

## Lab 05: Big SQL

*Hands-On Lab*



Table of Contents

1 Introduction ..... 3

2 About this Lab ..... 3

3 Objective ..... 3

4 Environment Setup Requirements..... 3

5 Preparing for the Lab ..... 3

6 BigInsights Installation..... 4

7 Launching the BigInsights Web Console..... 4

8 Working with Big SQL..... 5

    8.1 Starting/Stopping the Big SQL server ..... 5

    8.2 Configuring the Big SQL server ..... 8

    8.3 Using the jsqsh command line client..... 8

    8.4 Admin Commands ..... 12

9 DDL, DML, Queries..... 14

    9.1 Simple query and local-access hint ..... 15

    9.2 CTAS (Create table as select), join..... 18

10 Summary..... 20

## 1 Introduction

Today many organizations need to process huge amounts of data to make better business decisions. Such processing requires scalable distributed systems. Lately Apache Hadoop and its map-reduce framework have become very popular for its robust, scalable distributed processing. While Hadoop is very good at munching big data, it is complex and time-consuming to develop map-reduce applications. Scripting language such as Pig try to solve this problem, however it requires mastering these new languages. Big SQL alleviates both of these problems by allowing users to write their queries in well-understood SQL language, and under the hood take advantage Hadoop's scalable distributed processing. While there are other SQL processors for Hadoop, Big SQL is much superior in terms of functionality and performance.

Big SQL provides SQL developers with an easy alternative for querying data managed by Hadoop. It enables data administrators to create new tables for data stored in Hive, HBase, or their BigInsights distributed file system. In addition, a LOAD command enables administrators to populate Big SQL tables with data from various sources. And Big SQL's JDBC and ODBC drivers enable many existing tools to use Big SQL to query this distributed data.

## 2 About this Lab

The purpose of this lab is to get you familiar with Big SQL server and client. You will learn how to configure and manage Big SQL jobs and run queries using the Big SQL client

## 3 Objective

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

1. Start, stop, restart, get status of Big SQL server
2. Configure Big SQL server
3. Perform basic administration of Big SQL server
4. Connect to and disconnect from Big SQL server using a client
5. Execute some DDL, DML, Queries using Big SQL client on Big SQL server

## 4 Environment Setup Requirements

To complete this lab you will need the following:

1. IBM InfoSphere BigInsights Bootcamp VMware® image
2. VMware Workstation 6.x or later

## 5 Preparing for the Lab

1. Start the VMware image by clicking the  button in VMware Workstation if it is not already on.
2. Log in to the VMware virtual machine using the following information:
  - User: **biadmin**
  - Password: **password**
3. Right-click and select "Open in Terminal", to launch the terminal window.

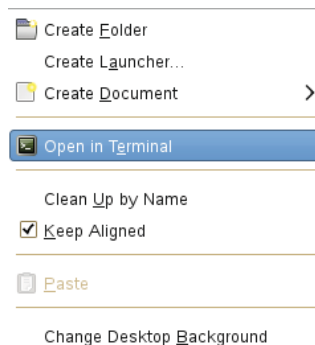


Figure 1 - Open a new terminal window

4. Start the BigInsights services by executing the following command:

```
start-all.sh
```

5. Verify all services have started successfully by checking the result of the above start command. The Failed Components list should be empty:

```
[INFO] Progress - 100%  
[INFO] DeployManager - Start; SUCCEEDED components: [zookeeper, hadoop, derby, hive, hbase, bigsql, oozie, orchestrator, console, httpfs, monitoring]; Consumes : 194038ms
```

## 6 BigInsights Installation

IBM InfoSphere BigInsights has been pre-installed on the virtual image that you have been provided. The installation has been deployed in a pseudo-distributed mode in a single node configuration.

The installation directory has been kept as `/opt/ibm/biginsights` and is easily identifiable via the `$BIGINSIGHTS_HOME` environment variable.

## 7 Launching the BigInsights Web Console

1. To launch the web console entering the following URL into a web browser. For this lab, use localhost and the default port; 8080. The format for the URL is:

```
http://<host>:<port>
```

or

```
http://<host>:<port>/data/html/index.html
```

For your convenience, there is a shortcut on the desktop, which will launch the web console when double-clicked.

2. Security has been disabled. You will not be prompted to enter a user ID and password.
3. Verify that the BigInsights Web Console is active as shown below:



Figure 2 – BigInsights Web Console

## 8 Working with Big SQL

The Web console allows administrators to inspect the overall health of the system as well as perform basic functions, such as starting and stopping specific servers (or components), adding nodes to your cluster, and so on. You'll explore a subset of these capabilities here.

### 8.1 Starting/Stopping the Big SQL server

In this section you will learn how to run basic administrative tasks on the Big SQL server, such as starting, stopping, restarting, and getting the status of the Big SQL server. Only biadmin can perform these tasks. These tasks can be performed through both the terminal and the web console.

From the terminal:

1. Change to the **\$BIGSQL\_HOME** (which by default is set to `/opt/ibm/biginsights/bigsql/bin`).

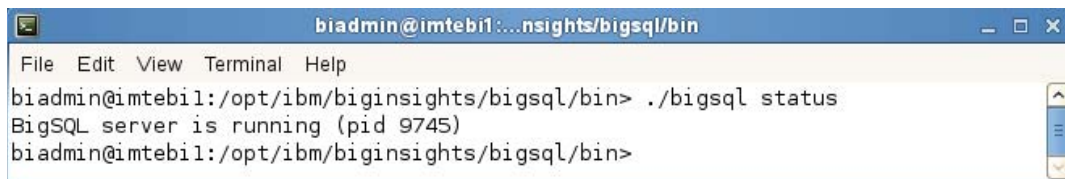
```
cd $BIGSQL_HOME/bin
```

or

```
cd /opt/ibm/biginsights/bigsql/bin
```

2. To get the status of the Big SQL server (or to see if it is running), execute:

```
./bigsql status
```



```
biadmin@imtebil:...nsights/bigsql/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin> ./bigsql status
BigSQL server is running (pid 9745)
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin>
```

Figure 3 – Get the status of Big SQL server

3. To stop the Big SQL server, execute:

```
./bigsql stop
```

Or in certain cases;

```
./bigsql forcetop
```



```
biadmin@imtebil:...nsights/bigsql/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin> ./bigsql stop
BigSQL pid 9745 stopped.
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin>
```

Figure 4 – Stop Big SQL server

4. To start the Big SQL server execute:

```
./bigsql start
```

Optionally, after a forced stop, you can also restart the Big SQL server

```
./bigsql restart
```

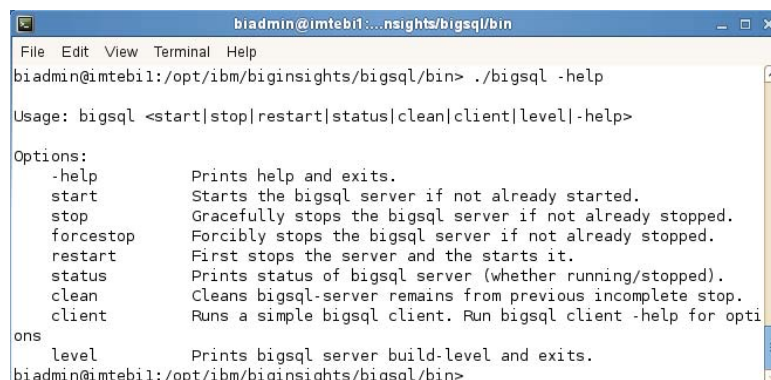


```
biadmin@imtebil:...nsights/bigsql/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin> ./bigsql start
BigSQL running, pid 13250.
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin>
```

Figure 5 – Start Big SQL server

5. For more options, execute:

```
./bigsql -help
```



```
biadmin@imtebil:...nsights/bigsql/bin
File Edit View Terminal Help
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin> ./bigsql -help

Usage: bigsql <start|stop|restart|status|clean|client|level| -help>

Options:
  -help      Prints help and exits.
  start      Starts the bigsql server if not already started.
  stop       Gracefully stops the bigsql server if not already stopped.
  forcetop   Forcibly stops the bigsql server if not already stopped.
  restart    First stops the server and the starts it.
  status     Prints status of bigsql server (whether running/stopped).
  clean      Cleans bigsql-server remains from previous incomplete stop.
  client     Runs a simple bigsql client. Run bigsql client -help for opti
ons
  level      Prints bigsql server build-level and exits.
biadmin@imtebil:/opt/ibm/biginsights/bigsql/bin>
```

Figure 6 – View the Big SQL help options

6. Navigate to the Web Console, click on “view, start, or stop a service” in tasks panel

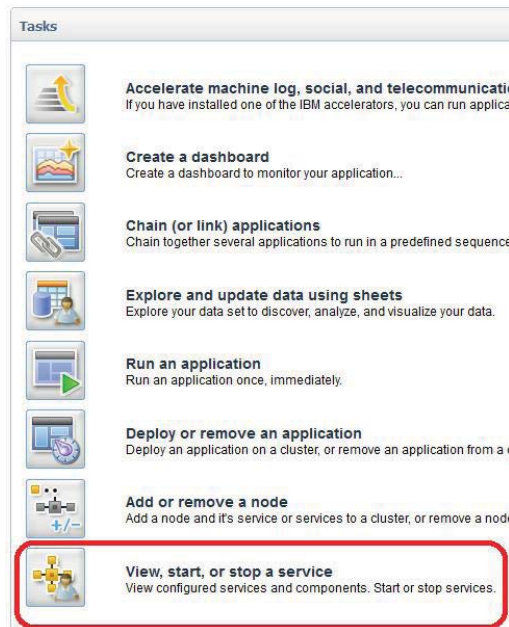


Figure 7 – View the Big SQL help options

7. The Big SQL server service will be listed on the on left-side window pane. Select it to show start/stop buttons, or to check the status of the server.
8. In the pane at right (which displays the Big SQL status), click the red Stop button to stop the service.

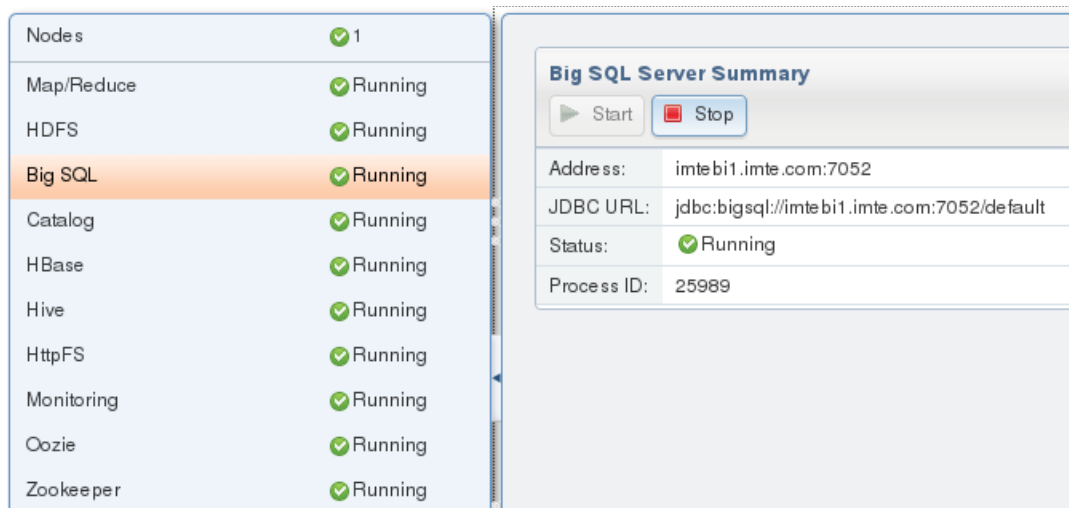


Figure 8 – Web Console showing Big SQL status running and stop button

9. When prompted to confirm that you want to stop the Big SQL service, click OK and wait for the operation to complete. The right pane should appear similar to the following image. If nothing seems to happen after a few minutes, refresh your browser.



Figure 9 – Web Console showing Big SQL service stopped

- Restart the Big SQL service by clicking the green arrow just beneath the Big SQL Server Summary heading. When the operation completes, the Web console will indicate that Big SQL is running again, likely under a process ID that differs from the earlier Big SQL process ID shown at the beginning of this lab module.

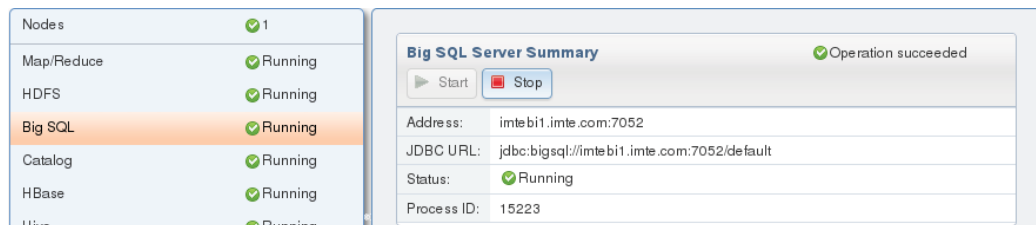


Figure 10 – Web Console showing the status after it is started back again

## 8.2 Configuring the Big SQL server

In this section, we will be configuring the Big SQL server with the appropriate variables and configuration settings. The following steps outline the bootstrap configuration via environment variables. By default, the Big SQL server uses 1 GB of memory initially, and up to one third of the memory of the machine. We are going to set the variables to lower values.

- In order to set initial and/or max memory available to Big SQL JVM, we must set these two environmental variables from the terminal:

```
export BIGSQL_CONF_INSTANCE_INITIAL_MEM=1024m # default is 1GB
export BIGSQL_CONF_INSTANCE_MAX_MEM=2048m # default is 2GB
```

- Restart the Big SQL server.

```
./bigsql restart
```

## 8.3 Using the jsqsh command line client

Although the IBM Big SQL service can be used with any client tools that uses JDBC or ODBC driver, you can use the Java SQL Shell (jsqsh) command interface that is provided with BigInsights. In this section you will get some experience using the jsqsh client.

- Change to the **\$BIGSQL\_HOME**.

```
cd $BIGSQL_HOME/bin
```

or

```
cd /opt/ibm/biginsights/bigsql/bin
```

- To start the jsqsh command line client, execute:

```
./jsqsh
```

The first time you run this command it may enter a wizard.

When prompted, hit the **Enter** key and type “Q” to exit the wizard. This should only occur the first time you start up the client.

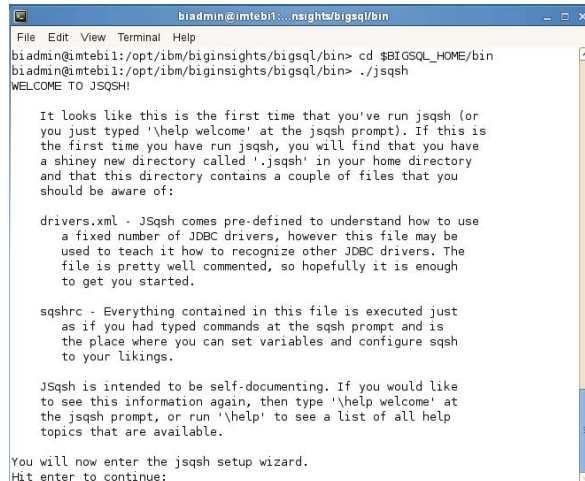


Figure 11 – jsqsh command line client wizard

3. Create a connection alias to the database-server:

**Command:** `\connect -U<user> -P<password> -S<host> -p<machine> -d<driver> a<alias_for_this_connection_for_future>`

```
\connect -Ubiadmin -Ppassword -Slocalhost -p7052 -dbigsql -auttam1
```

4. Ensure the connection alias is there:

```
\connect -l
```



Figure 12 – Display connection aliases

5. Connect using alias uttam1:

```
\connect uttam1
```

6. To show the currently connected user's user-name, execute:

```
set -v jaql.system.dataSource.userName;
```

```

biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> set -V jaql.system.dataSource.userName;
+-----+
| key | value |
+-----+
| jaql.system.dataSource.userName | biadmin |
+-----+
1 row in results(first row: 0.70s; total: 0.72s)
[localhost][biadmin] 1>

```

Figure 13 – Display connected user's user-name

7. To show the schemas, execute:

```
\show schemas
```

```

biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> \show schemas
+-----+
| TABLE_SCHEM | TABLE_CATALOG |
+-----+
| default | [NULL] |
| syscat | [NULL] |
| system | [NULL] |
+-----+
[localhost][biadmin] 1>

```

Figure 14 – Display schemas

8. To show the tables, execute:

```
\show tables
```

```

biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> \show tables
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| TABLE_CAT | TABLE_SCHE | TABLE_NAME | TABLE_TYPE | REMARKS | TYPE_CAT | TYPE_SCHEM | TYPE_NAME | SELF_REFER | REF_GENERA |
| | M | | | | | | | ENCING_COL | TION |
| | | | | | | | | _NAME | |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| [NULL] | syscat | columns | TABLE | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] |
| [NULL] | system | dual | TABLE | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] |
| [NULL] | syscat | indexcolumn | TABLE | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] |
| | | ns | | | | | | | |
| [NULL] | system | integers | TABLE | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] |
| [NULL] | syscat | schemas | TABLE | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] |
| [NULL] | syscat | tables | TABLE | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] | [NULL] |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
[localhost][biadmin] 1>

```

Figure 15 – Display tables

9. To show the columns, execute:

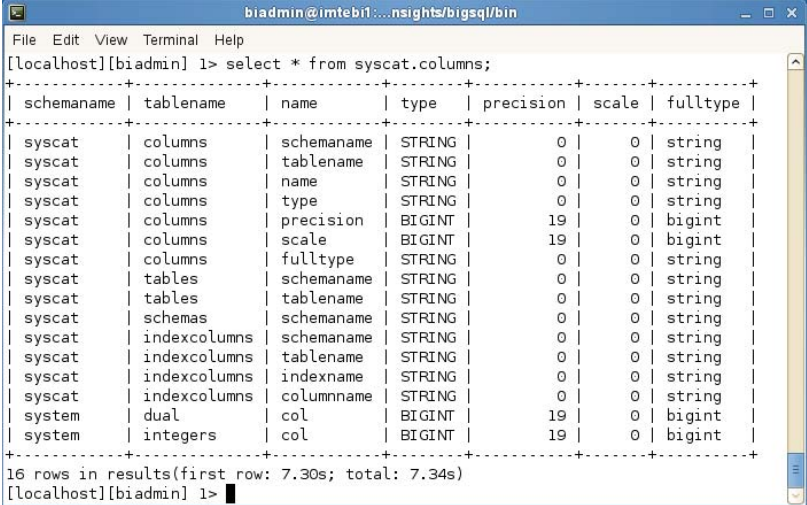
```
\show columns
```



```
\describe system.dual;
```



```
select * from syscat.columns;
```



```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> select * from syscat.columns;
```

| schemaname | tablename    | name       | type   | precision | scale | fulltype |
|------------|--------------|------------|--------|-----------|-------|----------|
| syscat     | columns      | schemaname | STRING | 0         | 0     | string   |
| syscat     | columns      | tablename  | STRING | 0         | 0     | string   |
| syscat     | columns      | name       | STRING | 0         | 0     | string   |
| syscat     | columns      | type       | STRING | 0         | 0     | string   |
| syscat     | columns      | precision  | BIGINT | 19        | 0     | bigint   |
| syscat     | columns      | scale      | BIGINT | 19        | 0     | bigint   |
| syscat     | columns      | fulltype   | STRING | 0         | 0     | string   |
| syscat     | tables       | schemaname | STRING | 0         | 0     | string   |
| syscat     | tables       | tablename  | STRING | 0         | 0     | string   |
| syscat     | schemas      | schemaname | STRING | 0         | 0     | string   |
| syscat     | indexcolumns | schemaname | STRING | 0         | 0     | string   |
| syscat     | indexcolumns | tablename  | STRING | 0         | 0     | string   |
| syscat     | indexcolumns | indexname  | STRING | 0         | 0     | string   |
| syscat     | indexcolumns | columnname | STRING | 0         | 0     | string   |
| system     | dual         | col        | BIGINT | 19        | 0     | bigint   |
| system     | integers     | col        | BIGINT | 19        | 0     | bigint   |

```
16 rows in results(first row: 7.30s; total: 7.34s)
[localhost][biadmin] 1>
```

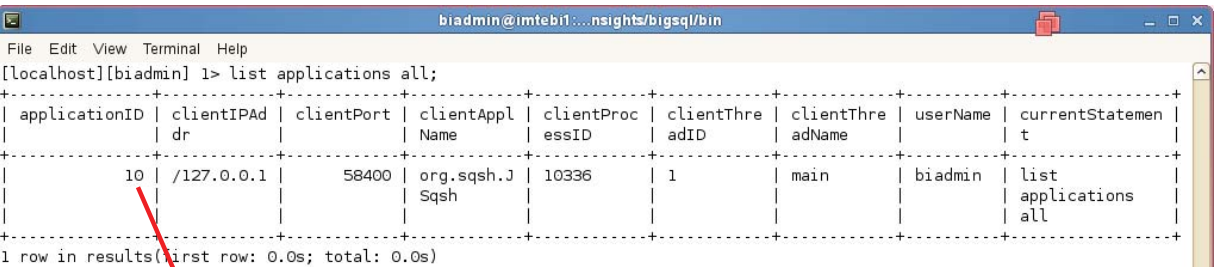
Figure 18 – Display all from syscat.columns

## 8.4 Admin Commands

In this section, it shows how to manage applications. Examples in this section use the **biadmin** user which has a **/user/biadmin** directory in its distributed file system. If you're accessing a BigInsights cluster using a different user ID, adapt the instructions in this exercise to work with your home directory in HDFS.

1. To list applications/connections info (e.g. application-id, client-ip, client-port, and so on) for all applications, execute:

```
list applications all;
```



```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> list applications all;
```

| applicationID | clientIPAddr | clientPort | clientAppName  | clientProcessID | clientThreadID | clientThreadName | userName | currentStatement      |
|---------------|--------------|------------|----------------|-----------------|----------------|------------------|----------|-----------------------|
| 10            | /127.0.0.1   | 58400      | org.sqsh.JSqsh | 10336           | 1              | main             | biadmin  | list applications all |

```
1 row in results(first row: 0.0s; total: 0.0s)
```

Figure 19 – Display list of all applications

2. To list the applications/connections info for specific applications (see application id on the previous command output), execute:

```
list applications 10;
```

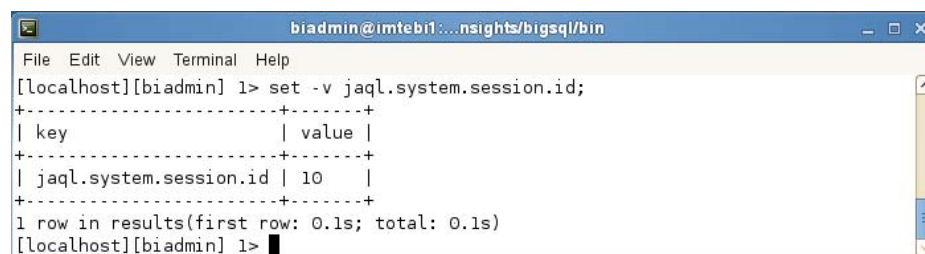


```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> list applications 10;
+-----+-----+-----+-----+-----+-----+-----+-----+
| applicationID | clientIPAddr | clientPort | clientAppName | clientProcessID | clientThreadID | clientThreadName | userName | currentStatement |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 10 | /127.0.0.1 | 58400 | org.sqsh.JSqsh | 10336 | 1 | main | biadmin | list applications 10 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in results(first row: 0.0s; total: 0.0s)
```

Figure 20 – Display info for application 10

3. To figure out the application-id of this connection, execute:

```
set -v jaql.system.session.id;
```

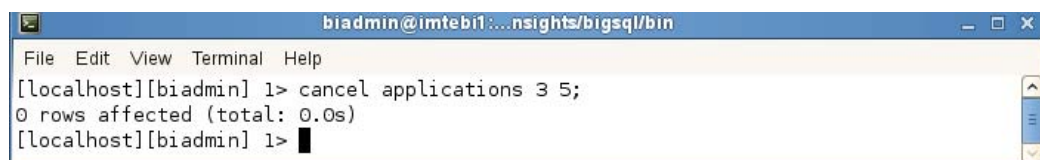


```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> set -v jaql.system.session.id;
+-----+-----+
| key | value |
+-----+-----+
| jaql.system.session.id | 10 |
+-----+-----+
1 row in results(first row: 0.1s; total: 0.1s)
[localhost][biadmin] 1>
```

Figure 21 – Display application-id connection

4. To cancel specific or all applications/connections, execute:

```
cancel applications 3 5;
```



```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> cancel applications 3 5;
0 rows affected (total: 0.0s)
[localhost][biadmin] 1>
```

Figure 22 – Command to cancel specific applications

In this case there will be 0 affected rows, because there are no applications 3 or 5.

Following should be used with caution:

```
Jsqsh> cancel applications all;
```

Note:

- admin (e.g. biadmin) can list/cancel any user's connections. Regular users can list/cancel only their connections.
- If application-list (1 3 77 92 etc, above) or "all" clause is specified then Big SQL will silently filter out what a user is not allowed to list/cancel.
- With this command, you cannot cancel the session that you are in

## 9 DDL, DML, Queries

We will create a schema named “mybigsql”.

1. Drop the existing schema, if any, with the same name:

```
DROP SCHEMA IF EXISTS mybigsql CASCADE;
```

2. Create our brand new schema

```
CREATE SCHEMA IF NOT EXISTS mybigsql;
```

3. Make sure that schema is known to Big SQL (i.e. schema exists in catalog)

```
\show schemas
```

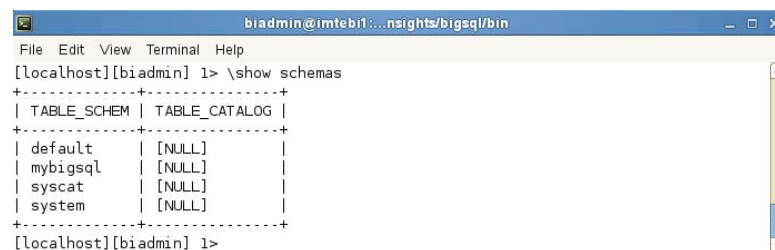


Figure 23 – Show schemas that exist in the catalog

Tell Big SQL server that all objects (table etc) we refer to in our DDL, DML, queries should use this schema.

**TIP:** Every time a new Big SQL connection is established, this should be the 1st statement. Otherwise you will end up creating objects in default schema

4. Set the schema by executing:

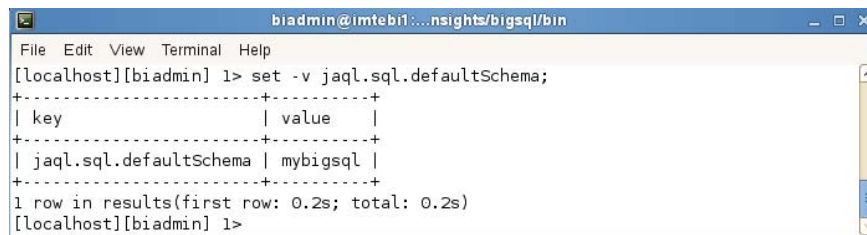
```
USE mybigsql;
```



Figure 24 – Set the schema to mybigsql

5. To check the current default schema:

```
set -v jaql.sql.defaultSchema;
```



```
biadmin@imtebi1...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> set -v jaql.sql.defaultSchema;
+-----+-----+
| key          | value    |
+-----+-----+
| jaql.sql.defaultSchema | mybigsql |
+-----+-----+
1 row in results(first row: 0.2s; total: 0.2s)
[localhost][biadmin] 1>
```

Figure 25 – Check the default schema

## 9.1 Simple query and local-access hint

In this section we will run some queries and also provide query hints to run it locally in BigSql Server to avoid MapReduce overhead for small data sets.

### 1. Set schema

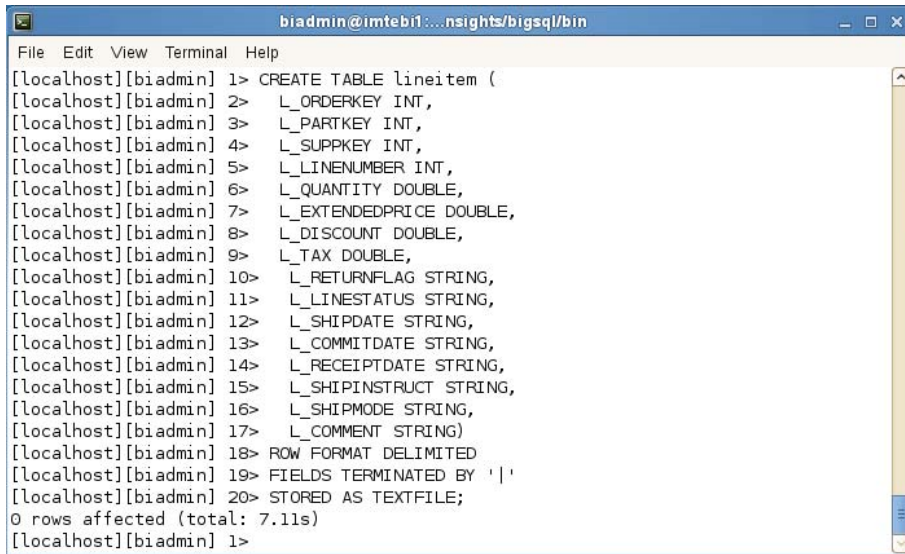
```
Use mybigsql;
```

### 2. Drop existing table

```
DROP TABLE IF EXISTS lineitem;
```

### 3. Create a table. (Note: “STORED AS TEXTFILE” clause tells that this will be a text-file in hdfs.)

```
CREATE TABLE lineitem (
  L_ORDERKEY INT,
  L_PARTKEY INT,
  L_SUPPKEY INT,
  L_LINENUMBER INT,
  L_QUANTITY DOUBLE,
  L_EXTENDEDPRICE DOUBLE,
  L_DISCOUNT DOUBLE,
  L_TAX DOUBLE,
  L_RETURNFLAG STRING,
  L_LINESTATUS STRING,
  L_SHIPDATE STRING,
  L_COMMITDATE STRING,
  L_RECEIPTDATE STRING,
  L_SHIPINSTRUCT STRING,
  L_SHIPMODE STRING,
  L_COMMENT STRING)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '|'
STORED AS TEXTFILE;
```

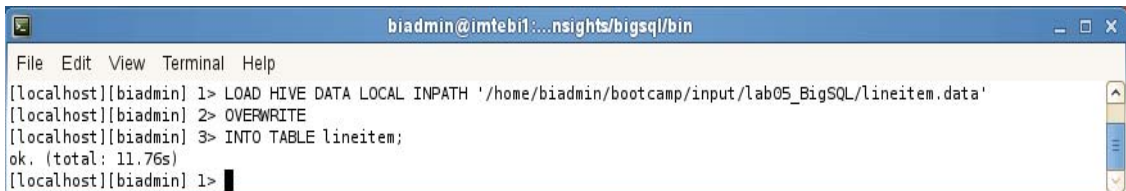


```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> CREATE TABLE lineitem (
[localhost][biadmin] 2>   L_ORDERKEY INT,
[localhost][biadmin] 3>   L_PARTKEY INT,
[localhost][biadmin] 4>   L_SUPPKEY INT,
[localhost][biadmin] 5>   L_LINENUMBER INT,
[localhost][biadmin] 6>   L_QUANTITY DOUBLE,
[localhost][biadmin] 7>   L_EXTENDEDPRICE DOUBLE,
[localhost][biadmin] 8>   L_DISCOUNT DOUBLE,
[localhost][biadmin] 9>   L_TAX DOUBLE,
[localhost][biadmin] 10>  L_RETURNFLAG STRING,
[localhost][biadmin] 11>  L_LINESTATUS STRING,
[localhost][biadmin] 12>  L_SHIPDATE STRING,
[localhost][biadmin] 13>  L_COMMITDATE STRING,
[localhost][biadmin] 14>  L_RECEIPTDATE STRING,
[localhost][biadmin] 15>  L_SHIPINSTRUCT STRING,
[localhost][biadmin] 16>  L_SHIPMODE STRING,
[localhost][biadmin] 17>  L_COMMENT STRING)
[localhost][biadmin] 18> ROW FORMAT DELIMITED
[localhost][biadmin] 19> FIELDS TERMINATED BY '|'
[localhost][biadmin] 20> STORED AS TEXTFILE;
0 rows affected (total: 7.11s)
[localhost][biadmin] 1>
```

Figure 26 – Create a table

4. Load 6,005 rows in newly created table

```
LOAD HIVE DATA LOCAL INPATH '/home/biadmin/bootcamp/input/lab05_BigSQL/lineitem.data'
OVERWRITE
INTO TABLE lineitem;
```

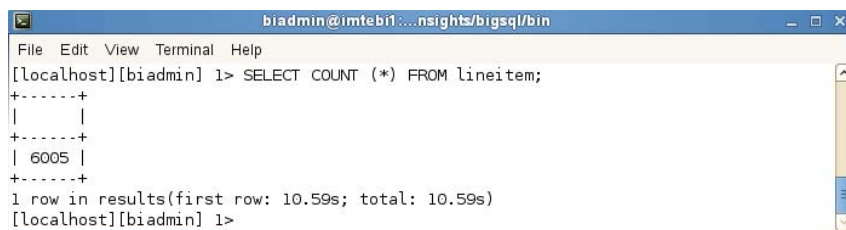


```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> LOAD HIVE DATA LOCAL INPATH '/home/biadmin/bootcamp/input/lab05_BigSQL/lineitem.data'
[localhost][biadmin] 2> OVERWRITE
[localhost][biadmin] 3> INTO TABLE lineitem;
ok. (total: 11.76s)
[localhost][biadmin] 1>
```

Figure 27 – Load data into Hive table from lineitem.data

5. Run a simple query

```
SELECT COUNT (*) FROM lineitem;
```



```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> SELECT COUNT (*) FROM lineitem;
+-----+
|       |
+-----+
| 6005  |
+-----+
1 row in results(first row: 10.59s; total: 10.59s)
[localhost][biadmin] 1>
```

Figure 28 – Run a query to count number of rows in lineitem table

6. You can force local-read mode using “accessmode” hint.

```
SELECT COUNT (*) FROM lineitem /*+ accessmode='local' */;
```

```

biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> SELECT COUNT (*) FROM lineitem /*+ accessmode='local' +*/;
+-----+
|       |
+-----+
| 6005 |
+-----+
1 row in results(first row: 1.84s; total: 1.84s)
[localhost][biadmin] 1>

```

Figure 29 – Run a query using local-read mode

7. Similarly you can force map-reduce as follows:

```
SELECT COUNT (*) FROM lineitem /*+ accessmode='MR' +*/;
```

```

biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> SELECT COUNT (*) FROM lineitem /*+ accessmode='MR' +*/;
+-----+
|       |
+-----+
| 6005 |
+-----+
1 row in results(first row: 1.25s; total: 1.25s)
[localhost][biadmin] 1>

```

Figure 30 – Run a query using Map Reduce

8. Here is how you can set local mode for all queries in the session:

```
SET FORCE LOCAL ON;
```

```
SELECT COUNT (*) FROM lineitem;
```

```
SELECT L_SHIPMODE, COUNT (*) FROM lineitem GROUP BY L_SHIPMODE;
```

```

biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> SELECT L_SHIPMODE, COUNT (*) FROM lineitem GROUP BY L_SHIPMODE;
+-----+
| l_shipmode |      |
+-----+
| SHIP      | 828 |
| RAIL      | 868 |
| TRUCK     | 903 |
| MAIL     | 824 |
| FOB      | 865 |
| REG AIR   | 879 |
| AIR      | 838 |
+-----+
7 rows in results(first row: 3.50s; total: 3.50s)
[localhost][biadmin] 1>

```

Figure 31 – Run a select query in local mode

```
SET FORCE LOCAL OFF;
```

## 9.2 CTAS (Create table as select), join

In this section we see how **Big SQL** provides the ability to create a table from a **SELECT** statement, similar to Hive and also execute a join on two tables.

1. Set schema.

```
USE mybigsql;
```

2. Drop existing table.

```
DROP TABLE IF EXISTS orders1;
```

3. Create table.

```
CREATE TABLE orders1
(
  O_ORDERKEY BIGINT,
  O_CUSTKEY INTEGER,
  O_ORDERSTATUS CHAR(1),
  O_TOTALPRICE FLOAT,
  O_ORDERDATE TIMESTAMP,
  O_ORDERPRIORITY CHAR(15),
  O_CLERK CHAR(15),
  O_SHIPPRIORITY INTEGER,
  O_COMMENT VARCHAR(79)
)
row format delimited fields terminated by '|'
stored as textfile;
```

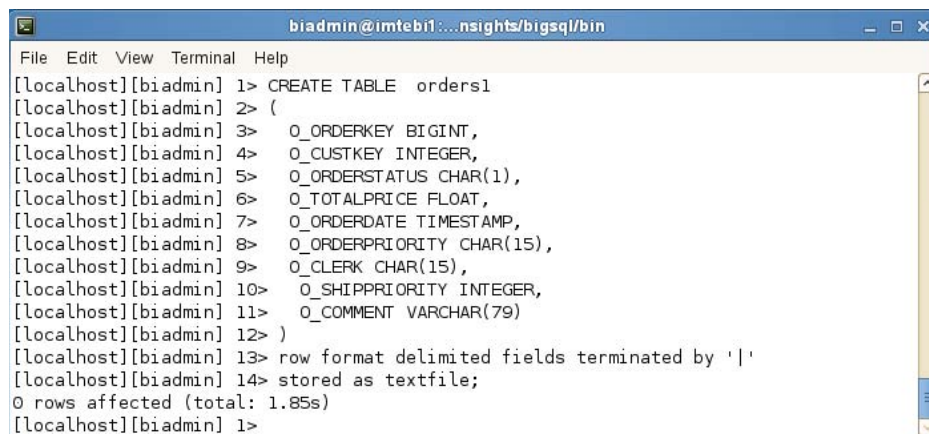


Figure 32 – Create a table orders1

4. Load 1,500 rows.

```
LOAD HIVE DATA LOCAL INPATH '/home/biadmin/bootcamp/input/lab05_BigSQL/orders.data'
OVERWRITE
INTO TABLE orders1;
```



```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> LOAD HIVE DATA LOCAL INPATH '/home/biadmin/bootcamp/input/lab05_BigSQL/orders.data'
[localhost][biadmin] 2> OVERWRITE
[localhost][biadmin] 3> INTO TABLE orders1;
ok. (total: 13.75s)
[localhost][biadmin] 1>
```

Figure 33 – Load data from orders.data file in Hive table

5. Verify data.

```
SELECT COUNT (*) FROM orders1 /*+ accessmode='local' +*/;
```

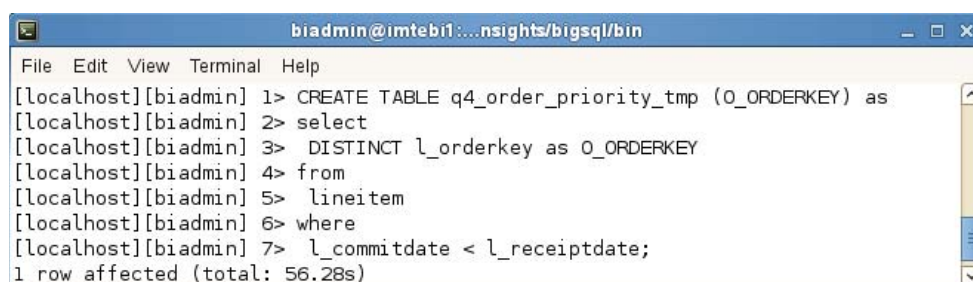


```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> SELECT COUNT (*) FROM orders1 /*+ accessmode='local' +*/;
+-----+
|       |
+-----+
| 1500 |
+-----+
1 row in results(first row: 31.10s; total: 31.10s)
```

Figure 34 – Run a select query to count rows from orders1

6. A CTAS (create table as select) query to prepare for another query: (this query might take a few minutes to run)

```
CREATE TABLE q4_order_priority_tmp (O_ORDERKEY) as
select
  DISTINCT l_orderkey as O_ORDERKEY
from
  lineitem
where
  l_commitdate < l_receiptdate;
```

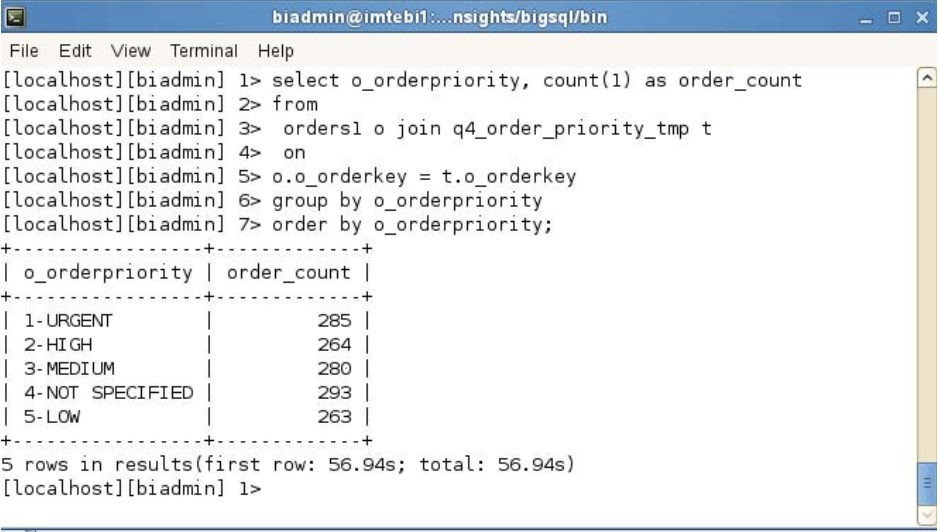


```
biadmin@imtebi1:...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> CREATE TABLE q4_order_priority_tmp (O_ORDERKEY) as
[localhost][biadmin] 2> select
[localhost][biadmin] 3>   DISTINCT l_orderkey as O_ORDERKEY
[localhost][biadmin] 4> from
[localhost][biadmin] 5>   lineitem
[localhost][biadmin] 6> where
[localhost][biadmin] 7>   l_commitdate < l_receiptdate;
1 row affected (total: 56.28s)
```

Figure 35 – Create a table q4\_order\_priority\_tmp as select

7. Order priority query:

```
select o_orderpriority, count(1) as order_count
from
  orders1 o join q4_order_priority_tmp t
on
  o.o_orderkey = t.o_orderkey
group by o_orderpriority
order by o_orderpriority;
```



```
biadmin@imtebi1...nsights/bigsql/bin
File Edit View Terminal Help
[localhost][biadmin] 1> select o_orderpriority, count(1) as order_count
[localhost][biadmin] 2> from
[localhost][biadmin] 3> orders1 o join q4_order_priority_tmp t
[localhost][biadmin] 4> on
[localhost][biadmin] 5> o.o_orderkey = t.o_orderkey
[localhost][biadmin] 6> group by o_orderpriority
[localhost][biadmin] 7> order by o_orderpriority;
+-----+-----+
| o_orderpriority | order_count |
+-----+-----+
| 1-URGENT        |          285 |
| 2-HIGH          |          264 |
| 3-MEDIUM       |          280 |
| 4-NOT SPECIFIED |          293 |
| 5-LOW          |          263 |
+-----+-----+
5 rows in results(first row: 56.94s; total: 56.94s)
[localhost][biadmin] 1>
```

Figure 36 – Order priority query

8. Finally, execute the command below to exit out of Java SQL Shell (jsqsh) command interface:

```
quit
```

## 10 Summary

You have just completed Lab 5 which focused on the basics of Big SQL. You should now be able to perform the following basic tasks on the platform:

- Start, stop, restart, get status of Big SQL server
- Configure Big SQL server
- Perform basic administration of Big SQL server
- Connect to and disconnect from Big SQL server using a client
- Execute some DDL, DML, Queries using Big SQL client on Big SQL server