190000 |
| SYSIBM   | COLCOUNT | 7     | SMALLINT | 2      |
| SYSIBM   | FID     | 8     | SMALLINT | 2      |
| SYSIBM   | TID     | 9     | SMALLINT | 2      |
+-----+-----+-----+-----+-----+
10 rows in results(first row: 0.16s; total: 0.18s)
[rvml.svl.ibm.com][bigsql] 1>
[root@rvml tmp]# _
```

__37. Inspect the output and then delete the files you created in the /user/bigsql directory. Type:

```
rm test*
```

Press **y** and then press **Enter** to confirm each delete.

```
[root@rvml tmp]# rm test*
rm: remove regular file 'test1.out'? y
rm: remove regular file 'test1.sql'? y
[root@rvml tmp]# _
```

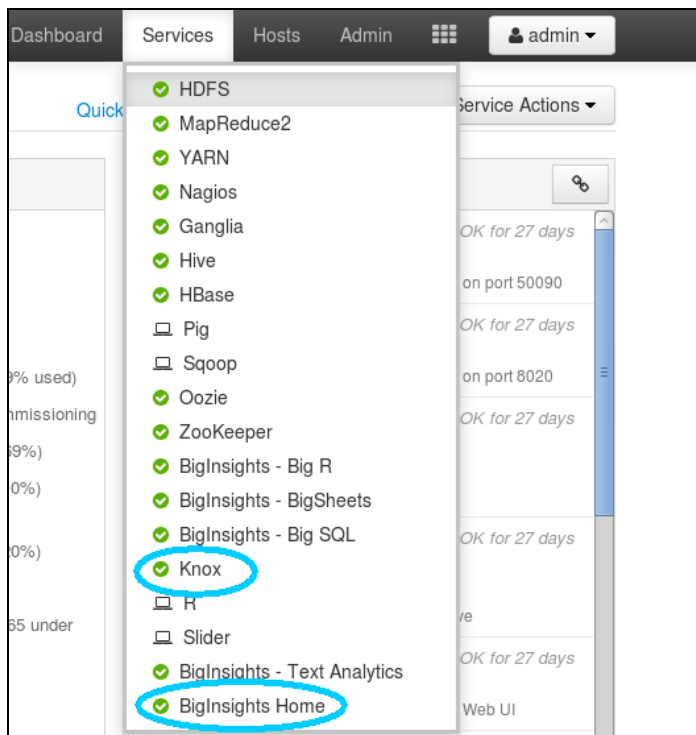
__38. Type **exit** to quit the docker shell, and then type **exit** again to close the container window.

- ___39. Close any opened windows. For now, you will leave the TEST1 table with one record intact for use in the next section.

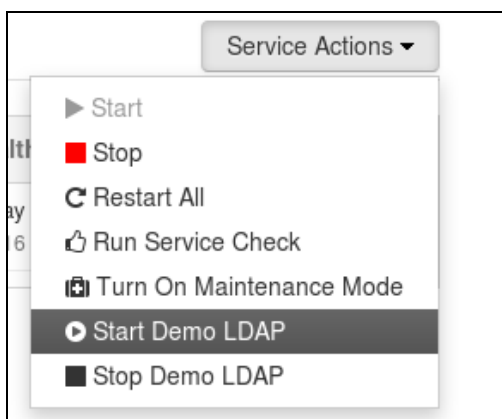
1.4 Using the BigInsights Home to run Big SQL queries

In this section, you will work with Big SQL using the BigInsights Home. First you will start the Demo LDAP that is needed to authenticate through BigInsights Home.

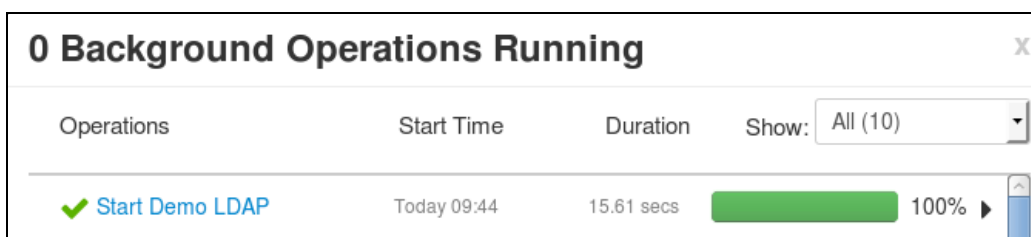
- ___1. Launch Ambari by starting a Web browser and navigating to <http://rvm.svl.ibm.com>.
- ___2. Log in using the Apache Ambari credentials specified at the beginning of this document.
- ___3. Click the **Services** tab, and then hover the mouse cursor over the Services tab to see the various services and their status. Ensure that the *Knox* and *BigInsights Home* services have been started.



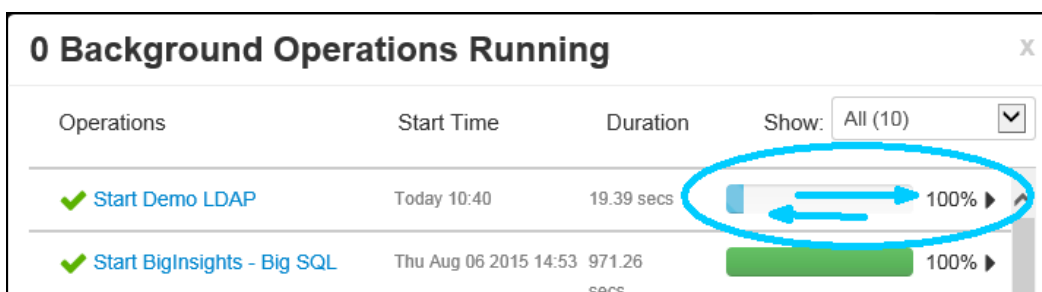
- ___4. Click the **Knox** component.
- ___5. Click the **Service Actions** button, click **Start Demo LDAP**, and then click **OK** on the Confirmation window.



- __6. Once the Background Operations window indicates that *Start Demo LDAP* is 100% started, click **OK**.



If the Start Demo LDAP percentage is 100%, the process has started. It may be that it reads 100%, but the colored indicator shows that it is only partially started, as in the following capture. This may have to do with the refresh operations of Ambari and is not a big concern. If it bothers you, you can log out of Ambari, log in again, and then examine the background operations.

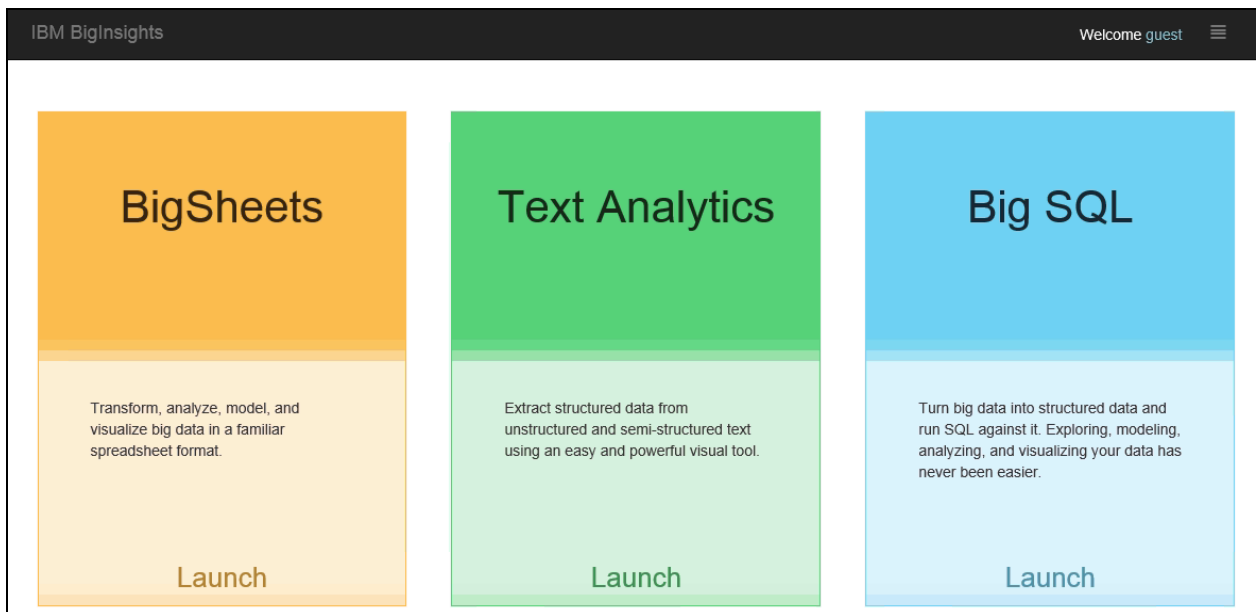


- __7. Open a new tab in the Web browser and navigate to <https://rvm.svl.ibm.com:8443/gateway/default/BigInsightsWeb/index.html>.

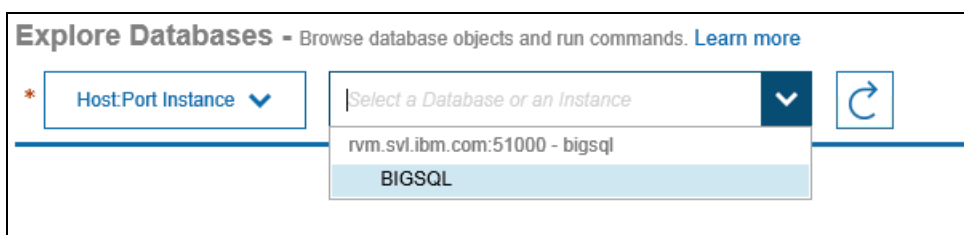
If a warning is displayed indicating that there the security certificate is not trusted, proceed in spite of the warning by clicking Continue to this website (not recommended).

[Continue to this website \(not recommended\).](#)

- ___8. If prompted, log in as guest/guest-password. If you have signed in previously during this session, the login screen may be bypassed..

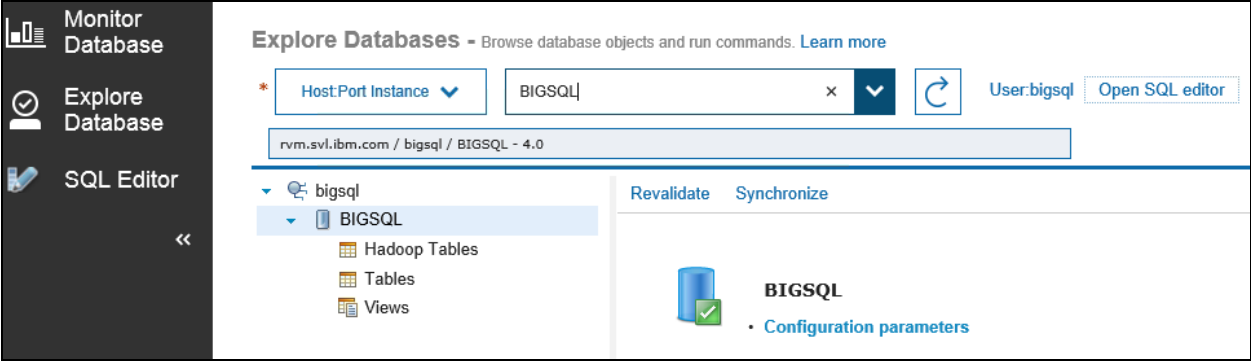


- ___9. Under Big SQL, click **Launch**.
- ___10. In the pane on the left, click **Explore Database**, and from the *Select a database or instance* box drop-down list, click **BIGSQL**.

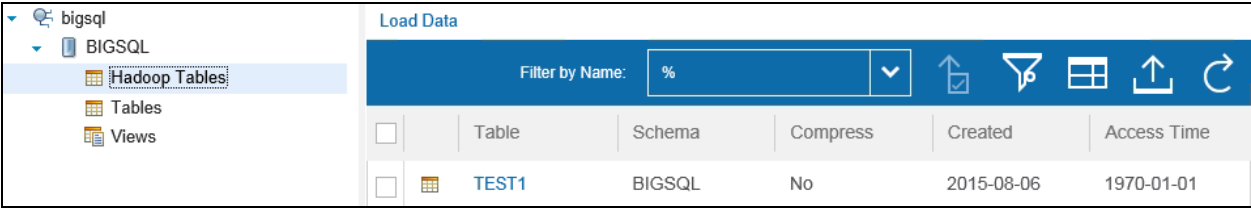


- ___11. In the *Connect: BIGSQL* window, enter **bigsql** for the User ID and **bigsql** for the password, and then click **OK**.

Allow ample time for the screen to refresh. The bigsql explorer tree is displayed.

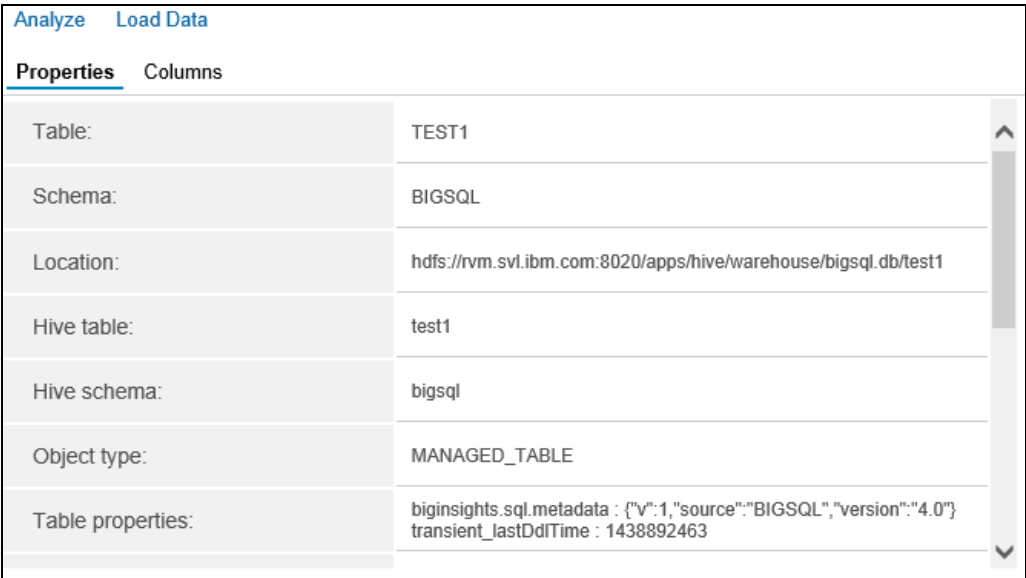



__12. In the explorer pane on the left, click **Hadoop Tables**.

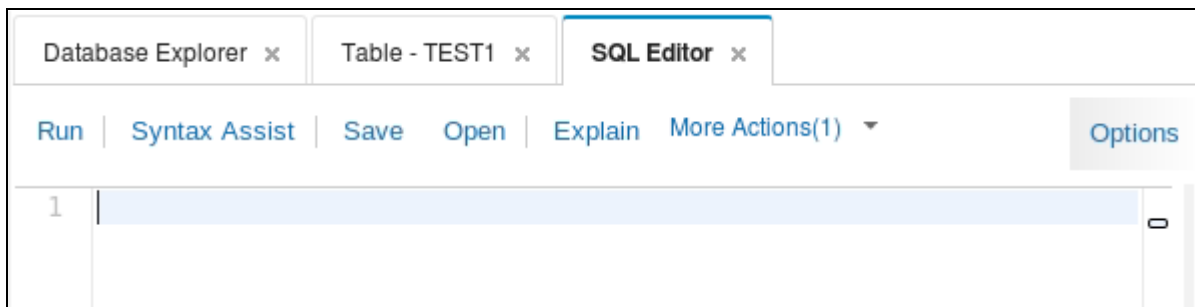


Notice that the pane on the right refreshes to display the Hadoop table you created previously in this exercise named TEST1.

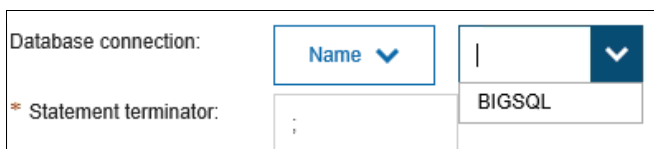
__13. In the *Table* column, click **TEST1** and then scroll up and down to view all the metadata for the TEST1 table.




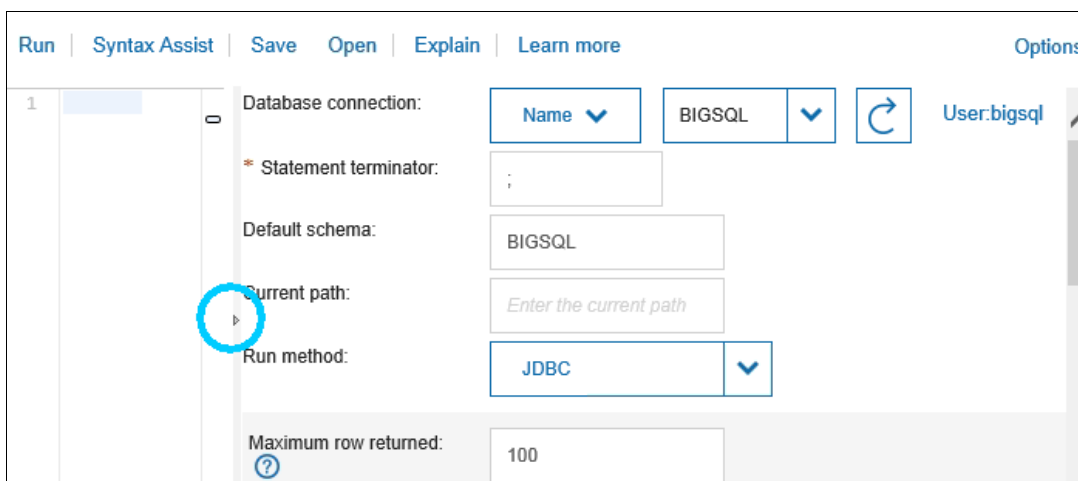
- __14. Click **Open SQL editor** , and notice the new *SQL Editor* tab that opens on the right.



- __15. Click **Options**, and then next to *Database connection*, select **BIGSQL** from the drop-down list.

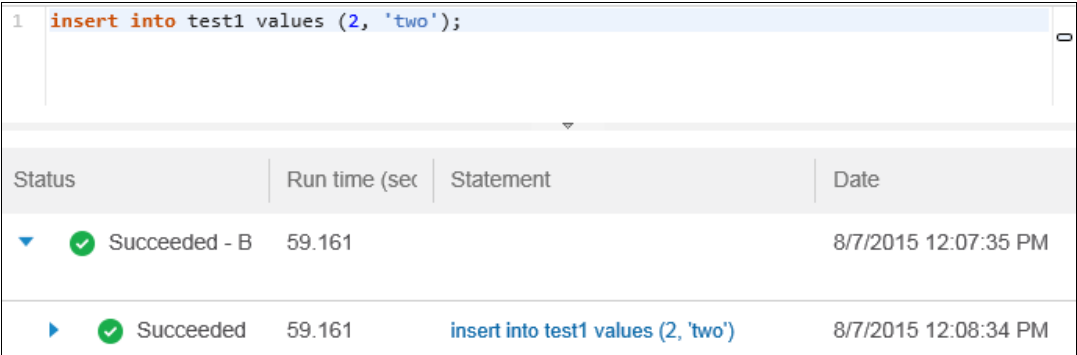


- __16. In the *Default schema* box, type **BIGSQL**, and then at the left side of the Options pane, click the arrow  to collapse the option selections.



- __17. Insert another row of data into the *TEST1* table that you created earlier. Type in:
- ```
insert into test1 values (2, 'two');
```

\_\_18. Click **Run**.



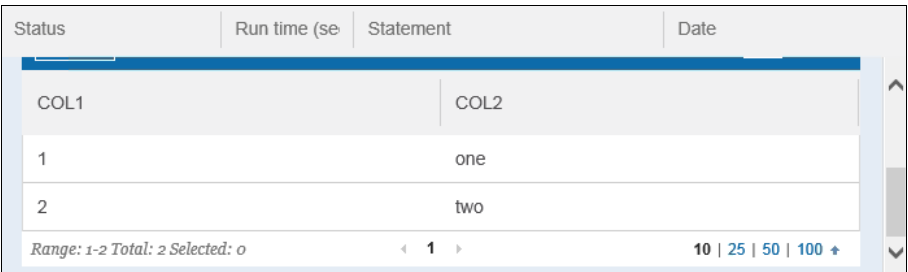
The screenshot shows a SQL Editor window with a single line of code: `insert into test1 values (2, 'two');`. Below the editor is a table showing the execution status of the statement.

| Status           | Run time (sec) | Statement                                        | Date                 |
|------------------|----------------|--------------------------------------------------|----------------------|
| ▼  Succeeded - B | 59.161         |                                                  | 8/7/2015 12:07:35 PM |
| ▶  Succeeded     | 59.161         | <code>insert into test1 values (2, 'two')</code> | 8/7/2015 12:08:34 PM |

\_\_19. Query the table to see the results. Highlight the insert statement you just executed, press **Delete**, and then type in:

```
select * from test1;
```

\_\_20. Click **Run**, and then in the *Status* pane, scroll down to view your results.

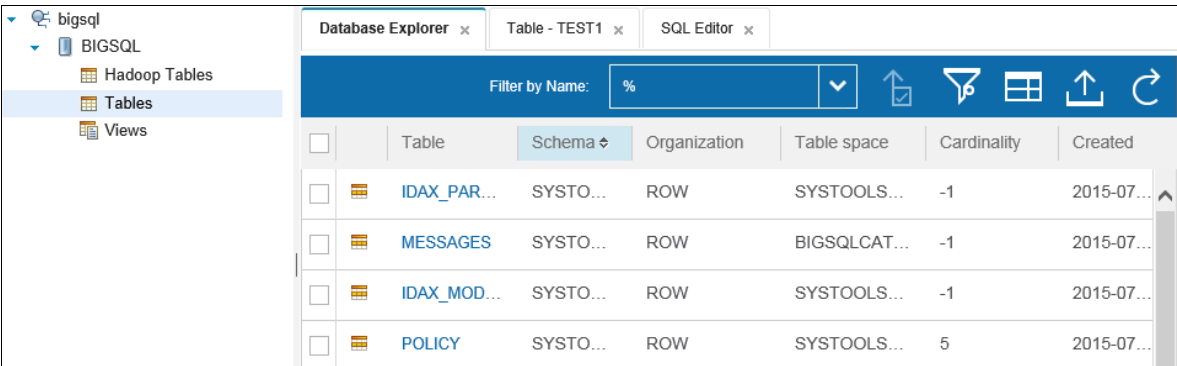


The screenshot shows the Status pane with a table of results. The table has two columns: COL1 and COL2. The results are as follows:

| COL1 | COL2 |
|------|------|
| 1    | one  |
| 2    | two  |

Below the table, there is a status bar that reads: "Range: 1-2 Total: 2 Selected: 0".

\_\_21. In the explorer-style pane on the left, click **Tables** and notice that non-Hadoop tables are displayed in the Database Explorer tab.



The screenshot shows the Database Explorer pane with a list of tables. The left pane shows the tree structure with "Tables" selected. The right pane shows a table with the following columns: Table, Schema, Organization, Table space, Cardinality, and Created.

| Table       | Schema   | Organization | Table space  | Cardinality | Created    |
|-------------|----------|--------------|--------------|-------------|------------|
| IDAX_PAR... | SYSTO... | ROW          | SYSTOOLS...  | -1          | 2015-07... |
| MESSAGES    | SYSTO... | ROW          | BIGSQLCAT... | -1          | 2015-07... |
| IDAX_MOD... | SYSTO... | ROW          | SYSTOOLS...  | -1          | 2015-07... |
| POLICY      | SYSTO... | ROW          | SYSTOOLS...  | 5           | 2015-07... |

- \_\_22. Click the **SQL Editor** tab, delete the existing query, and execute two statements at once. On separate lines then type:

```
insert into test1 values (3, 'three');
```

```
insert into test1 values (4, 'four');
```

- \_\_23. If you input more than one statement, each terminated with a semi-colon, then there will be multiple results displayed for each of the statements. Click **Run** and scroll down to view the results.

| Status           | Run time (sec) | Statement                             | Date                 |
|------------------|----------------|---------------------------------------|----------------------|
| ▼  Succeeded - B | 33.657         |                                       | 8/7/2015 12:14:41 PM |
| ▶  Succeeded     | 26.583         | insert into test1 values (3, 'three') | 8/7/2015 12:15:08 PM |
| ▶  Succeeded     | 7.074          | insert into test1 values (4, 'four')  | 8/7/2015 12:15:15 PM |

- \_\_24. Highlight and delete the existing SQL statements, and then query to see the results in reverse order. Type:

```
select col1, col2 from test1 order by col1 desc;
```

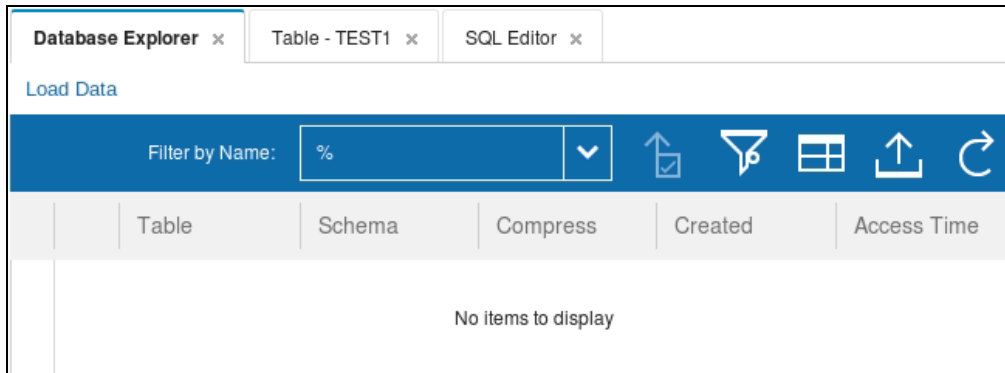
- \_\_25. Click **Run**, and then scroll down to view the results of all your statements.

| Status | Run time (se | Statement |
|--------|--------------|-----------|
| COL1   |              | COL2      |
| 4      |              | four      |
| 3      |              | three     |
| 2      |              | two       |
| 1      |              | one       |

- \_\_26. Clean up your table for subsequent exercises. Execute the drop table statement.

```
drop table test1;
```

- \_\_\_27. Verify that the TEST1 table was removed. Click the **Database Explorer** tab, and then from the explorer tree, click **Hadoop Tables**.



- \_\_\_28. Close all open windows and applications.

## Summary

Having completed this exercise, you should now be able to start using Big SQL and interact with Big SQL using JSqsh and the BigInsights console.

## NOTES

## NOTES

[illegible]







---

© Copyright IBM Corporation 2013.

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and [ibm.com](http://ibm.com) are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).



Please Recycle

---