Homework Problem Set C Submission Form

# Overview

| Your Name | Nick Videtti |
| --- | --- |
| Your SU Email | nvidetti@syr.edu |

# Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screenshot guidelines from the first homework.

Remember as you complete the homework that it is not only about getting it right/correct. We will discuss the answers in class so it’s important to articulate anything you would like to contribute to the discussion in your answer:

* If you feel the question is vague, include any assumptions you've made.
* If you feel the answer requires interpretation or justification, provide it.
* If you do not know the answer to the question, articulate what you tried and how you are stuck.
* Highlight any doubts or questions you would like me to review.

This how you receive credit for answering questions that might not be correct. In addition, you must complete the reflection portion of the homework assignment for full credit. Since most answers will be similar this is an important part of your individual submission.

Complete Part II of this document first, then go back and complete the Reflection in Part I.

# Part I: Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment, you must be as descriptive and personal as possible with your reflection.

1. As you completed this assignment, identify what you learned.

**Walking through each of these exercises was extremely beneficial. Many of these concepts were a bit difficult to understand at first, but going through these hands-on exercises helped greatly.**

1. What barriers or challenges did you encounter while completing this assignment?

**There was quite a bit of trial and error that resulted in the exercises taking quite a long time. There was a lot of having to reference the asynchronous work and a little bit of Googling. Also, exercise 7 required a prefix to the directory that was specified, and finding what to put for the prefix took a couple hours of Googling, trial, and error.**

1. How prepared were you to complete this assignment? What can you do to be better prepared?

**I was not very prepared to complete this assignment, but I don’t think I could have been any more prepared. Going through the asynchronous work and the quiz prior to these assignments certainly help, but I think the real learning happens in these assignments.**

1. Rate your comfort level with this week’s material. Use the rubric provided.

4 ==> I understand this material and can explain it to others.  
**3 ==> I understand this material.**  
2 ==> I somewhat understand the material but sometimes need guidance from others.  
1 ==> I understand very little of this material and need extra help.

# Part II: Questions

Paste your answers to the Exercises found in the lab document. Make sure to include your NetID in any screenshots you provide. If the question asks for commands, only include those commands that are necessary to complete the tasks. Number each answer.

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.  
  
**root@hive-server:/opt# hadoop fs -mkdir /user/root**

**root@hive-server:/opt# hadoop fs -mkdir /user/root/labc**

**root@hive-server:/opt# hadoop fs -mkdir /user/root/labc/customers**

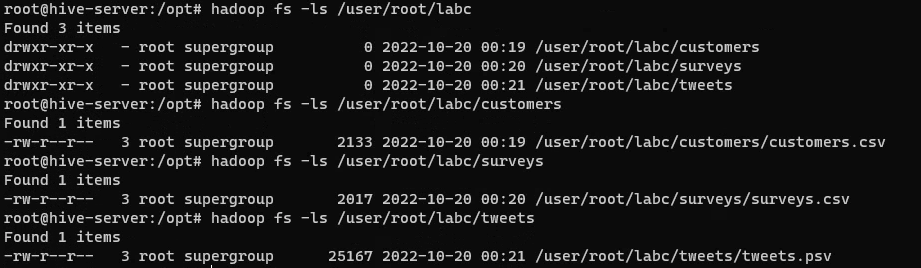
**root@hive-server:/opt# hadoop fs -mkdir /user/root/labc/surveys**

**root@hive-server:/opt# hadoop fs -mkdir /user/root/labc/tweets**

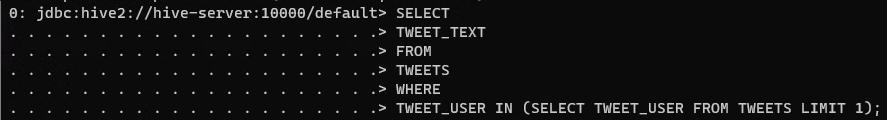
**root@hive-server:/opt# hadoop fs -put /datasets/customers/customers.csv /user/root/labc/customers/customers.csv**

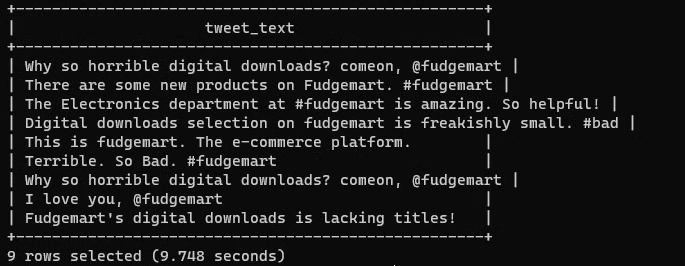
**root@hive-server:/opt# hadoop fs -put /datasets/customers/surveys.csv /user/root/labc/surveys/surveys.csv**

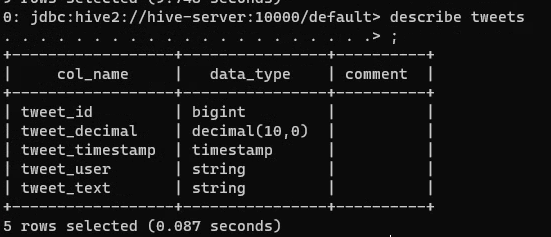
**root@hive-server:/opt# hadoop fs -put /datasets/tweets/tweets.psv /user/root/labc/tweets/tweets.psv**

****

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.

**NOTE 1:** If you screw up you will need to drop table and reload the file back into HDFS from Step 1.

**NOTE 2:** There is a header row in this file; you might need to search the Hive docs on the web for how to exclude this first row.

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**.

**0: jdbc:hive2://hive-server:10000/default> create table customers(FIRST\_NAME string, last\_name string, email\_address string, sex string, last\_ip\_address string, city string, state string, total\_orders int, total\_purchased int, months\_customer int)**

**. . . . . . . . . . . . . . . . . . . . .> row format delimited**

**. . . . . . . . . . . . . . . . . . . . .> fields terminated by ','**

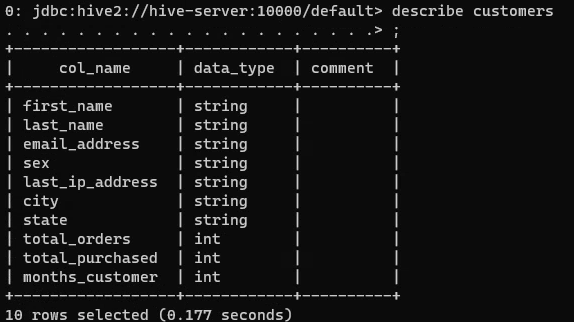
**. . . . . . . . . . . . . . . . . . . . .> TBLPROPERTIES("skip.header.line.count"="1");**

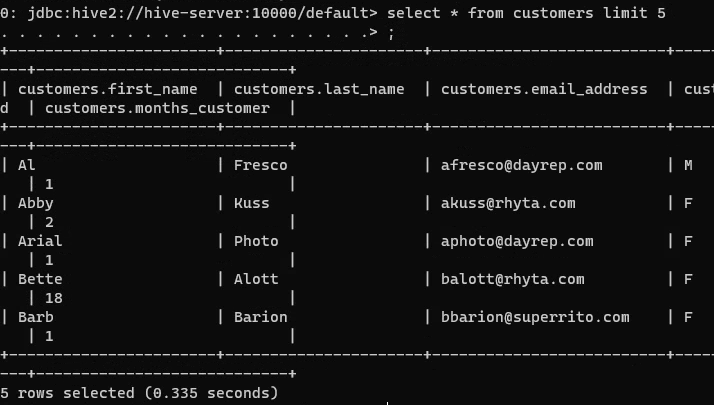
**No rows affected (0.218 seconds)**

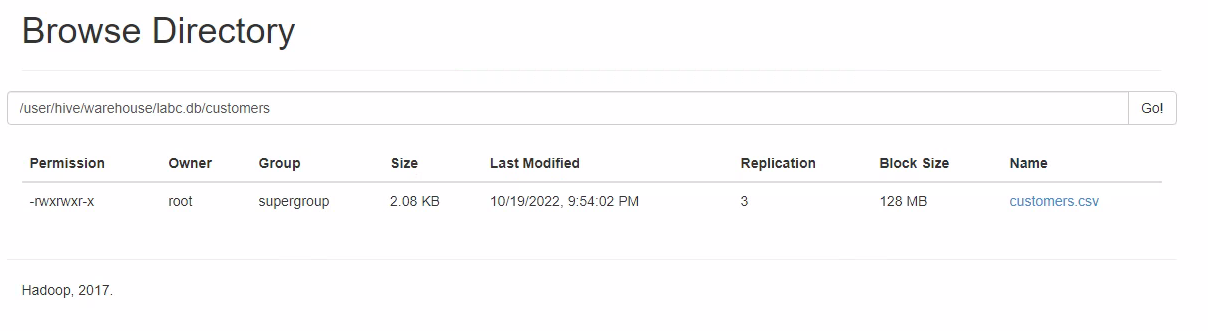
**0: jdbc:hive2://hive-server:10000/default> load data inpath '/user/root/labc/customers/customers.csv'**

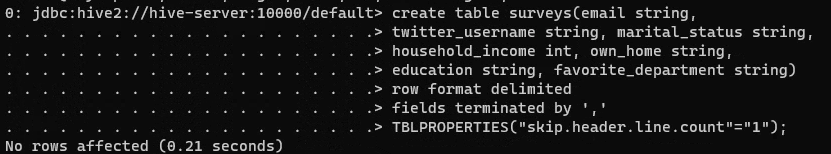
**. . . . . . . . . . . . . . . . . . . . .> overwrite into table customers;**

**No rows affected (0.709 seconds)**

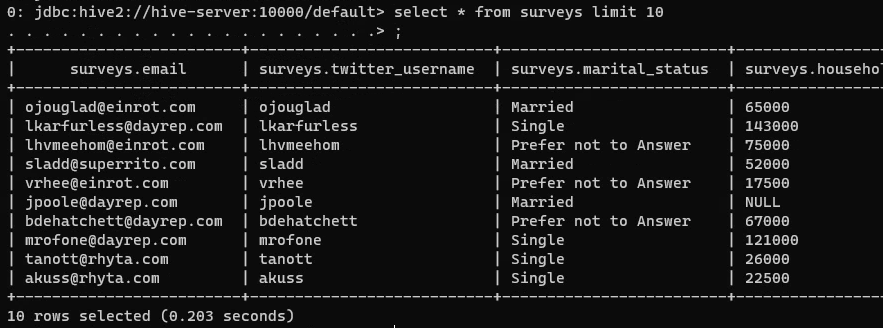
****

****

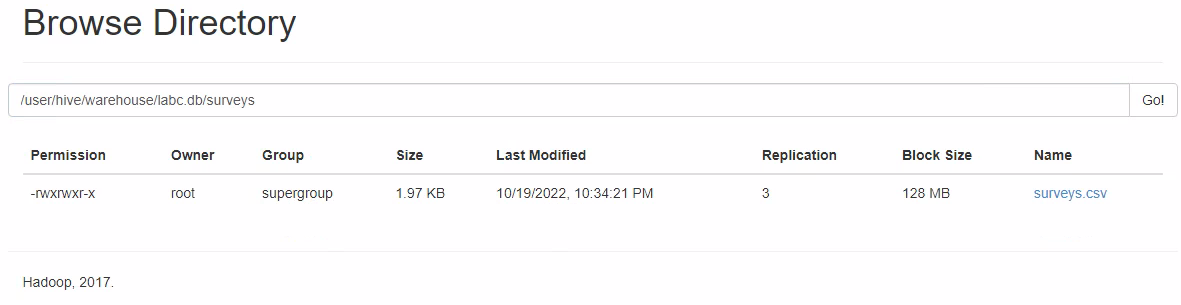
****

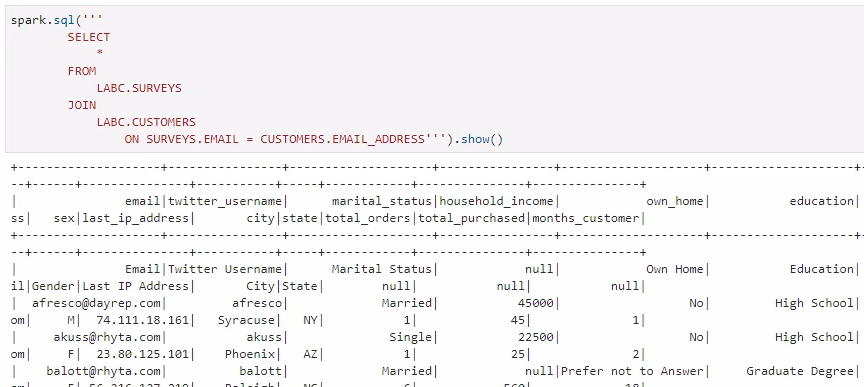
1. Like the previous step, import the surveys.csv into a Hive internal table in the **labc** database called **surveys**. When you have created the table and imported the data, provide all the commands you entered to complete the task, a screenshot of the table description, the select statement output,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



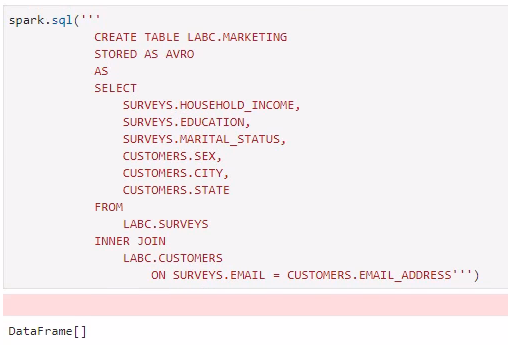


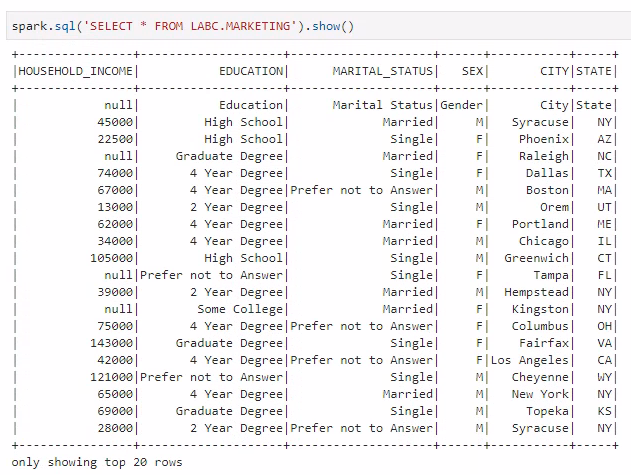




1. Open Jupyter Lab. Create a new notebook called **labc**. Copy over the code from an example to create a Spark session connected to Hive.   
     
   In a separate cell, write Spark SQL code to join customers to surveys on email address. Include all rows and columns and show output in the notebook. 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** Juypter 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.





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.



