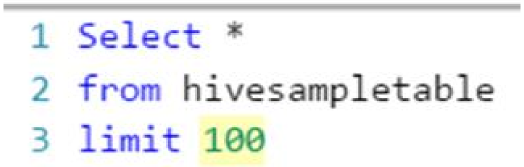
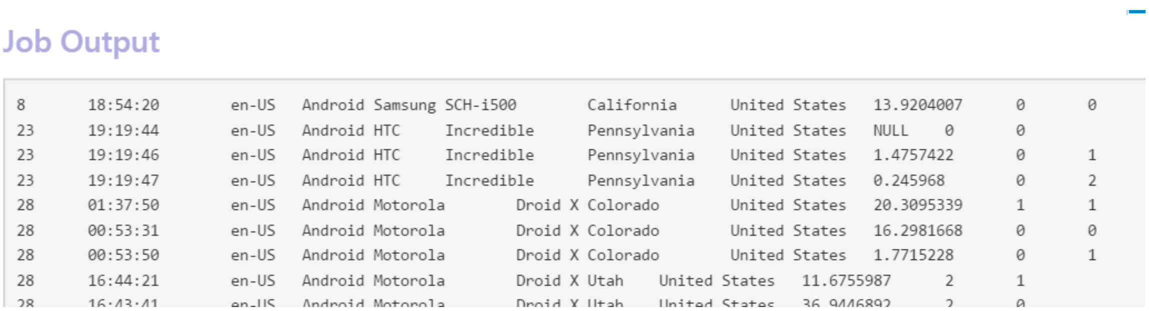
Tip For a full list of class types you can visit `http://docs.hortonworks.com/HDPDocuments/HDP1/HDP-1.2.2/ds_Hive/language_manual/datatypes.html`. HortonWorks is an entity that focuses on the development and support of Hadoop. This type of list will be important for adding more data and creating tables. HDInsight is Microsoft's implementation of HortonWorks in the cloud.

3. To get a preview of our data, use the "limit" clause at the end of the HiveQL statement. The "*" tells Hive to grab each row and "limit" returns X number of rows at random. Since limit returns a random subset of rows, this is a great keyword for partitioning or sampling large datasets (Figure 7.9).



```
Select *
from hivesampletable
limit 100
```

Figure 7.9: Limit grabs 100 random rows from the hivesampletable



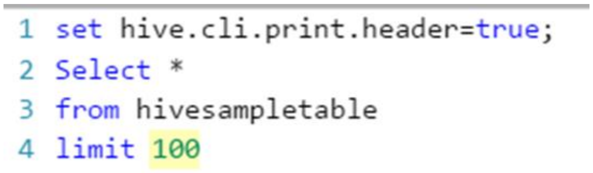
Job Output

```
8       18:54:20        en-US   Android Samsung SCH-i500        California      United States   13.9204007      0       0
23      19:19:44        en-US   Android HTC     Incredible      Pennsylvania    United States   NULL    0       0
23      19:19:46        en-US   Android HTC     Incredible      Pennsylvania    United States   1.4757422       0       1
23      19:19:47        en-US   Android HTC     Incredible      Pennsylvania    United States   0.245968        0       2
28      01:37:50        en-US   Android Motorola        Droid X Colorado        United States   20.3095339      1       1
28      00:53:31        en-US   Android Motorola        Droid X Colorado        United States   16.2981668      0       0
28      00:53:50        en-US   Android Motorola        Droid X Colorado        United States   1.7715228       0       1
28      16:44:21        en-US   Android Motorola        Droid X Utah    United States   11.6755987      2       1
28      16:43:41        en-US   Android Motorola        Droid X Utah    United States   36.9446892      2       0
```

Figure 7.10: Output preview of the 100 random rows

4. Include "**set hive.cli.print.header=true;**" to see headers. (Figure 7.11)

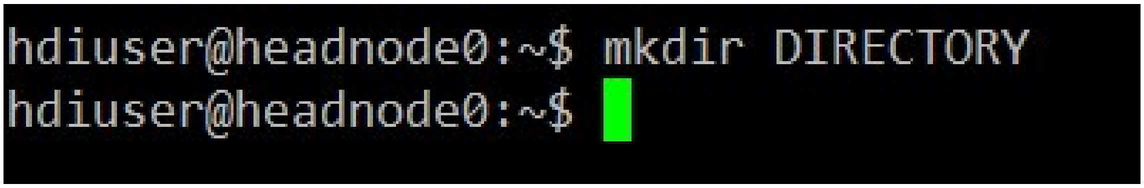
Hive does not print headers by default. Hadoop is designed for unstructured data, which does not require headers or prescribed schemas. It is also important to take note of the ";" at the end. That means it is a separate statement altogether and two query results will be returned instead of one. This is a Hive command rather than a HiveQL statement. The Hive engine will perform two statements; the first statement will query a mini SQL database stored on the Hadoop cluster that stores all hive table schemas, the second statement will query "hivesampletable" within the Hadoop cluster's HDFS. Both results will be returned and vertically joined together, also called a "union" in SQL terms.



```
set hive.cli.print.header=true;
Select *
from hivesampletable
limit 100
```

Figure 7.11: Explicitly adding headers to the query

After you type a command and press Enter, the command will execute and underneath it, you should get a message similar to the picture above. If there was an error in your command PuTTY will let you know. A successful command execution looks like the following:



```
hdiuser@headnode0:~$ mkdir DIRECTORY
hdiuser@headnode0:~$
```

Figure 8.8: Successful command execution

This command made a DIRECTORY called DIRECTORY.

8.1.5 Exercise: Constructing the Model

1. We are now ready to start constructing our model. We do this with the following command:

```
hadoop fs -mkdir /census
```

2. You can call the directory anything, but make sure that you include the slash before the directory name. Enter this command into PuTTY and press enter.
   Notice how Hadoop commands are similar to the commands to the UNIX commands above. For many cases, Hadoop commands can be written as

```
hadoop fs -UNIX_COMMAND
```

   The above line will apply the UNIX-COMMAND to HDFS. Entering Hadoop commands are very similar to entering UNIX commands.
3. Next, we get the data to build the model. We will copy the data from a blob storage account into the working directory you created in Step 5. Do this with the following command

```
hadoop fs -cp \
wasb://census@mahoutclassification.blob.core.windows.net/census.csv \
/census
```

   The wasb stands for Windows Azure Storage Blob. It is syntax that will natively reach out to another blob storage account in the cloud without resorting to IP's or long security tokens
4. Let's look at the data we just loaded. HDFS supports some standard unix/linux commands, but there are some noticeable gaps. For instance, neither less nor more are supported by HDFS. To look at an HDFS file without printing the entirety of it to the console, you must use the pipe command (|) to send the output of cat to a linux command. For example, try entering:

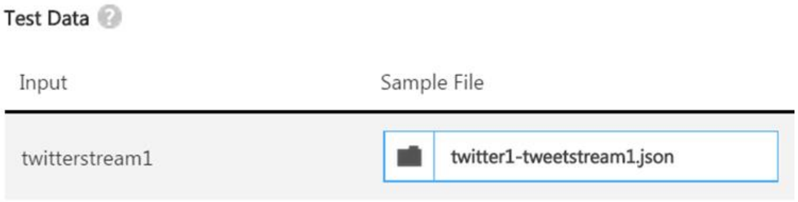
```
hadoop fs -cat /census/census.csv | less
```

   This will send the output of cat to less, allowing you to read it more easily. Notice that there are no file headers: this is important, as Mahout will not skip the file headers for you.
   In general piping multiple commands together will send the output of one command as input to the next command.

3. For this lab, we will select the following attributes and their corresponding values. “created_at”, “id” and “text”. We will also rename the columns as “tweet_date”, “tweet_id” and “tweet_body” to make it a little bit more descriptive. We are selecting from your **input alias** name. If you do not remember what that is, click on “inputs” within your stream processor, and look at the name of your input.

```
select
    created_at as tweet_date,
    id as tweet_id,
    text as tweet_body
from twitterstream1
```

4. After you have a valid query with no errors. Hit the “**test**” button.
5. It’s going to ask for your sample data from your input alias. Upload your json file that you sampled and saved earlier (Figure 10.50). Hit OK.

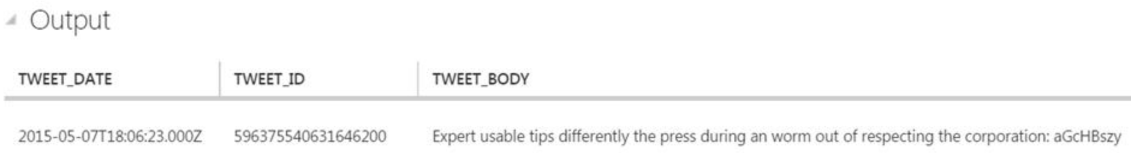


Test Data

Input	Sample File
twitterstream1	twitter1-tweetstream1.json

Figure 10.50: Upload the json file

6. You should get an output like that in Figure 10.51:



Output

TWEET_DATE	TWEET_ID	TWEET_BODY
2015-05-07T18:06:23.000Z	596375540631646200	Expert usable tips differently the press during an worm out of respecting the corporation: aGcHBszy

Figure 10.51: Output of query

7. Congratulations we have just added structure to semi-structured data. It is now in row/columnar fashion and can be exported to any database that you wish. Without the use of programmatic transformations via Java, Python, C# etc.