Hadoop: HDFS, Hive, Spark

1. Connect to the Linux shell on the **hive-server** (this is where the Hadoop client has been installed for you). On this server you will see the **/datasets** folder is mounted. Load the:
   1. customers/customers.csv,
   2. customers/surveys.csv, and
   3. tweets/tweets.psv into HDFS.

Specifically:

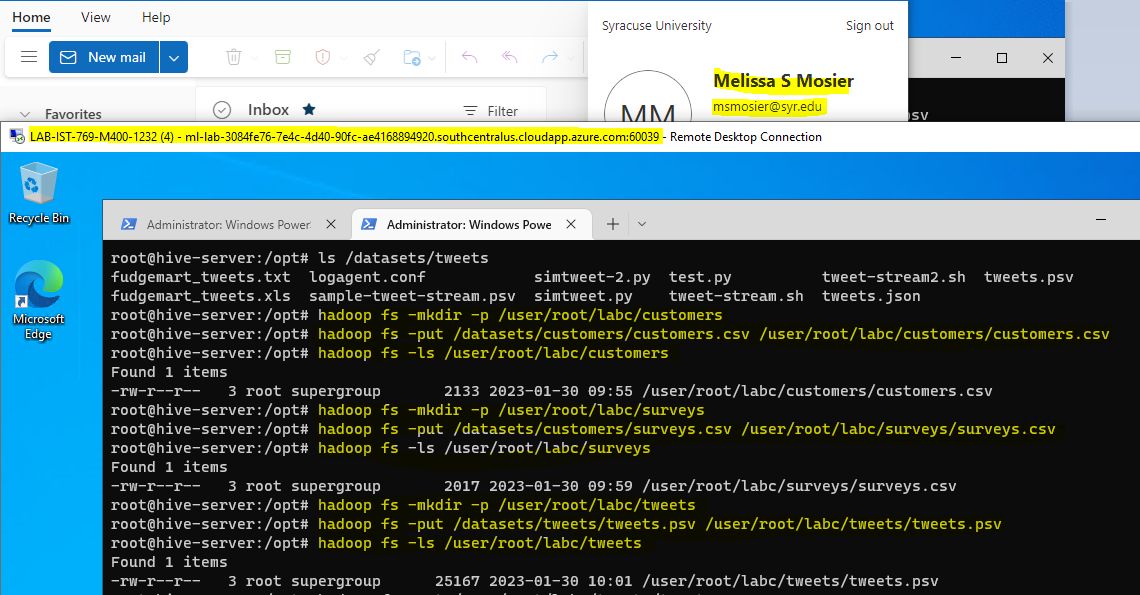
Source HDFS Location

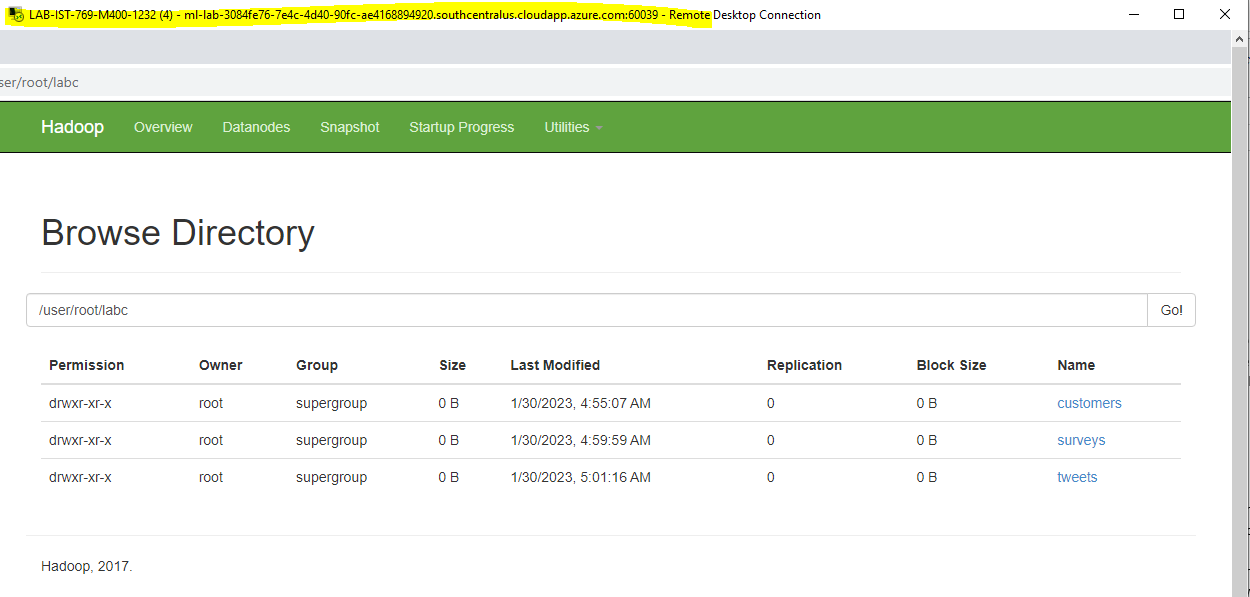
customers/customers.csv /user/root/labc/customers/customers.csv

customers/surveys.csv /user/root/labc/surveys/surveys.csv

tweets/tweets.psv /user/root/labc/tweets/tweets.psv

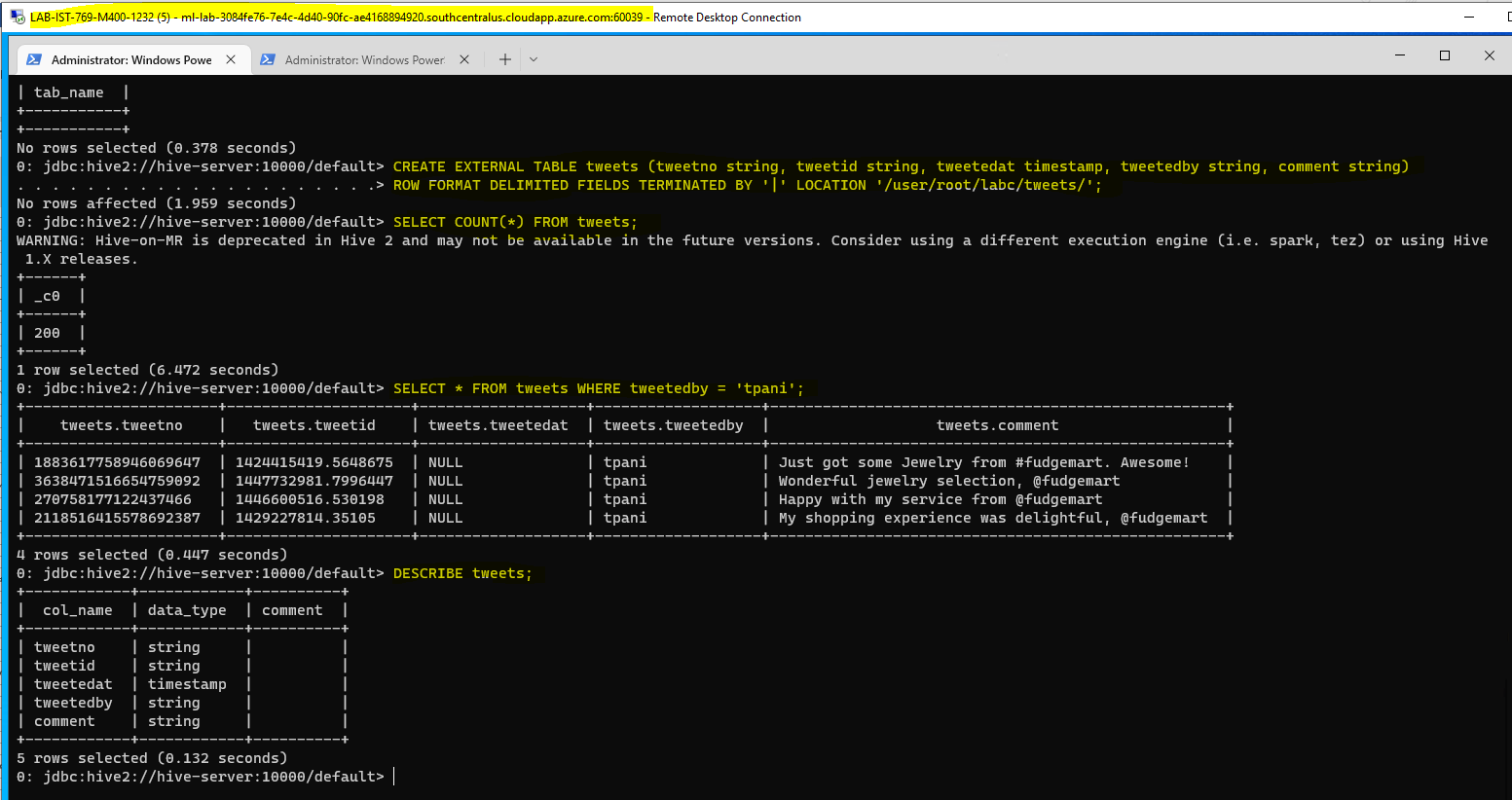
Record the Hadoop commands you entered to complete this task. Provide a screenshot of evidence these files are in HDFS. The screenshot can use the Hadoop client output or the HDFS website.



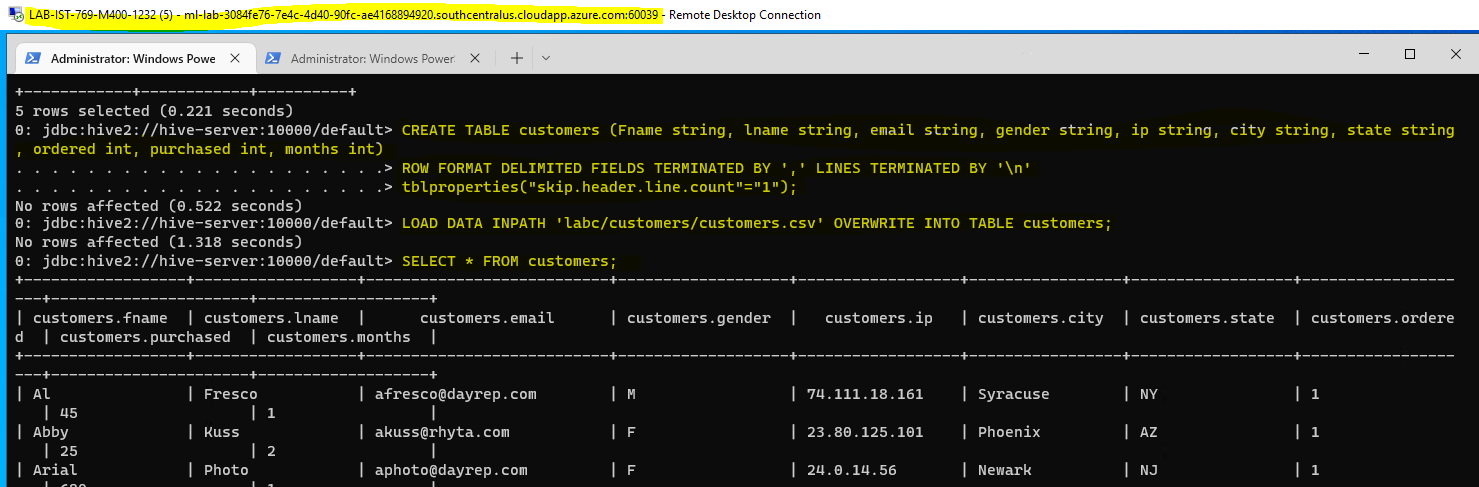


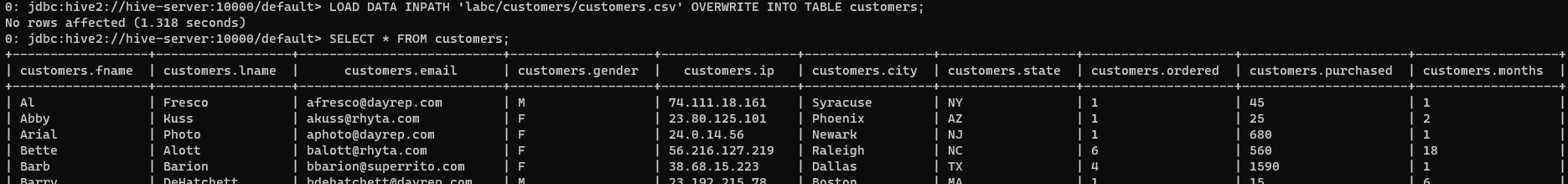
1. Create a Hive database called **labc**. In the **labc** database create an external Hive table for the **tweets**. Your external table will point to the existing location on HDFS.   
   **NOTE:** You will need to view the tweets.psv file to see the format of the file before you can create the table schema correctly.

After you create the table, write a SELECT query to display all of the tweets for a user a single user of your choice. Please include the HQL code you wrote to create and query the **tweets** table. Along with screenshots of a **describe tweets** command output along with your SELECT query output.

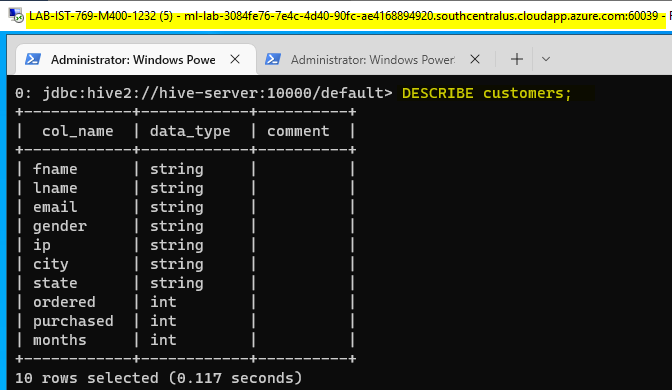


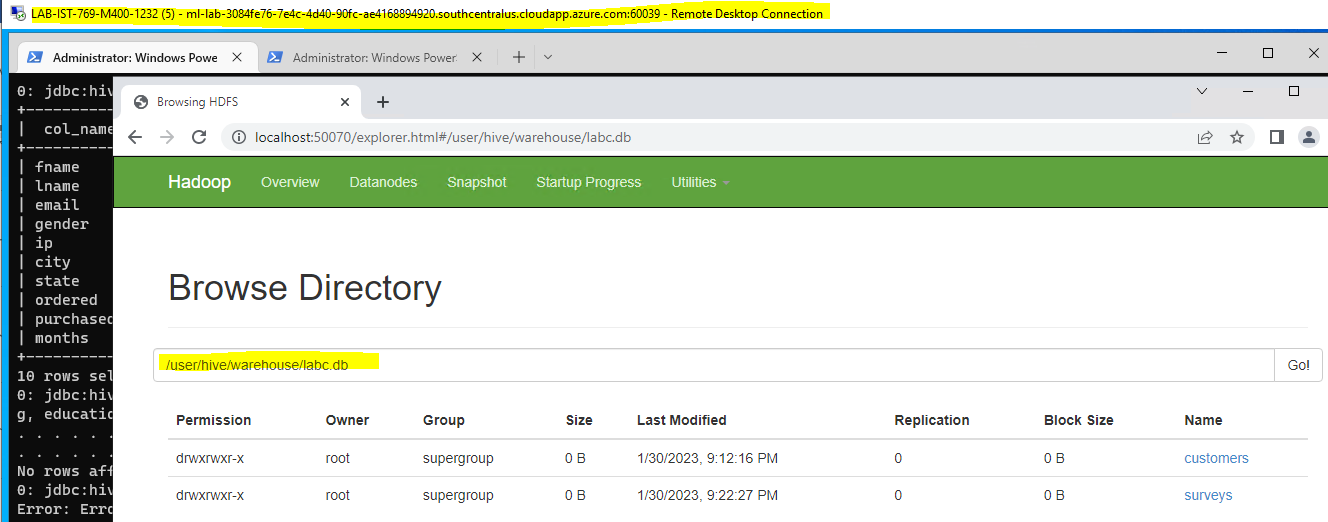
1. In the **labc** database, let’s create an internal hive table for **customers**. After you create the table, use the LOAD command to move the data from the current HDFS location into the Hive data warehouse.



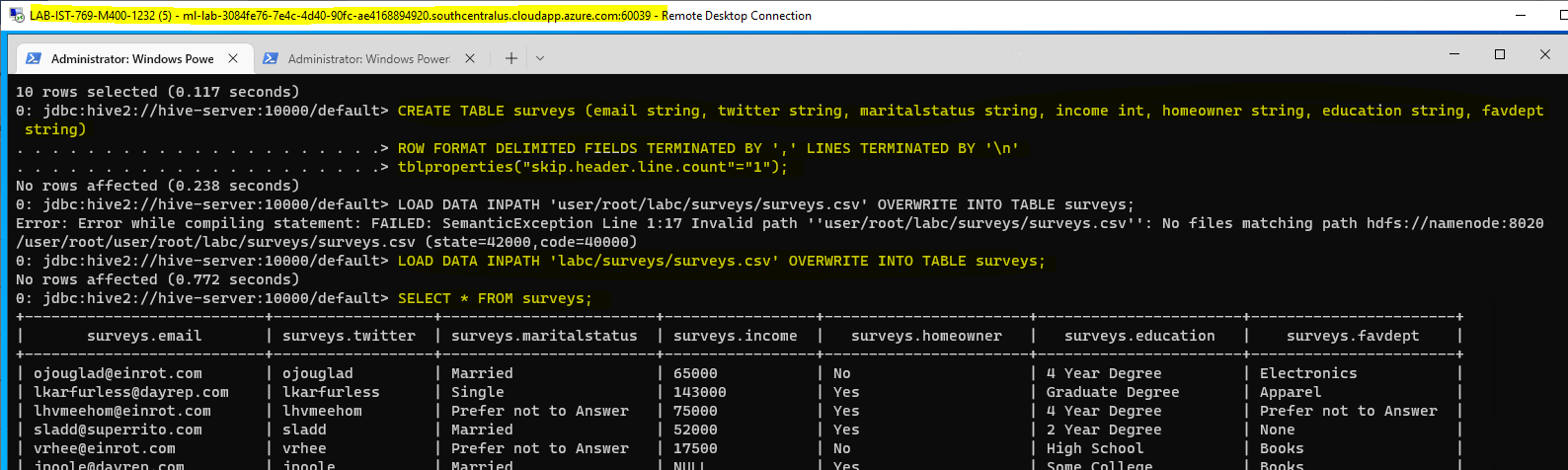


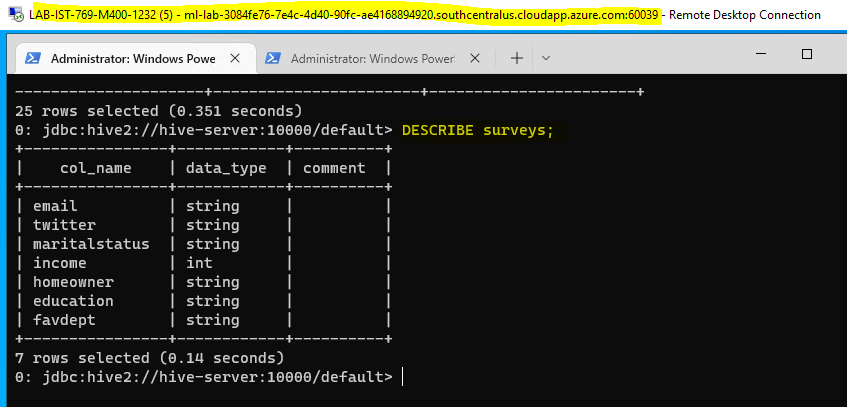
When you have created the table and imported the data, provide the HQL code you entered to complete the task and provide screenshots of the **describe customers** command, a SELECT output to show data is there, and a screenshot on web HDFS to show the data is located in **/user/hive/warehouse**.





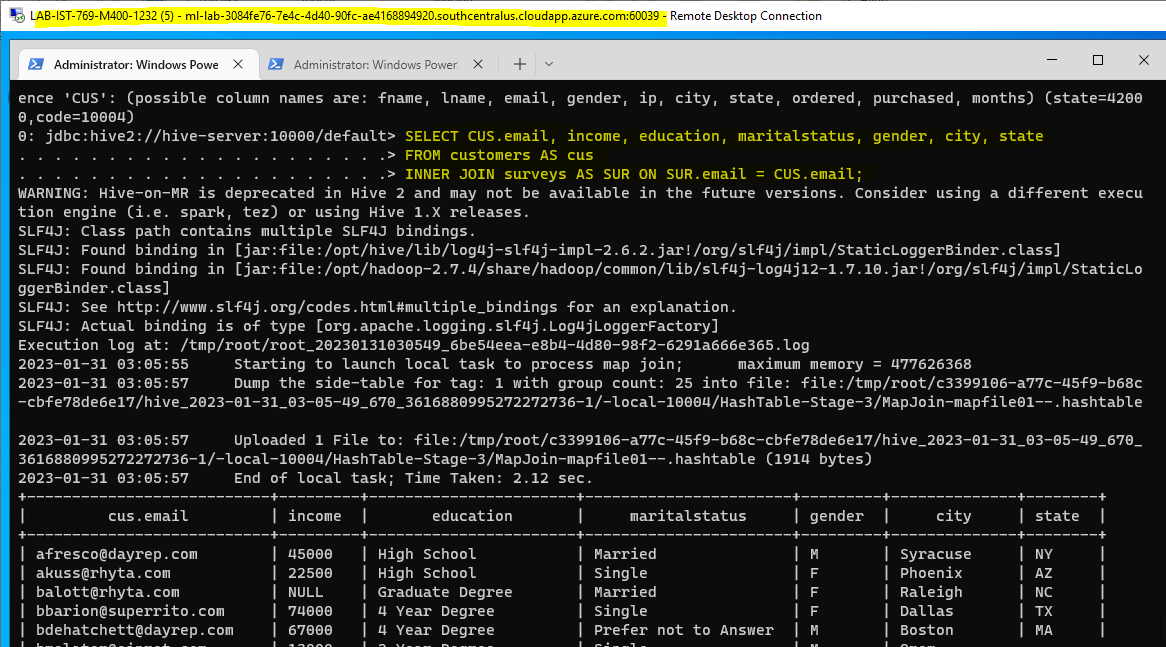
1. Like the previous step, import the surveys.csv into a Hive internal table in the **labc** database called **surveys**. When you have created the table and imported the data, provide all the commands you entered to complete the task, a screenshot of the table description, the select statement output, and web HDFS location.

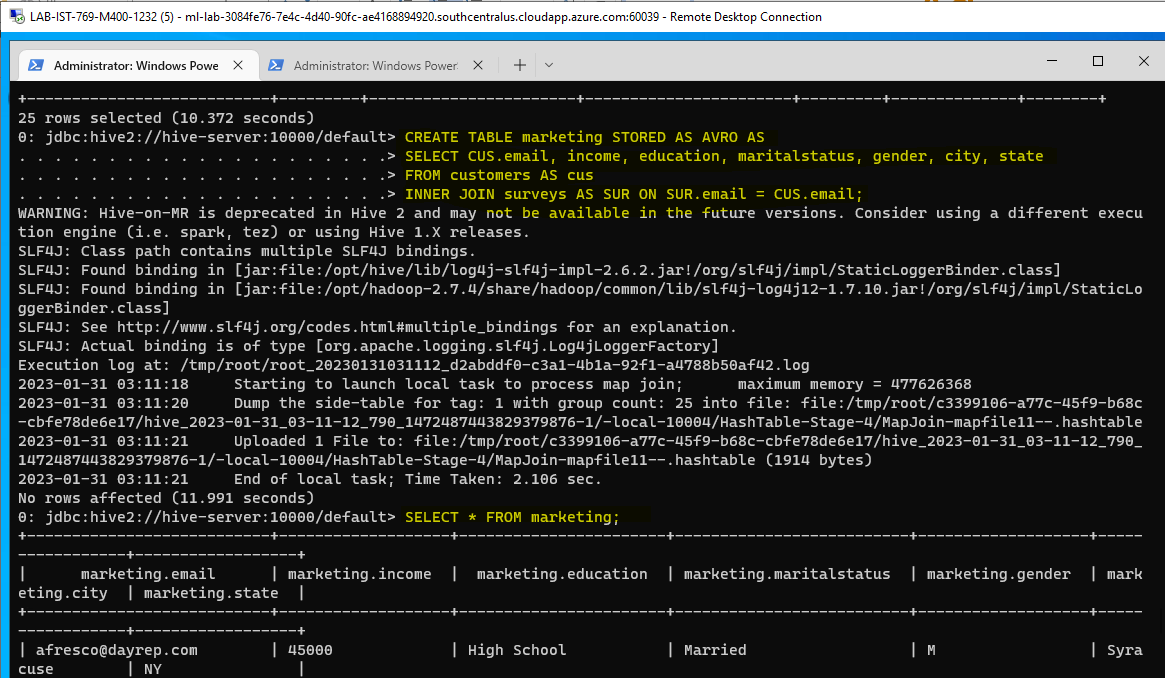




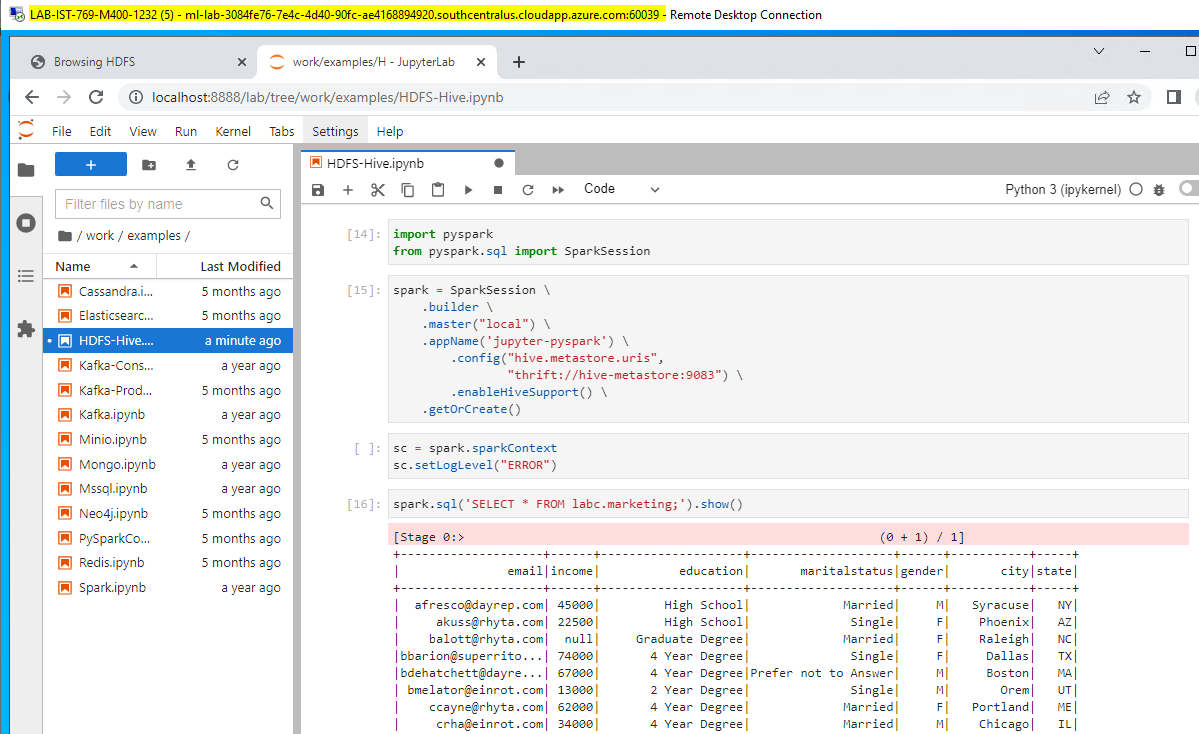
1. Open Jupyter Lab. Create a new notebook called **labc**. Copy over the code from an example to create a Spark session connected to Hive.

**Created tables using Hive in Command Line.**





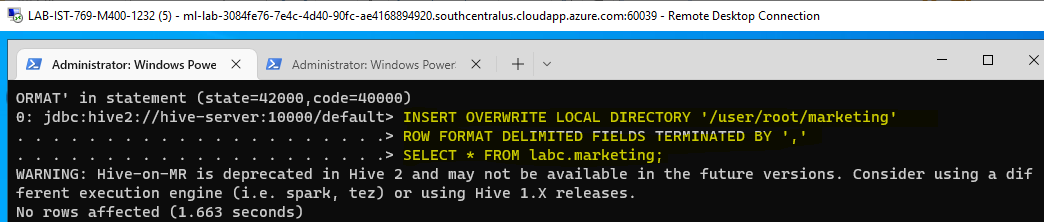
In a separate cell, write Spark SQL code to join customers to surveys on email address. Include all rows and columns and show output in the notebook. Provide a screenshot of the notebook cell with a reasonable amount of output (doesn’t need to be the entire set of rows and columns as that will be too large).



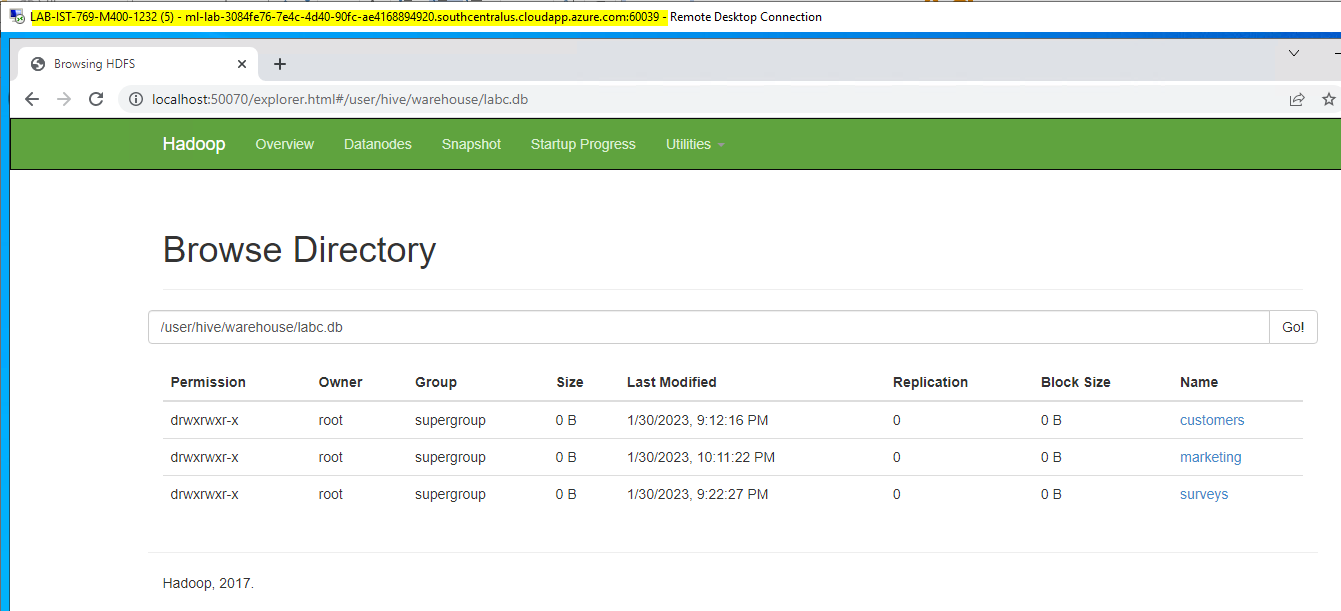
1. The marketing department would like a data set of customers/surveys for analysis. In a separate cell in the **labc** Jupyter Notebook, write a Spark SQL query to create a Hive table called **marketing** in **AVRO** file format from a SELECT query that once again joins customers and surveys on email addresses. Include the following columns in the new table: **Household Income**, **Education**, **Marital Status**, **Gender**, **City**, and **State**.  
     
   Provide a screenshot of the Jupyter cell and output that creates the new table, and another of the cell and output of executing a SELECT on the table.

**See Number 5**

1. Stupid marketing doesn’t know what they want! Now they would like the same query in the previous step, only output as a comma-delimited file instead of a Hive table. In a new Jupyter Lab cell, write Spark SQL to execute the Hive query but save the output back to HDFS in the folder **/user/root/marketing**.  
     
   Provide a screenshot of the Spark code cell and its output, as well as a screenshot of the file on web HDFS.



**labc.marketing**



**User > root > marketing**

