

Demonstration 1

Integrating with Big SQL

At the end of this demonstration, you should be able to:

- Use Big SQL tables with BigSheets

Demonstration 1: Integrating with Big SQL

Demonstration 1: Integrating with Big SQL

Purpose:

The integration with Big SQL allows you to perform additional analytical queries against the data. It extends upon the capabilities of BigSheets allowing you to use common SQL queries to get insight from the data. You will use Big SQL tables with BigSheets.

User/Password: **biadmin/biadmin**
 Root/dalvm3
 Service Password: **ibm2blue**

Task 1. Loading the test data into the HDFS.

You are going to use the **blogs-data.txt** file for this demonstration. Do the following steps to load the file into the HDFS. The data in the blogs-data.txt file comes from blogs that reference the term IBM Watson.

In this demonstration you are going to turn that text data into a BigSheets workbook, and then use the functions in BigSheets to format the data into something that is easier to understand.

To examine the blogs data in the blogs-data.txt file, you will create a workbook and use that data for a new Big SQL table. This demonstration introduces a way of creating tables from data that you analyze by using BigSheets and a TSV reader format and a JSON Array format.

1. Open up a terminal and enter in this command:

```
hdfs dfs -mkdir /user/biadmin/Watson
```

1. Use this command to upload the blogs-data.txt

```
hdfs dfs -put /home/biadmin/labfiles/bigsheets/blogs-  
data.txt /user/biadmin/Watson
```

2. Do another listing to confirm that the file has been loaded:

```
hdfs dfs -ls /user/biadmin/Watson
```

3. Change the permissions on the Watson directory:

```
hdfs dfs -chmod 777 /user/biadmin/Watson
```

This material is meant for IBM Academic Initiative use only. NOT FOR RESALE

Task 2. Start up the Big SQL service.

1. Inside the Ambari console, ensure **BigInsights - Big SQL** has started. If not, start it up.

Big SQL requires that the monitoring utility package is started as well.

2. Open up a terminal to start the monitoring utility package.
3. Switch to the root user, by typing `su -` and then type the password `dalvm3`.
4. Change directory to the following path:

```
cd /usr/ibmpacks/bigsql/4.0/dsm/1.1/ibm-datasrvrmgr/bin/
```

You will run the `dsmKnoxSetup` script as the root user.

5. Run the script `/dsmKnoxSetup.sh -knoxHost <knox-host>`

where `<knox-host>` is the host where the Knox gateway is running. In our case, it would be *ibmclass.localdomain*

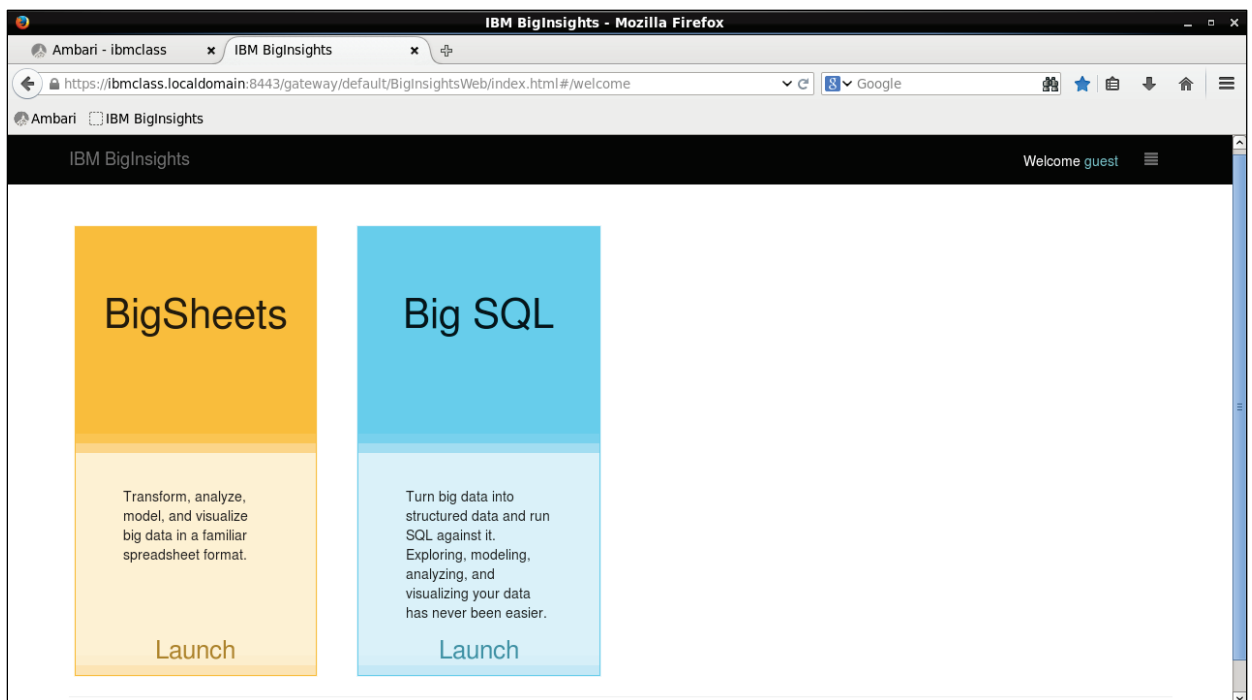
```
./dsmKnoxSetup.sh -knoxHost ibmclass.localdomain
```

2. When prompted to continue running with the above value, select **1**.

Remember, when you restart the Knox server, you will have to run the `dsmKnoxSetup` script again.





You will now be able to access Big SQL via the **BigInsights Home** page.

6. Check to see that **Big SQL** is available.



This material is meant for IBM Academic Initiative use only. NOT FOR RESALE

Task 3. Create a BigSheets Workbook.

1. Click the **BigSheets** link from the **BigInsights** Home.
This takes you to the list of all workbooks.
2. Click **New Workbook** .
3. In the **Name** field type **BlogsData**.
4. Drill down to **/user/biadmin**, expand the **Watson** directory and then select the **blogs-data.txt** file.
5. Click **Edit workbook** .
6. From the **Select a reader** drop-down, select **JSON Array** reader.
7. Click **Set Reader** .
8. Now with the data formatted properly, scroll down (if you have to) and click **Save workbook** .
9. You do not need all the columns in your Big SQL table, so you will remove some now. First you need to create a child workbook.
10. Click **Build new workbook**.
11. Rename the workbook by clicking **Edit workbook name** and then type **BlogsDataRevised**.
12. Remove multiple columns by following these steps:
 - a. Click the down arrow in any column heading and select **Organize Columns**
 - b. Click the **X** next to the following columns to mark them for removal
 - i. Crawled
 - ii. Inserted
 - iii. IsAdult
 - iv. PostSize
 - c. Click **Apply Settings** to remove the marked columns
13. Click **Save** to save the workbook.
14. Click **Exit**, and then click **Run** to run the workbook.

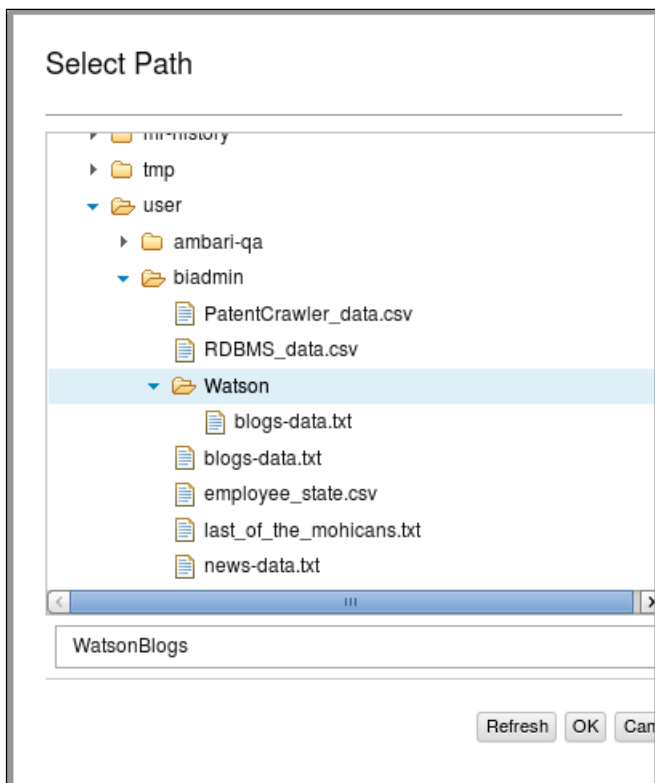
This material is meant for IBM Academic Initiative use only. NOT FOR RESALE

Task 4. Exporting the BigSheets blog data workbook to a TSV file.

You can export your BigSheets workbook to a file. Then, use that file to analyze the data in Big SQL. This capability allows you to not only use BigSheets data with Big SQL, but also to any number of systems. You can export the data into a number of different formats.

When run has reached 100%, proceed with the following steps.

1. In the menu bar of the **BlogsDataRevised** workbook, click **Export data**.
2. In the drop-down window, select **TSV** in the **Format Type** field.
3. In the **Export to** radio buttons, select **File** as the export target.
4. Click **Browse** to select a destination directory in the DFS.
5. Select your path as **/user/biadmin/Watson**.
6. Type **WatsonBlogs** as the name of the file.



7. Click **OK**.
8. Ensure that the **Include Headers** check box is not checked and then click **OK**.
9. Click **OK** to close the message dialog.

This material is meant for IBM Academic Initiative use only. NOT FOR RESALE

10. Make a note of the column names and the type of data from the BigSheets workbook that you want to define in Big SQL. You exported these columns from BigSheets:

Country - contains a two-letter country identifier.

FeedInfo - contains information from web feeds, with varying lengths.

Language - contains the string that identifies the language of the feed.

Published - contains a date and time stamp.

SubjectHtml - contains a subject that is of varying length.

Tags - contains a string of varying length that provides categories.

Type - contains the source of the web feed, whether a news blog or a public feed.

URL - contains the web address of the feed, with varying length.

Task 5. Creating a Big SQL script that creates Big SQL tables from the exported TSV file.

In this section, you create an SQL script to create Big SQL queries based on the BigSheets blogs data workbook.

1. In the Linux command line, create a SQL script named **NewsBlogs.sql**, type or paste the following code:

```
cat > /home/biadmin/labfiles/bigsheets/NewsBlogs.sql
```

```
CREATE SCHEMA IF NOT EXISTS BigSheetsAnalysis;
USE BigSheetsAnalysis;
```

```
CREATE HADOOP TABLE BigSheetsAnalysis.sheetsOut
(country VARCHAR(2), FeedInfo VARCHAR(300),
 language VARCHAR(25), published VARCHAR(25),
 subject VARCHAR(300), tags VARCHAR(100),
 type VARCHAR(20), url VARCHAR(100))
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
```

```
LOAD HADOOP USING FILE URL
'/user/biadmin/Watson/WatsonBlogs.tsv'
with SOURCE PROPERTIES ('field.delimiter'='\t')
INTO TABLE BigSheetsAnalysis.sheetsOut OVERWRITE;
```

```
SELECT * FROM BigSheetsAnalysis.sheetsOut;
```

3. Click **CTRL-D** to save and exit out of the file.

This material is meant for IBM Academic Initiative use only. NOT FOR RESALE

4. View the contents of the file to ensure you created it properly by typing the following:

```
cat /home/biadmin/labfiles/bigsheets/NewsBlogs.sql
```
5. Go to the BigInsights - Home page by using the browser bookmark or type in this URL:
<https://ibmclass.localdomain:8443/gateway/default/BigInsightsWeb/index.html>
6. Click the **Big SQL** link to open up the Big SQL UI.
 Note: If Big SQL is unavailable, you need to run the dsmKnoxSetup script as indicated in Task 2 of this lab exercise.
7. Click the **SQL Editor** link from the left side.
8. Click the **Open** link.
9. With Local selected, click the **Browse** button and navigate to **/home/biadmin/labfiles/bigsheets/NewsBlogs.sql**.

Open SQL Script

☒ Local:

☐ Server: ▼

Statement terminator: *

10. Click **OK** to open the script.
 The script you created earlier should now be inside the editor.
11. Click the **Options** link (far right of the window).
 This will expand the pane with more options.

12. Under **Database connection**, select **Name**, and then select **BIGSQL** from the dropdown.

Options

Database connection: Name ▼ BIGSQL ▼ ↻ User:bigsql

* Statement terminator: ;

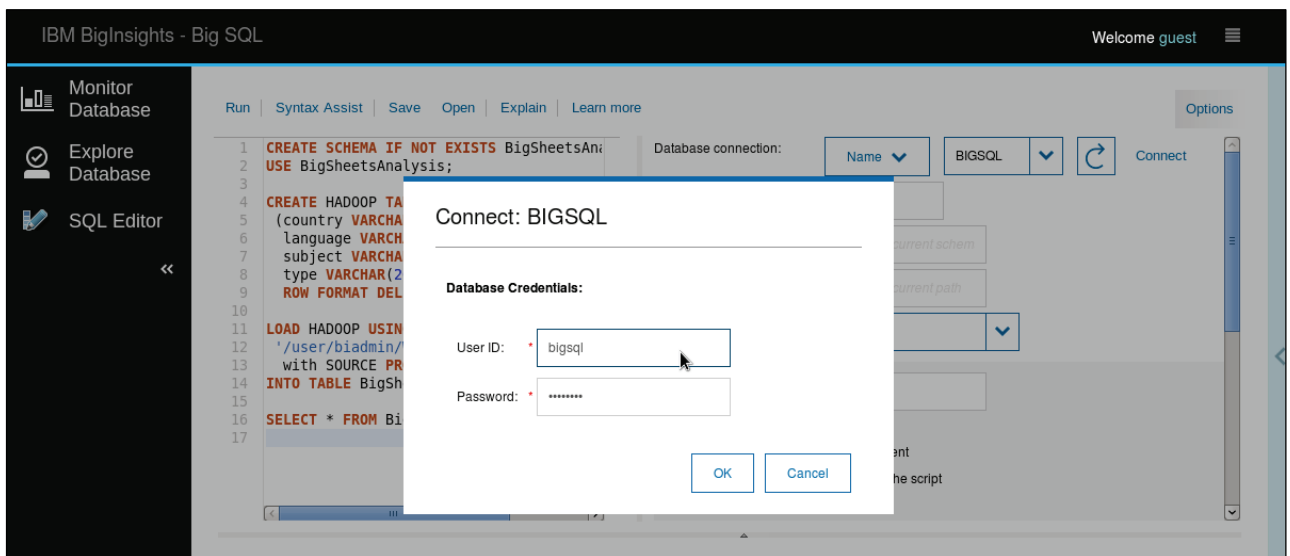
Default schema: Enter the current schema

Current path: Enter the current path

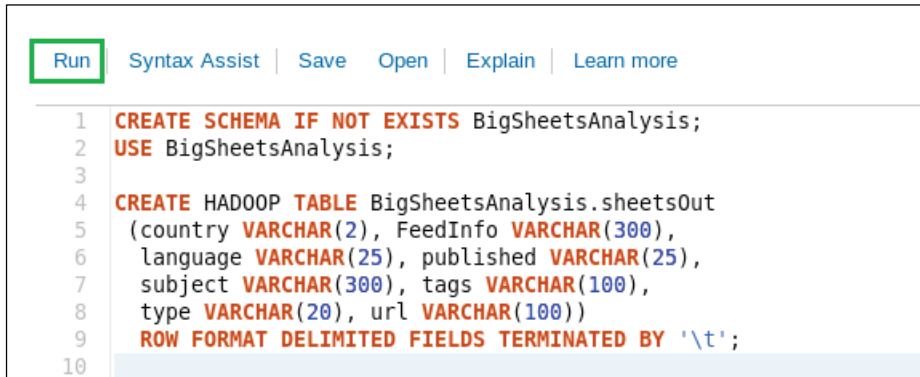
Run method: JDBC ▼

13. Click **Connect**.

14. Provide the login credentials **bigsql/ibm2blue**, and then click **OK**.



15. Click the **Run** link to run the query.

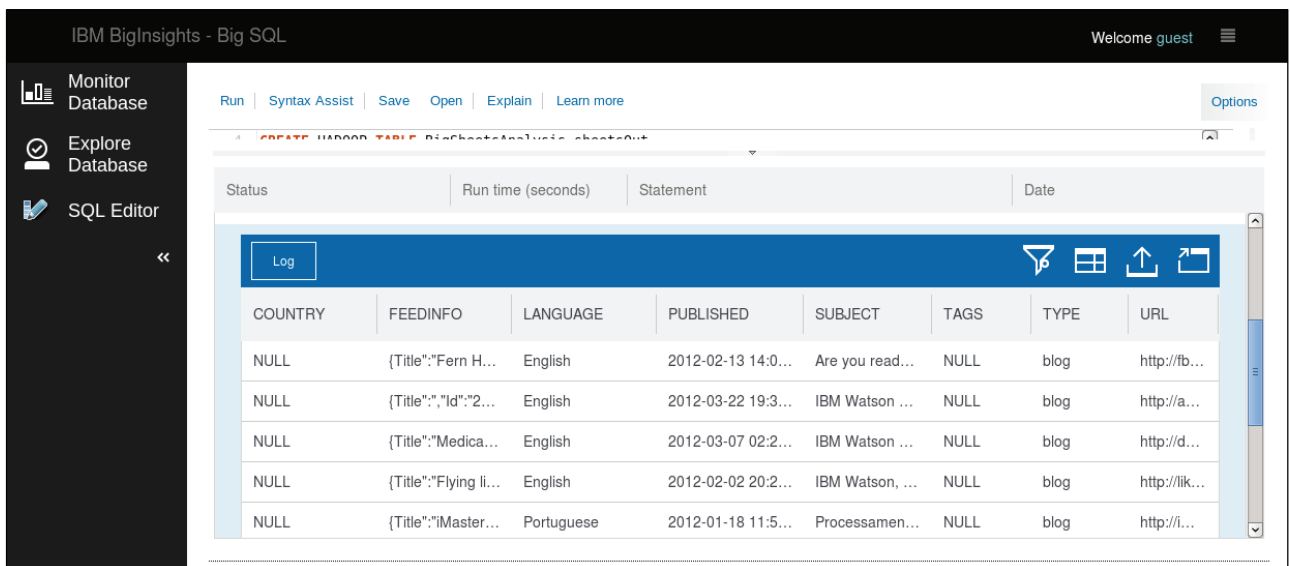


```

1 CREATE SCHEMA IF NOT EXISTS BigSheetsAnalysis;
2 USE BigSheetsAnalysis;
3
4 CREATE HADOOP TABLE BigSheetsAnalysis.sheetsOut
5 (country VARCHAR(2), FeedInfo VARCHAR(300),
6  language VARCHAR(25), published VARCHAR(25),
7  subject VARCHAR(300), tags VARCHAR(100),
8  type VARCHAR(20), url VARCHAR(100))
9 ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t';
10

```

The results are shown below:



COUNTRY	FEEDINFO	LANGUAGE	PUBLISHED	SUBJECT	TAGS	TYPE	URL
NULL	{Title:"Fern H...	English	2012-02-13 14:0...	Are you read...	NULL	blog	http://fb...
NULL	{Title":",Id":2...	English	2012-03-22 19:3...	IBM Watson ...	NULL	blog	http://a...
NULL	{Title": "Medica...	English	2012-03-07 02:2...	IBM Watson ...	NULL	blog	http://d...
NULL	{Title": "Flying li...	English	2012-02-02 20:2...	IBM Watson, ...	NULL	blog	http://lik...
NULL	{Title": "iMaster...	Portuguese	2012-01-18 11:5...	Processamen...	NULL	blog	http://i...

Now that the results of the workbook are imported into Big SQL, you can perform analytical queries. That is beyond the scope of this lab. Refer to the Big SQL course for more information.

Task 6. Creating a Big SQL table using built-in integration.

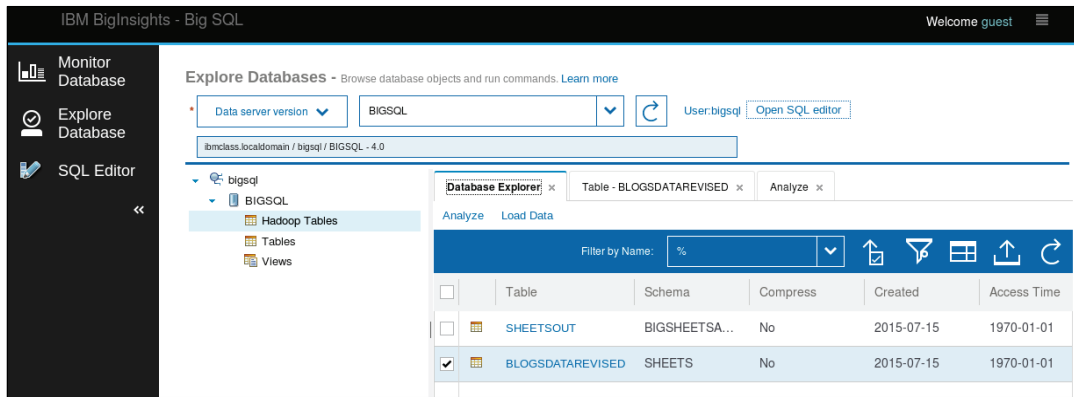
You saw in the previous section how to export BigSheets workbook as a TSV file. In fact, you can export out in a number of different file formats to be used with any number of database systems, such as Big SQL. However, there is an added bonus if you choose to use Big SQL. I'm sure you have heard of the "easy" button, and that's exactly what you have here.

1. In the **BlogsDataRevised** workbook, click **Create Table** and keep the default schema and table name (*sheets.BlogsDataRevised*).
2. Click **Confirm**.

Notice that the button changed its label to Delete Table. This means that you can only have one Big SQL table for a workbook.

This material is meant for IBM Academic Initiative use only. NOT FOR RESALE

3. Go to the **Big SQL** page and click the **Explore Database** link.
4. Connect to the **BIGSQL** database using your login credentials.
5. Navigate through the Hadoop Tables view to find your newly created table (from the click of a button).



Again, now you can work with this table as if it were any database table. You can run queries against it to find out more insight from the data.

Task 7. Troubleshooting (Optional).

Big SQL has some limitations running on a single node cluster (actually, anything less than a 3 node cluster) and for good reasons too. You do not ever want to have a single node cluster for your production environment.

In the training world, simpler is better. Your lab environment should have been configured with this fix to allow to create Big SQL tables, but if it wasn't, you can run this yourself:

1. First, open up a new terminal.
2. Switch to the **bigsql** user.
3. Run this command to connect to the bigsql database:

```
db2 connect to bigsql
Database Connection Information
Database server          = DB2/LINUX8664 10.6.3
SQL authorization ID     = BIGSQL
Local database alias     = BIGSQL
```

4. Run this command for the fix:
`db2set DB2_DYNAMIC_PMAP=INCLUDE_HEAD_NODE`
5. Restart the **Big SQL** service from **Ambari**.

This will allow you to create Big SQL tables (if you were not able to before).

Purpose:
You used Big SQL tables with BigSheets.

This material is meant for IBM Academic Initiative use only. NOT FOR RESALE