Welcome!

Codio Big Data Lab Tutorial

Welcome to the Codio Big Data lab tutorial! Following this tutorial will not only give you examples on how to get started with some of the tools provided in the Codio Big Data lab environment, but also give you a taste of what it means to ask bigger questions. By the end of this tutorial you will:

- Understand how to use some Big Data tools
- Know how to setup and execute some basic business intelligence and analytics use cases
- Be able to explain to your manager why they need to give you a raise!

Getting Started

Define a Business Question

In this short tutorial you are presented examples in the context of a made-up corporation called DataCo, and your mission is to help the organization get better insight by asking bigger questions.

Scenario:

Your Management: is talking euphorically about Big Data...

You: are carefully skeptical, as it will most likely all land on your desk anyway. Alternatively, it has already landed on you, with the nice project description of: Go figure this Hadoop thing out...

Good to Know

Any successful PoC needs to address something your organization cares about. Hence, the first thing you need to do is to: ***define a business question***.

It won't just impress your manager that you think big and have perspective on the business needs of your organization (which in English means you just helped your manager to look good in front of his management). It will also help you to go through a well scoped PoC and get the investments you need to be successful.

Without a well-defined question, you won't know how to properly model your data, i.e. what structure to apply at query time, or what data sets and tools to use to best serve the use case.

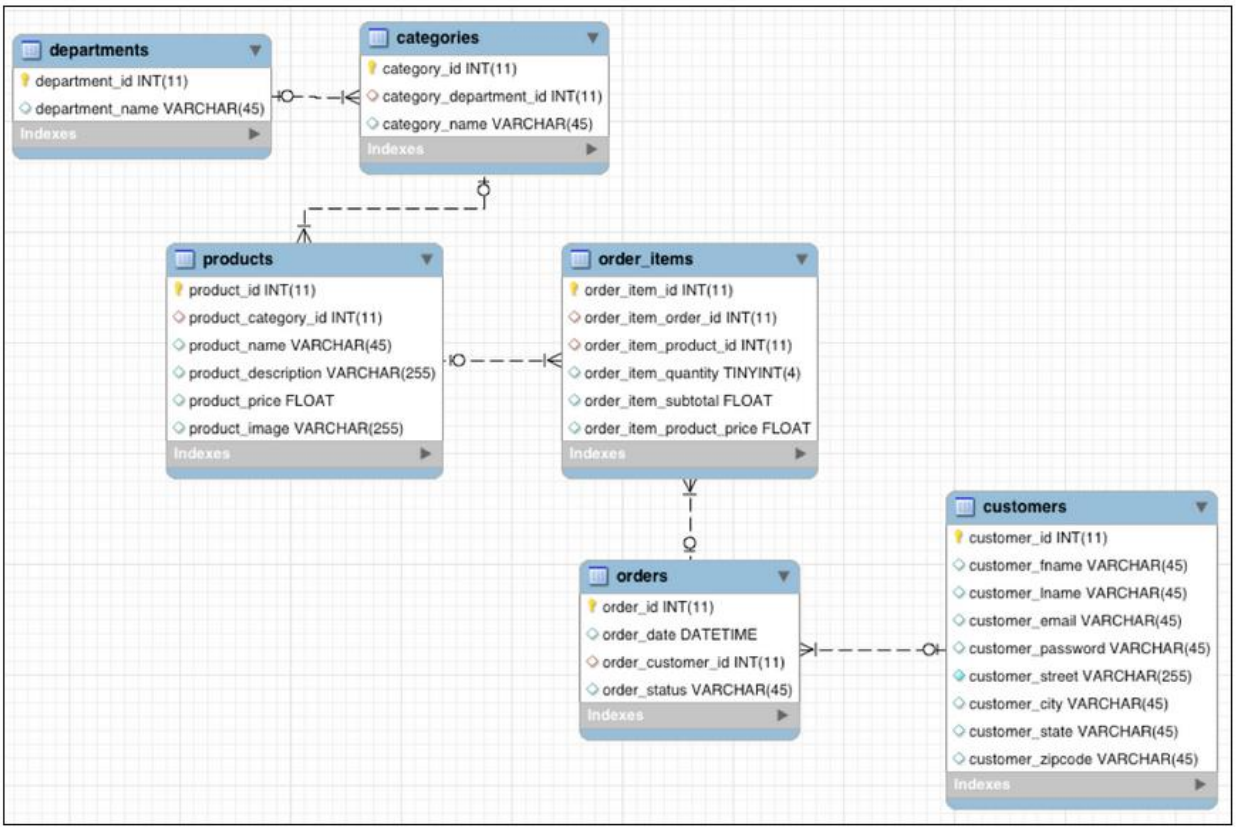
Lab Exercise 1

Ingest and Query Relational Data

In this scenario, DataCo's business question is: What products do our customers like to buy? To answer this question, the first thought might be to look at the transaction data, which should indicate what customers actually do buy and like to buy, right?

This is probably something you can do in your regular RDBMS environment, but a benefit with a Big Data platform is that you can do it at greater scale at lower cost, on the same system that you may also use for many other types of analysis.

What this exercise demonstrates is how to do exactly the same thing you may already know how to do with traditional databases, but in Codio. Seamless integration is important when evaluating any new infrastructure. Hence, it's important to be able to do what you normally do, and not break any regular business intelligence (BI) reports or workloads over the dataset you plan to migrate.



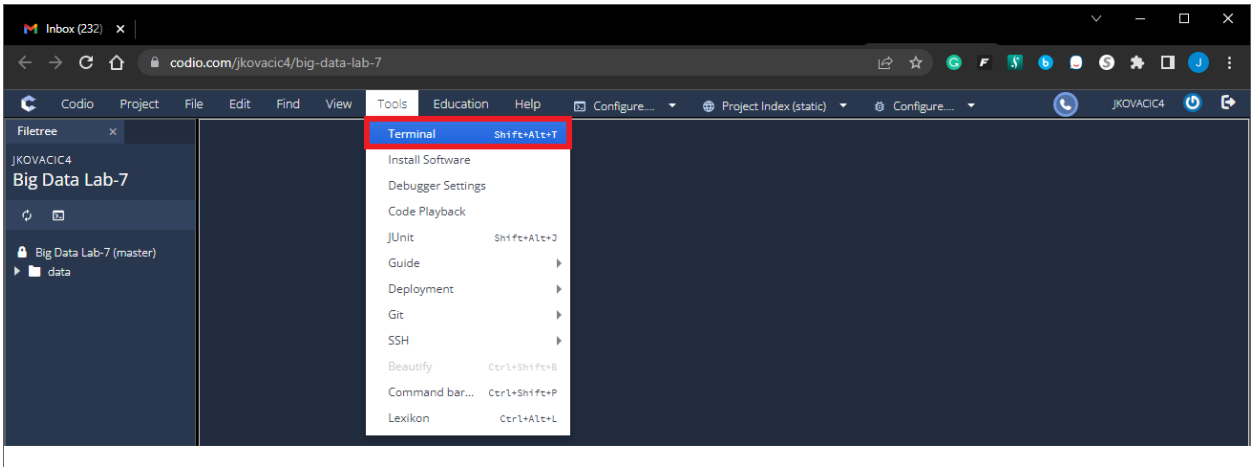
About Sqoop:

Apache Sqoop is a tool that uses MapReduce to transfer data between Hadoop clusters and relational databases very efficiently. It works by spawning tasks on multiple data nodes to download various portions of the data in parallel. When you're finished, each piece of data is replicated to ensure reliability, and spread out across the cluster to ensure you can process it in parallel on your cluster.

To analyze the transaction data in the new platform, we need to ingest it into the Hadoop Distributed File System (HDFS). We need to find a tool that easily transfers structured data from a RDBMS to HDFS, while preserving structure. That enables us to query the data, but not interfere with or break any regular workload on it.

Apache Sqoop is that tool. The nice thing about Sqoop is that we can automatically load our relational data from MySQL into HDFS, while preserving the structure.

You must first open a terminal, which you can do by clicking the "**Tools → Terminal**" from the top menu in your Codio Big Data Lab environment.



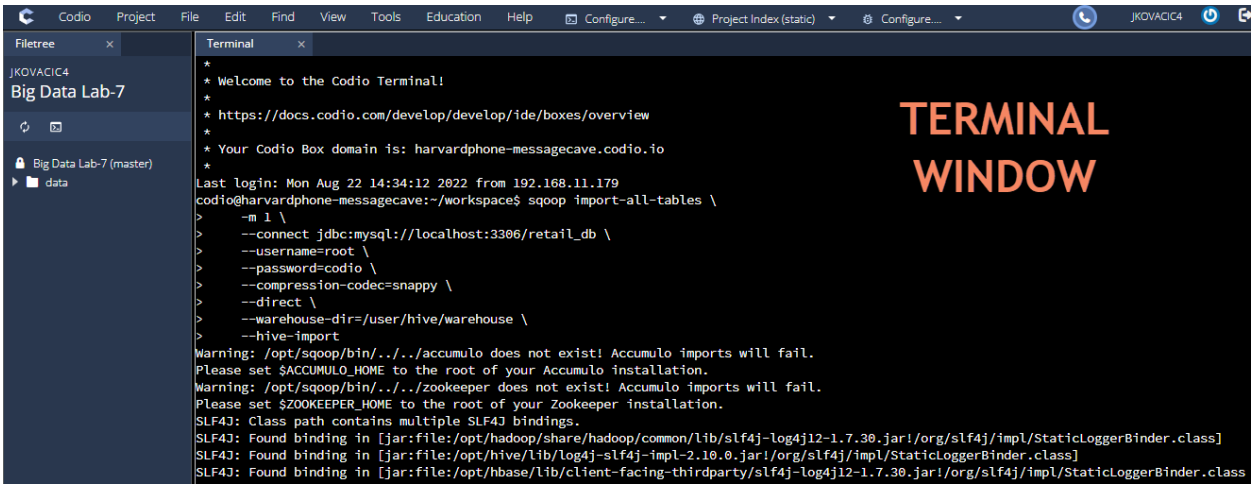
Once the Terminal window is open, execute the command presented below followed by pressing the **ENTER** key at the Linux prompt to start logging your session in the **screen.log** file:

```
script screen.log
```

```
Last login: Mon Aug 22 14:34:12 2022 from 192.168.11.179
codio@amazonpress-atlasvalue:~/workspace$ script screen.log
Script started, file is screen.log
codio@amazonpress-atlasvalue:~/workspace$
```

You can then enter the statement/command presented below followed by pressing the **ENTER** key at the Linux prompt to launch the Sqoop data transfer job:

```
sqoop import-all-tables \
    -m 1 \
    --connect jdbc:mysql://localhost:3306/retail_db \
    --username=root \
    --password=codio \
    --compression-codec=snappy \
    --direct \
    --warehouse-dir=/user/hive/warehouse \
    --hive-import
```



This Sqoop command may take a while to complete, but it is doing a lot. It is launching MapReduce jobs to pull the data from our MySQL database and write the data to HDFS, distributed across the cluster in Apache Parquet format. It is also creating tables to represent the HDFS files in Apache Hive with matching schema.

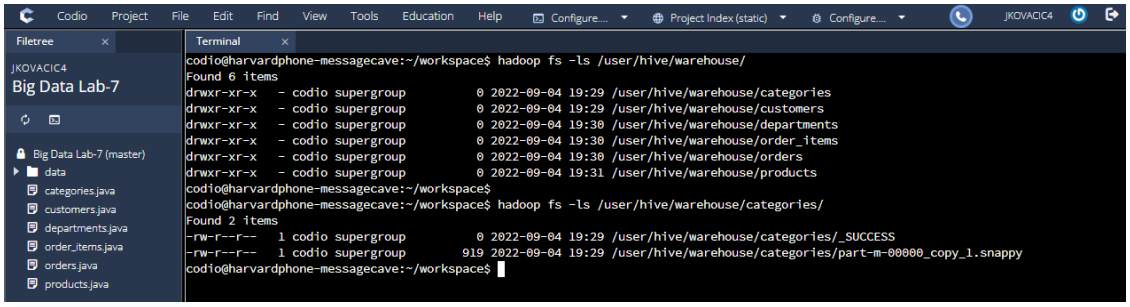
Parquet is a format designed for analytical applications on Hadoop. Instead of grouping your data into rows like typical data formats, it groups your data into columns. This is ideal for many analytical queries where instead of retrieving data from specific records, you're analyzing relationships between specific variables across many records. Parquet is designed to optimize data storage and retrieval in these scenarios.

Once the Sqoop job is complete, we can confirm that our data was imported into HDFS via the following commands at the Linux prompt:

```
hadoop fs -ls /user/hive/warehouse/

hadoop fs -ls /user/hive/warehouse/categories/
```

Press the **ENTER** key on your keyboard after entering each of the above commands.

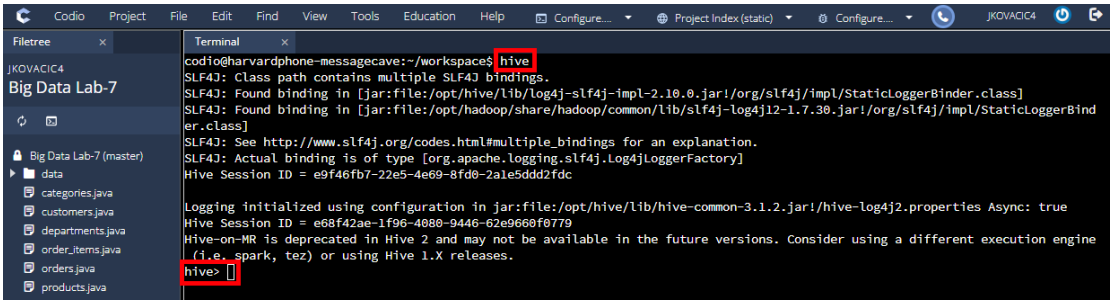
These commands will show the directories and the files inside them that make up our tables:



Note: The number of .parquet files shown will be equal to the number of mappers used by Sqoop. On a single-node you will just see one, but larger clusters will have a greater number of files.

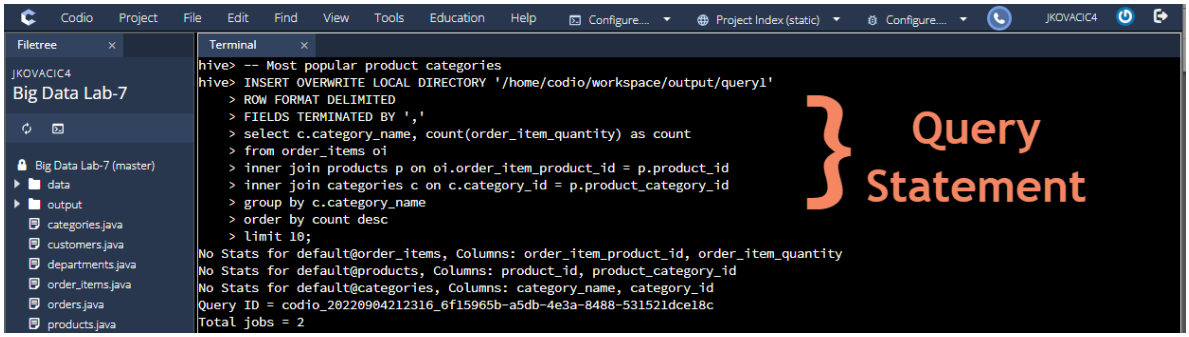
Apache Hive also allows you to create tables by defining a schema over existing files with **'CREATE EXTERNAL TABLE'** statements, similar to traditional relational databases. But Sqoop already created these tables for us, so we can go ahead and query them.

We're going to use the Apache Hive command line interface in the Terminal window to query our tables. Enter **hive** at the Linux command prompt followed by pressing the **ENTER** key to start the application.

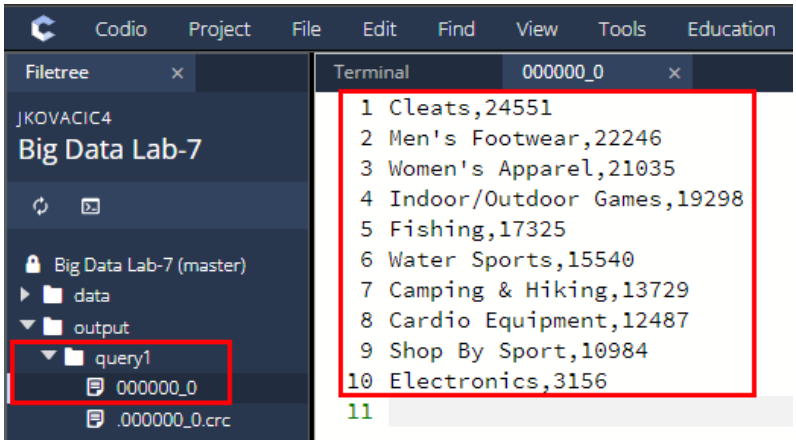


Now that your transaction data is readily available for structured queries in the Codio lab environment, it's time to address DataCo's business question. Enter the query statement presented below followed by pressing the **ENTER** key at the Apache Hive prompt for calculating the total number of ordered items in the most popular product categories:

```
-- Most popular product categories
INSERT OVERWRITE LOCAL DIRECTORY
'/home/codio/workspace/output/query1'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
select c.category_name, count(order_item_quantity) as count
from order_items oi
inner join products p on oi.order_item_product_id = p.product_id
inner join categories c on c.category_id = p.product_category_id
group by c.category_name
order by count desc
limit 10;
```

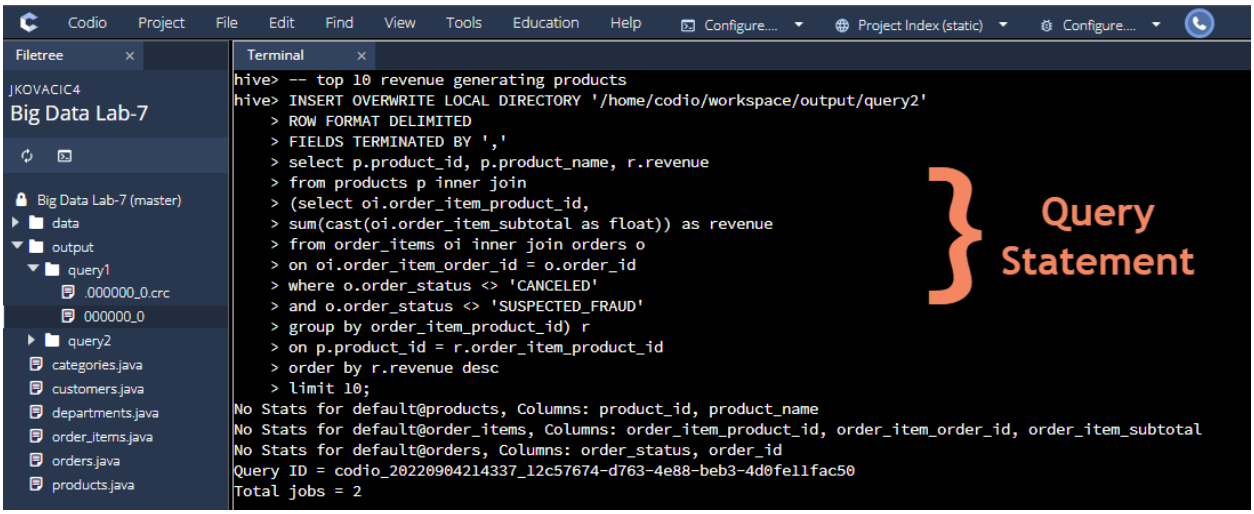


The output will be placed into a file contained in the **`/home/codio/workspace/output/query1`** folder. The raw output can be viewed by selecting the output file from the left file tree.

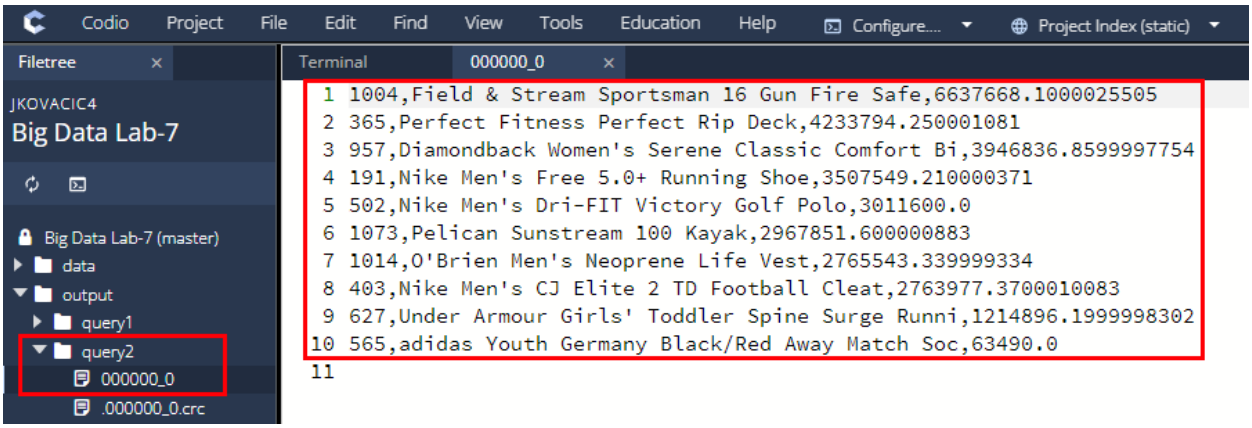


Enter the query statement presented below followed by pressing the **ENTER** key at the Apache Hive prompt in the Terminal window to obtain the top ten revenue generating products:

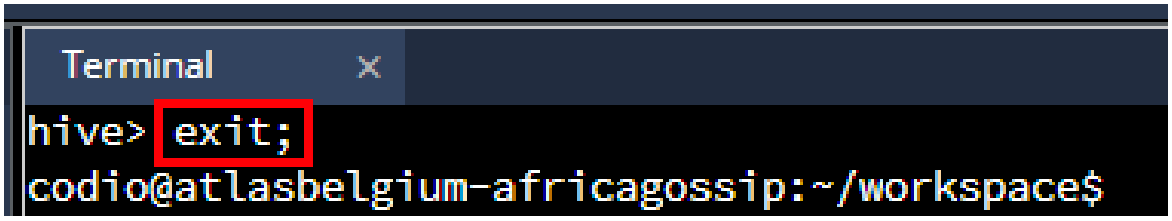
```
-- top 10 revenue generating products
INSERT OVERWRITE LOCAL DIRECTORY
'/home/codio/workspace/output/query2'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
select p.product_id, p.product_name, r.revenue
from products p inner join
(select oi.order_item_product_id,
sum(cast(oi.order_item_subtotal as float)) as revenue
from order_items oi inner join orders o
on oi.order_item_order_id = o.order_id
where o.order_status <> 'CANCELED'
and o.order_status <> 'SUSPECTED_FRAUD'
group by order_item_product_id) r
on p.product_id = r.order_item_product_id
order by r.revenue desc
limit 10;
```



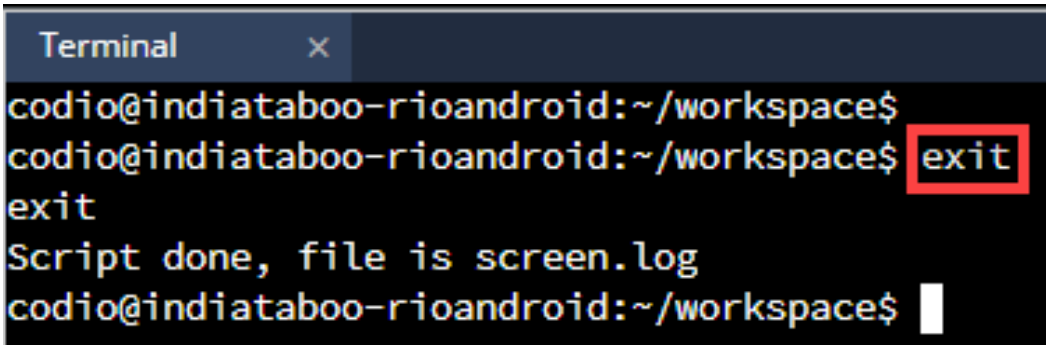
The output will be placed into a file contained in the **`/home/codio/workspace/output/query2`** folder. The raw output can be viewed by selecting the output file from the left file tree.



Enter **`exit;`** at the Apache Hive prompt in the Terminal window and then press the **ENTER** key to exit the Apache Hive application.



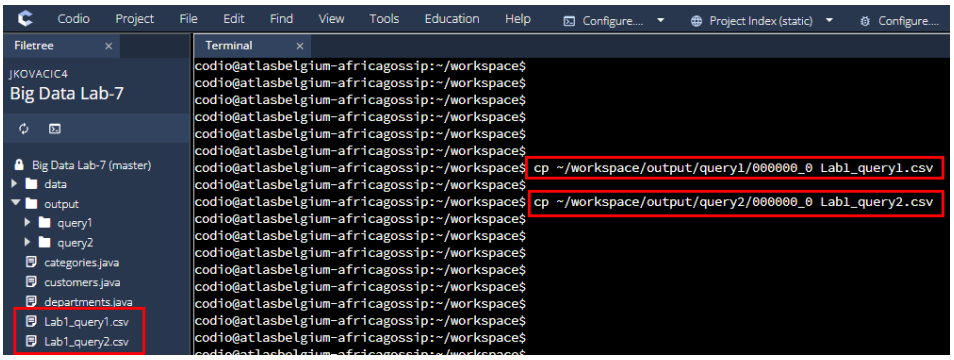
Enter **`exit`** at the Linux Terminal prompt and then press the **ENTER** key to stop writing to the **screen.log** file.



Enter the following commands at the Linux prompt in the Terminal window to copy the query output files to the **workspace** folder:

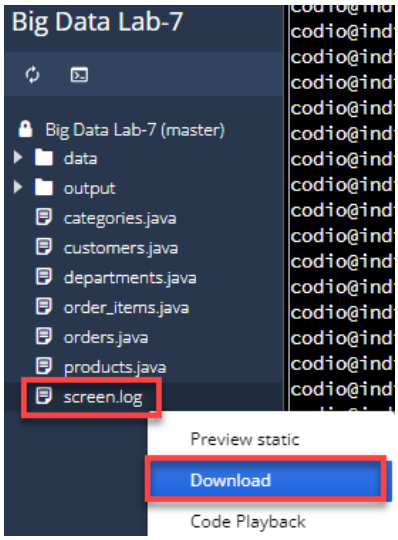
```
cp ~/workspace/output/query1/000000_0 Lab1_query1.csv

cp ~/workspace/output/query2/000000_0 Lab1_query2.csv
```

Press the **ENTER** key after entering each of the above statements to execute them.

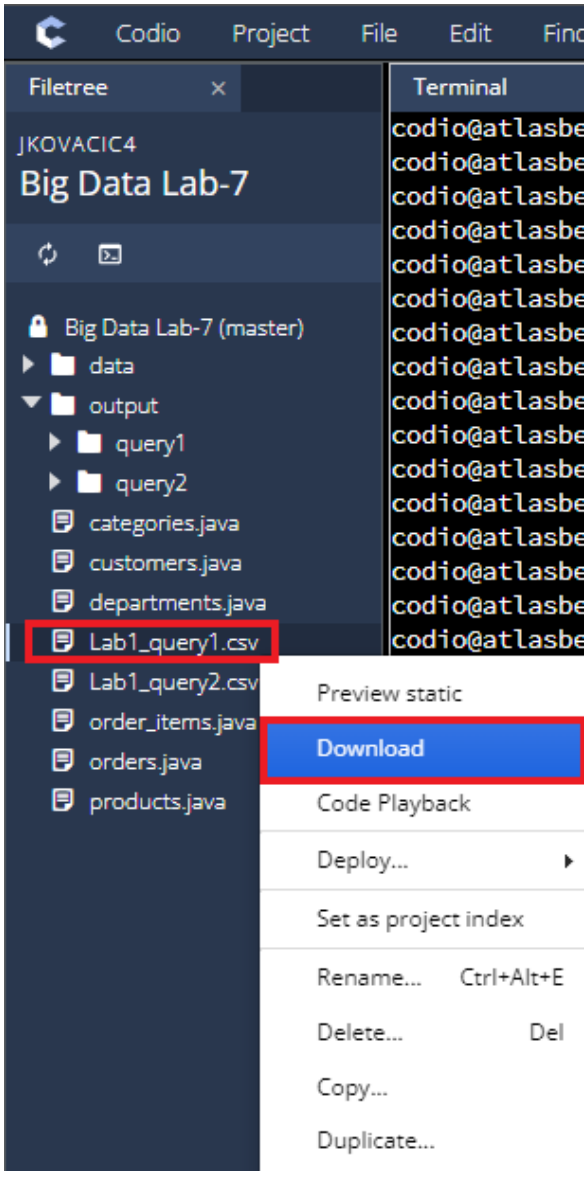


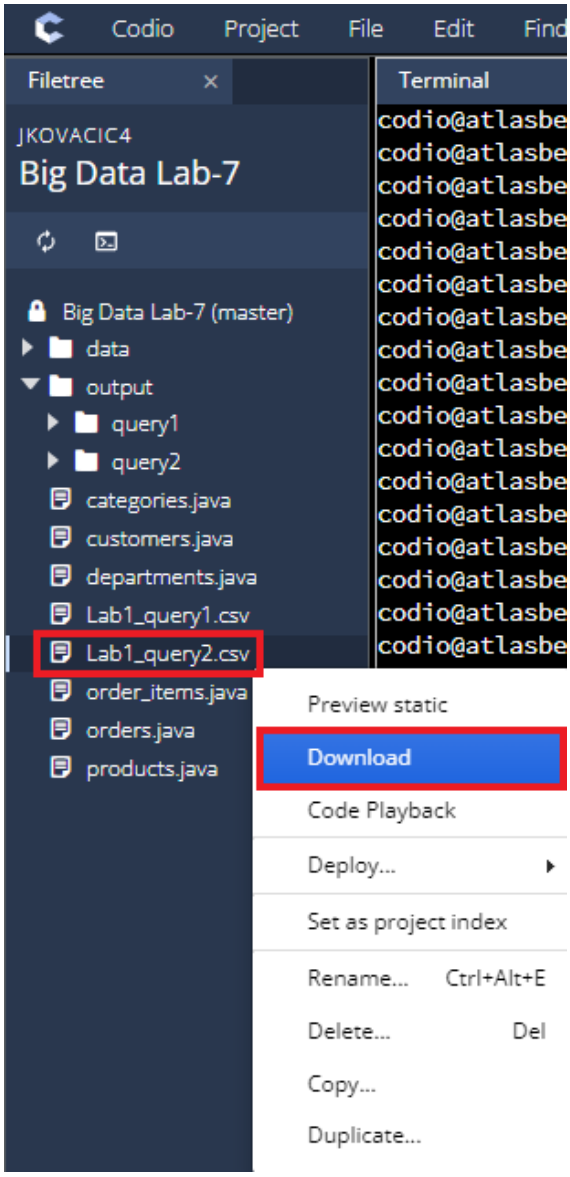
Please note the query output was placed in comma-separated values (CSV) format. Thus, the copied file names in the **workspace** folder have been given a .csv extension.

In the Codio file tree, right-click on the **screen.log** file and download it to an accessible location on your personal computer system. You will need to provide this file in your assignment submittal. This file can be viewed in a text editor like Microsoft Notepad.



In the Codio file tree, right-click on each of the CSV files (**Lab1_query1.csv** and **Lab1_query2.csv**) and download them to an accessible location on your personal computer system. You will need to provide these CSV files in your assignment submittal.





You can open up the CSV files in Microsoft Excel for viewing. Excel will place the data in a readable format that can be readily analyzed.

Lab1_query1.csv

	A	B
1	Cleats	24551
2	Men's Footwear	22246
3	Women's Apparel	21035
4	Indoor/Outdoor Games	19298
5	Fishing	17325
6	Water Sports	15540
7	Camping & Hiking	13729
8	Cardio Equipment	12487
9	Shop By Sport	10984
10	Electronics	3156

Lab1_query2.csv

	A	B	C
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668
2	365	Perfect Fitness Perfect Rip Deck	4233794
3	957	Diamondback Women's Serene Classic Comfort Bi	3946837
4	191	Nike Men's Free 5.0+ Running Shoe	3507549
5	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
6	1073	Pelican Sunstream 100 Kayak	2967852
7	1014	O'Brien Men's Neoprene Life Vest	2765543
8	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977
9	627	Under Armour Girls' Toddler Spine Surge Runni	1214896
10	565	adidas Youth Germany Black/Red Away Match Soc	63490

CONCLUSION

Now you have gone through the first basic steps to Sqoop structured data into HDFS, transform it into Avro file format, and import the schema files for use when we query this data.

You have learned how to create and query tables using Apache Hive and that you can use regular interfaces and tools (such as SQL) within a Hadoop environment as well. The idea here being you can do the same reports you usually do, but where the architecture of Hadoop vs traditional systems provides much larger scale and flexibility.

Showing Big Data Value

Going a Step Beyond

Scenario:

Your Management: is indifferent, you produced what you always produce - a report on structured data, but you really didn't prove any additional value.

You: are either also indifferent and just go back to what you have always done... or you have an ace up your sleeve...

Lab Exercise 2

Correlate Structured Data with Unstructured Data

Since you are a pretty smart data person, you realize another interesting business question would be: are the most viewed products also the most sold? (or for other scenarios, the most searched for, the most chatted about…). Since Hadoop can store unstructured and semi-structured data alongside structured data without remodeling an entire database, you can just as well ingest, store and process web log events. Let's find out what site visitors have actually viewed the most.

For this, you need the web clickstream data. The most common way to ingest web clickstream is to use Flume. Flume is a scalable real-time ingest framework that allows you to route, filter, aggregate, and do "mini-operations" on data on its way into the scalable processing platform.

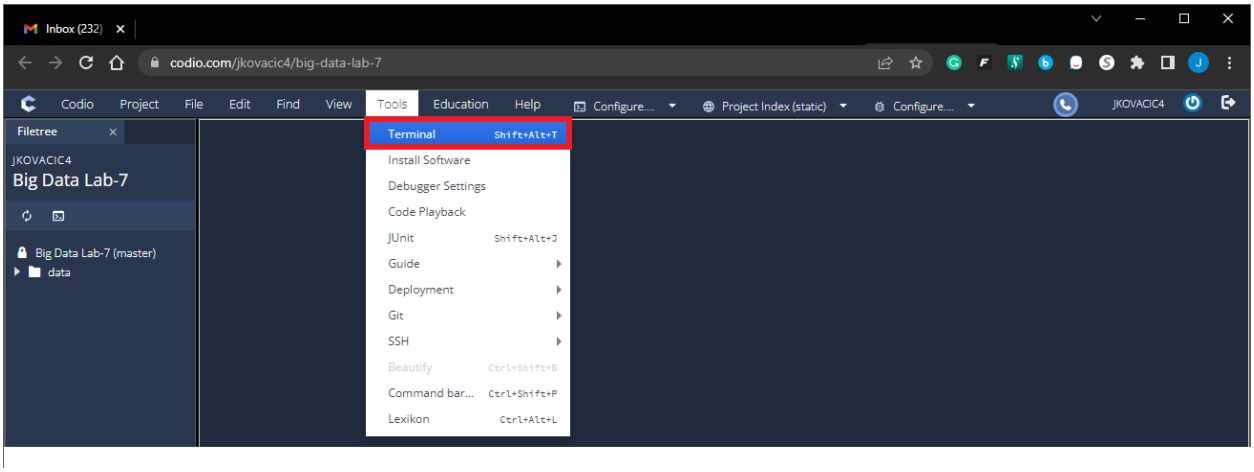
In this exercise you will bulk upload an existing web clickstream data set into HDFS directly.

Bulk Upload Data

For convenience, we have loaded a sample (about 180K lines) set of access log data into the`/home/codio/workspace/data/access.log.2` file in your Codio lab environment.

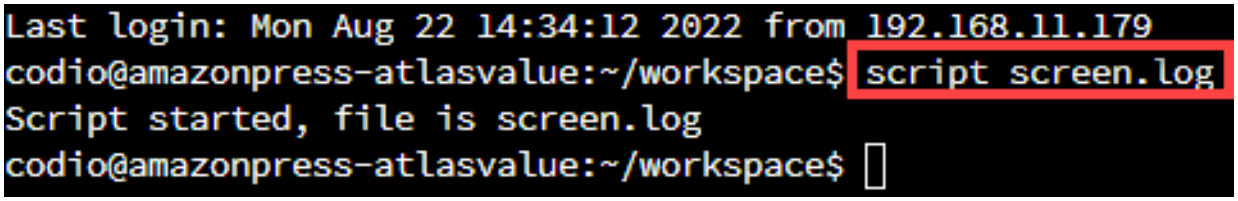
Let's move this data from the local filesystem into HDFS.

You must first open a terminal, which you can do by clicking the "**Tools → Terminal**" from the top menu in your Codio Big Data Lab environment.



Execute the command specified below followed by pressing the **ENTER** key at the Linux Terminal prompt to start logging your session in the **screen.log** file:

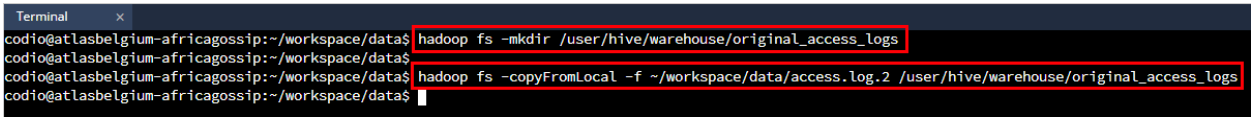
```
script screen.log
```



Go into your Codio Terminal window and execute the following commands at the Linux prompt:

```
hadoop fs -mkdir /user/hive/warehouse/original_access_logs

hadoop fs -copyFromLocal -f ~/workspace/data/access.log.2
/user/hive/warehouse/original_access_logs
```

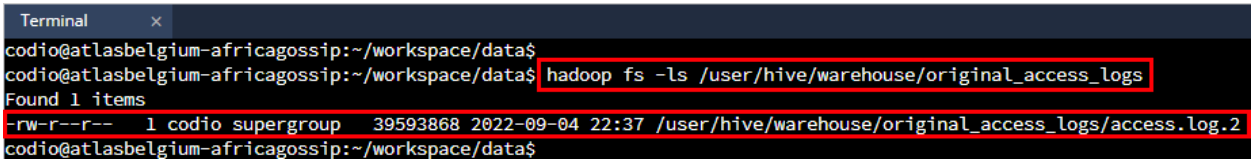
Press the **ENTER** key on your keyboard after you have entered each individual command. The copy command may take several minutes to complete.



Verify that your data is in HDFS by entering the following command and pressing the **ENTER** key at the Linux Terminal prompt:

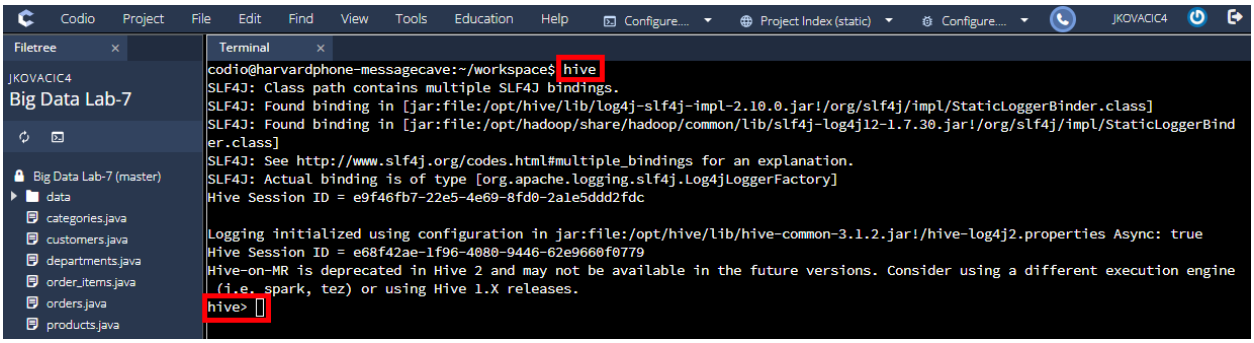
```
hadoop fs -ls /user/hive/warehouse/original_access_logs
```

You should see a result similar to the following:



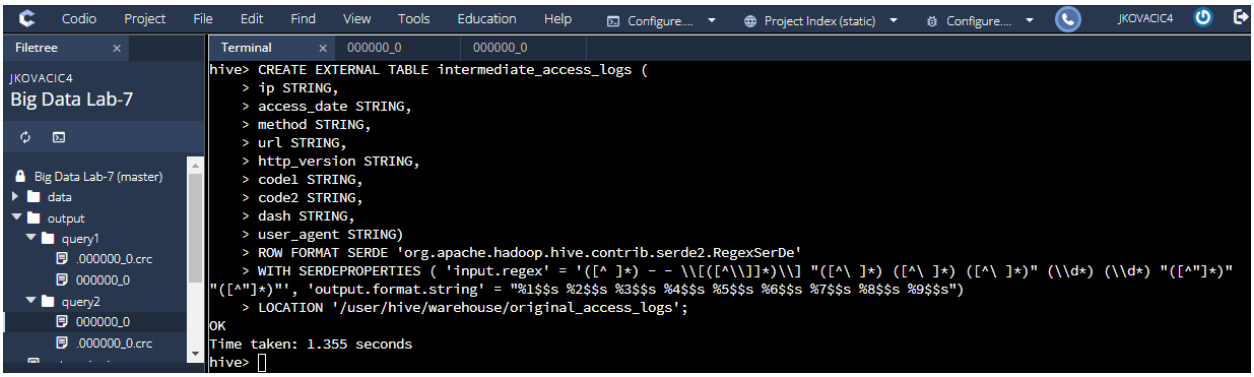
Now you can build a table in Apache Hive and then query the data. You'll build this table in 2 steps. First, you'll take advantage of Apache Hive's flexible SerDes (serializers / deserializers) to parse the logs into individual fields using a regular expression. Second, you'll transfer the data from this intermediate table to one that does not require any special SerDes. Once the data is in this table, you can query and examine it much more easily.

Start the Apache Hive application by entering **hive** at the Linux command prompt in the Terminal window followed by pressing the **ENTER** key.



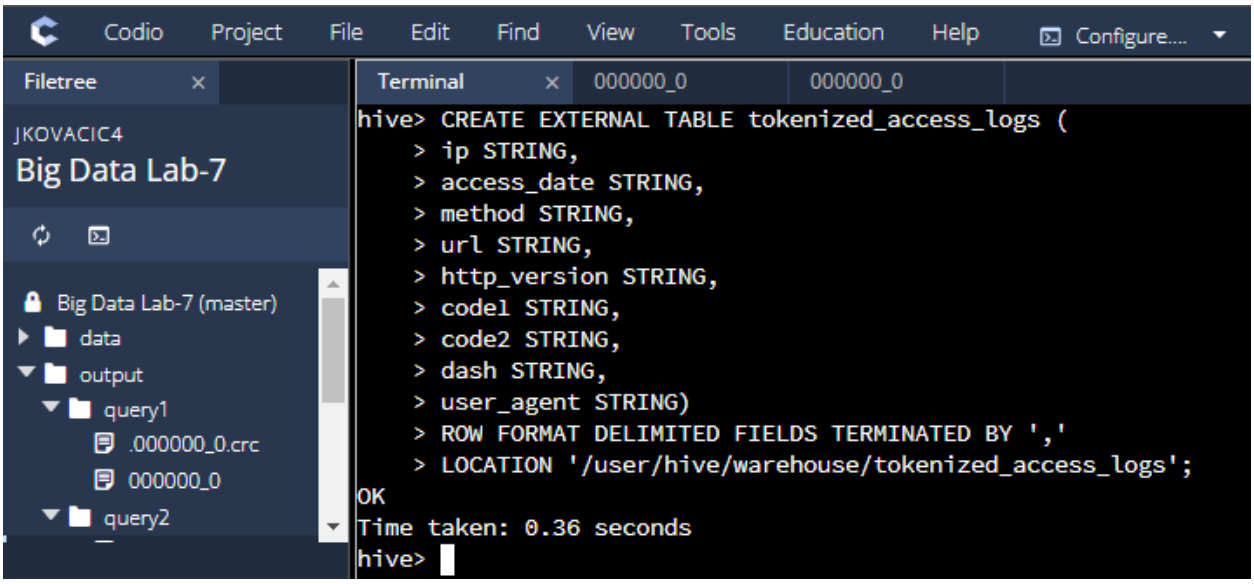
Enter the code presented below followed by pressing the **ENTER** key at the Apache Hive prompt to establish the intermediate access log data table:

```
CREATE EXTERNAL TABLE intermediate_access_logs (
ip STRING,
access_date STRING,
method STRING,
url STRING,
http_version STRING,
code1 STRING,
code2 STRING,
dash STRING,
user_agent STRING)
ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'
WITH SERDEPROPERTIES ( 'input.regex' = '([^ ]*) - - \\[([^\\]]*)\\]
 "([^\ ]*) ([^\ ]*) ([^\ ]*)" (\\d*) (\\d*) "([^"]*)" "([^"]*)"',
 'output.format.string' = "%1$$s %2$$s %3$$s %4$$s %5$$s %6$$s %7$$s
 %8$$s %9$$s")
LOCATION '/user/hive/warehouse/original_access_logs';
```



Enter the code presented below followed by pressing the **ENTER** key at the Apache Hive prompt in the Terminal window to establish the tokenized access log data table:

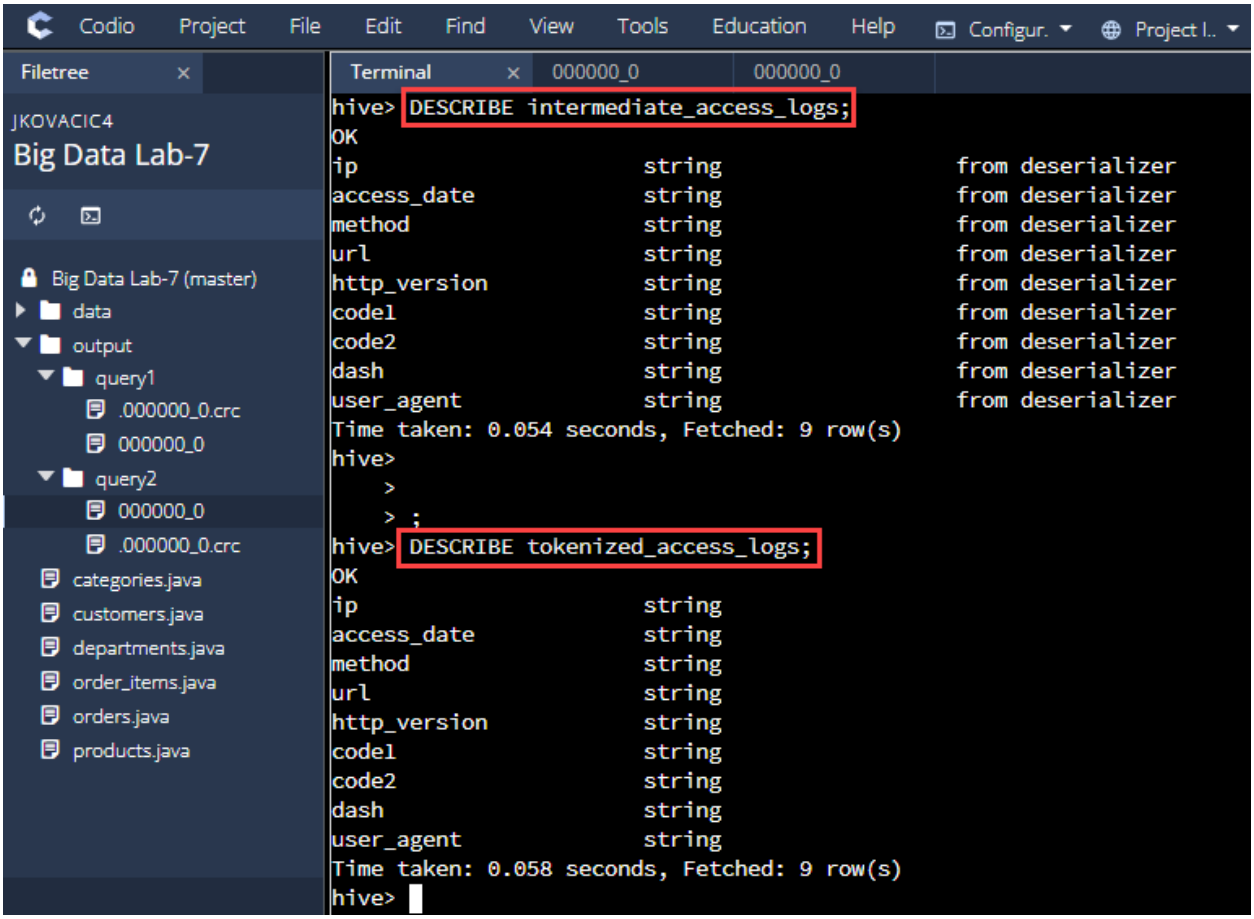
```
CREATE EXTERNAL TABLE tokenized_access_logs (
ip STRING,
access_date STRING,
method STRING,
url STRING,
http_version STRING,
code1 STRING,
code2 STRING,
dash STRING,
user_agent STRING)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/warehouse/tokenized_access_logs';
```



You can verify the creation of the intermediate and tokenized access log data tables using the **DESCRIBE** command at the Apache Hive prompt in the Terminal window.

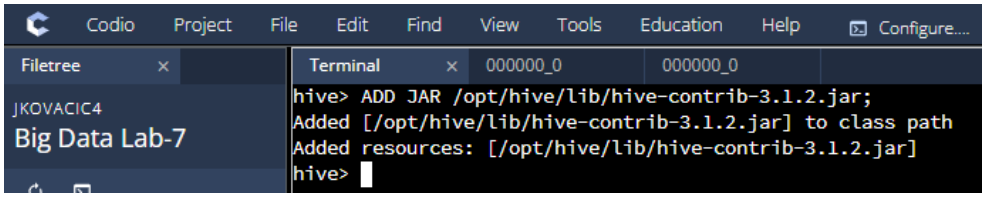
```
DESCRIBE intermediate_access_logs;

DESCRIBE tokenized_access_logs;
```

Press the **ENTER** key on your keyboard after you have entered each of the above individual commands.



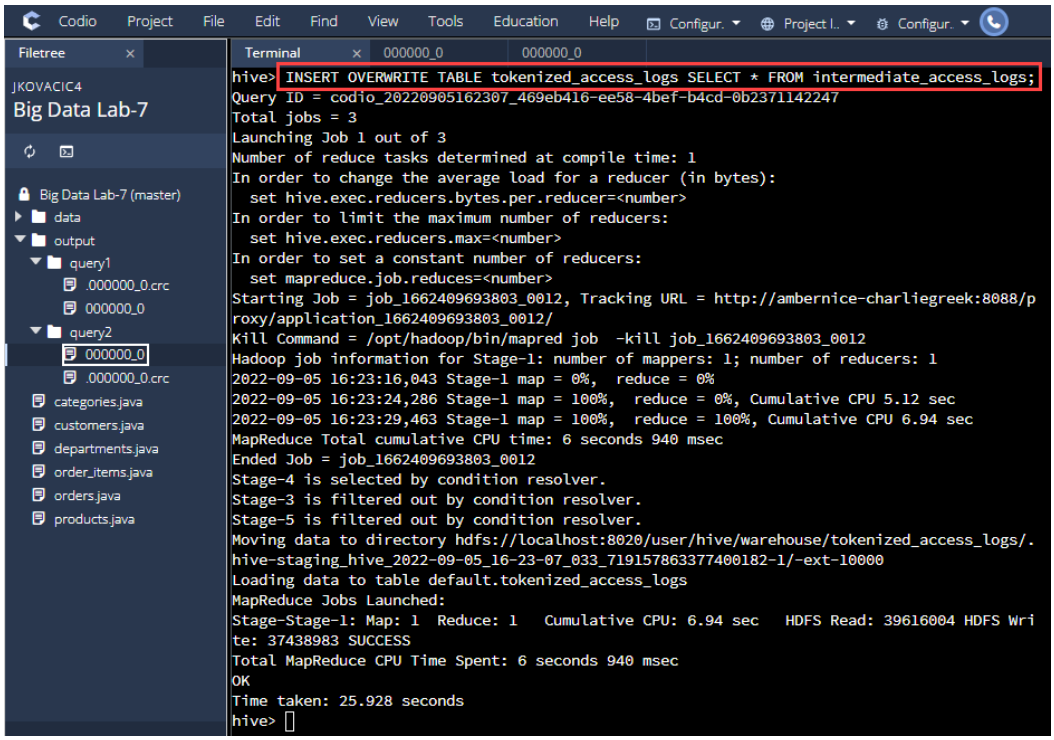
Enter the statement/command presented below followed by pressing the **ENTER** key at the Apache Hive prompt in the Terminal window to enable Apache Hive's flexible SerDes (serializers / deserializers) functionality:

```
ADD JAR /opt/hive/lib/hive-contrib-3.1.2.jar;
```



Enter the statement presented below followed by pressing the **ENTER** key at the Apache Hive prompt in the Terminal window to copy the parsed data from the **intermediate_access_logs** to the **tokenized_access_logs** table:

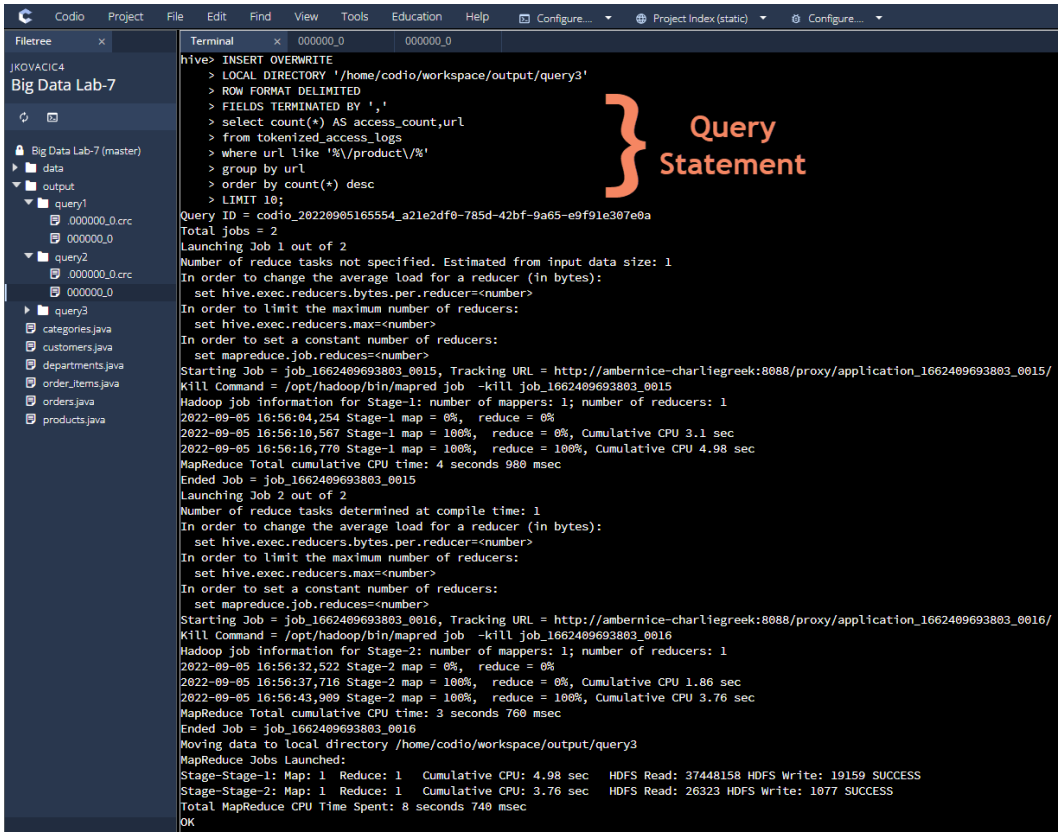
```
INSERT OVERWRITE TABLE tokenized_access_logs
SELECT * FROM intermediate_access_logs;
```



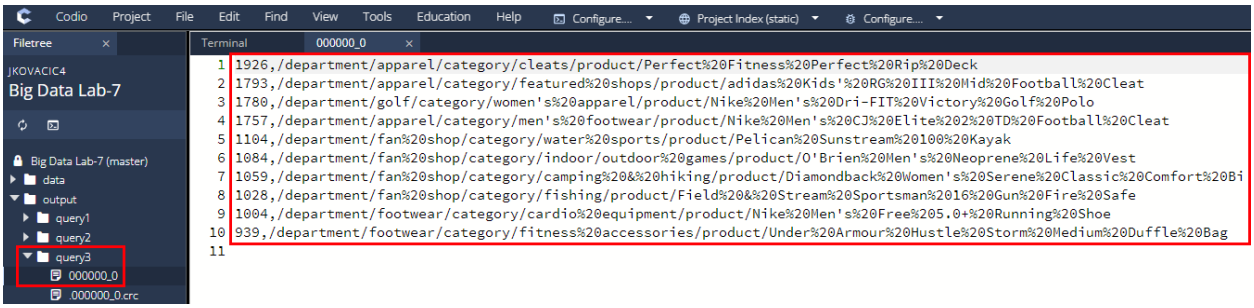
The data transfer operation will take a minute to run. It is using a MapReduce job, just like the Sqoop import did, to transfer the data from one table to the other in parallel.

Enter the query statement below followed by pressing the **ENTER** key at the Apache Hive prompt in the Terminal window to find the top-ten viewed products by site visitors:

```
INSERT OVERWRITE
LOCAL DIRECTORY '/home/codio/workspace/output/query3'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
select count(*) AS access_count,url
from tokenized_access_logs
where url like '%\/product\/%'
group by url
order by count(*) desc
LIMIT 10;
```

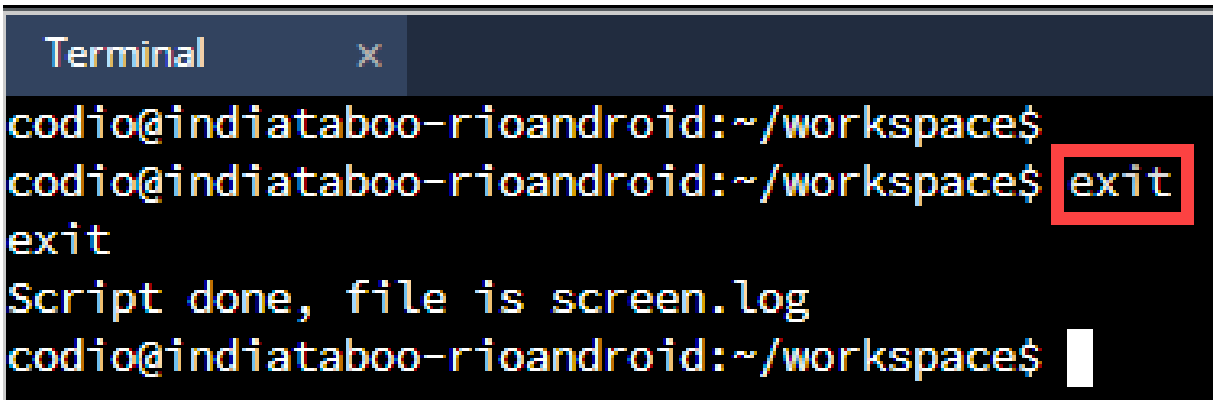


The output will be placed into a file contained in the **`/home/codio/workspace/output/query3`** folder. The raw output can be viewed by selecting the output file from the left file tree.



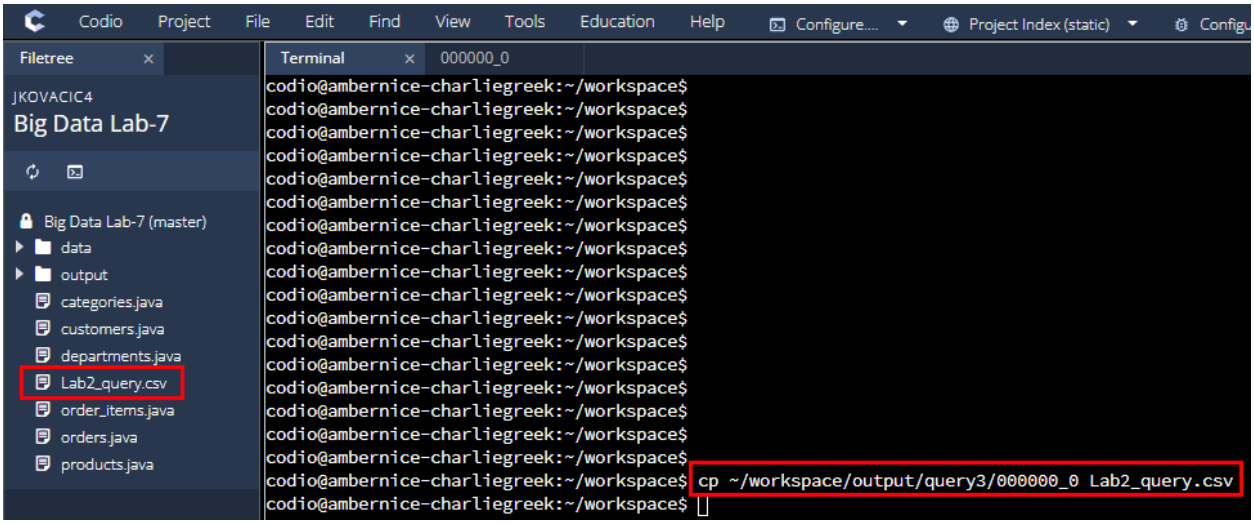
Enter **`exit;`** at the Apache Hive prompt in the Terminal window and then press the **ENTER** key to exit the Apache Hive application.

Enter **`exit`** at the Linux Terminal prompt and then press the **ENTER** key to stop writing to the **screen.log** file.



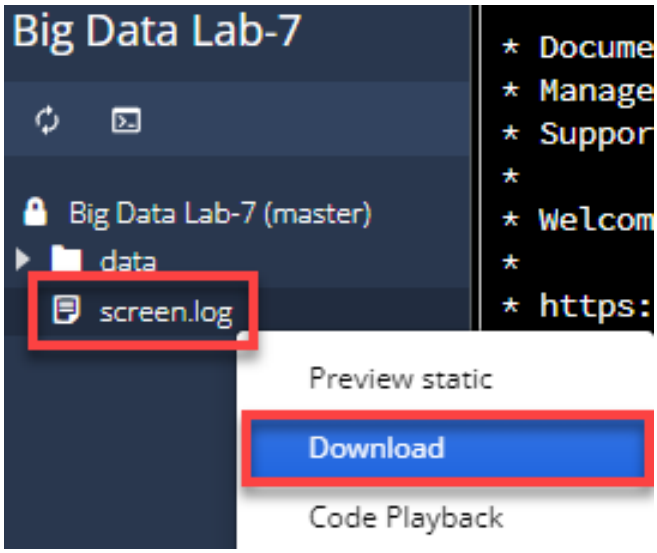
Enter the command presented below followed by pressing the **ENTER** key at the Linux prompt in the Terminal window to copy the query output file to the **workspace** folder:

```
cp ~/workspace/output/query3/000000_0 Lab2_query.csv
```

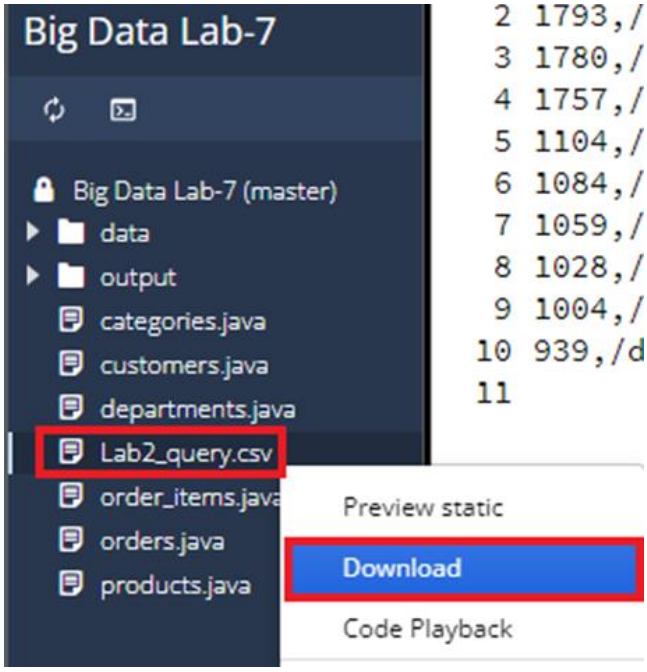


Please note the query output was placed in comma-separated values (CSV) format. Thus, the copied file names in the **workspace** folder have been given a .csv extension.

In the Codio file tree, right-click on the **screen.log** file and download it to an accessible location on your personal computer system. You will need to provide this file in your assignment submittal. This file can be viewed in a text editor like Microsoft Notepad.



In the Codio file tree, right-click on the CSV file (**Lab2_query.csv**) and download it to an accessible location on your personal computer system. You will need to provide this CSV file in your assignment submittal.



You can open up the CSV files in Microsoft Excel for viewing. Excel will place the data in a readable format that can be readily analyzed.

Lab2_query.csv

	A	B
1	1926	/department/apparel/category/cleats/product/Perfect%20Fitness%20Perfect%20Rip%20Deck
2	1793	/department/apparel/category/featured%20shops/product/adidas%20Kids'%20RG%20III%20Mid%20Football%20Cleat
3	1780	/department/golf/category/women's%20apparel/product/Nike%20Men's%20Dri-FIT%20Victory%20Golf%20Polo
4	1757	/department/apparel/category/men's%20footwear/product/Nike%20Men's%20CJ%20Elite%202%20TD%20Football%20Cleat
5	1104	/department/fan%20shop/category/water%20sports/product/Pelican%20Sunstream%20100%20Kayak
6	1084	/department/fan%20shop/category/indoor/outdoor%20games/product/O'Brien%20Men's%20Neoprene%20Life%20Vest
7	1059	/department/fan%20shop/category/camping%20&%20hiking/product/Diamondback%20Women's%20Serene%20Classic%20Comfort%20Bi
8	1028	/department/fan%20shop/category/fishing/product/Field%20&%20Stream%20Sportsman%2016%20Gun%20Fire%20Safe
9	1004	/department/footwear/category/cardio%20equipment/product/Nike%20Men's%20Free%205.0+%20Running%20Shoe
10	939	/department/footwear/category/fitness%20accessories/product/Under%20Armour%20Hustle%20Storm%20Medium%20Duffle%20Bag

By introspecting the results, you quickly realize that this list contains many of the products on the most sold list from previous tutorial steps, but there is one product that did not show up in the previous result. There is one product that seems to be viewed a lot, but never purchased. Why?

	product_id	product_name	revenue
1	1004	Field & Stream Sportsman 16 Gun Fire Safe	6637668.2823181152
2	365	Perfect Fitness Perfect Rip Deck	4233794.3682899475
3	957	Diamondback Women's Serene Classic Comfort Bi	3946837.0045471191
4	191	Nike Men's Free 5.0+ Running Shoe	3507549.2067337036
5	502	Nike Men's Dri-FIT Victory Golf Polo	3011600
6	1073	Pelican Sunstream 100 Kayak	2967851.6815185547
7	1014	O'Brien Men's Neoprene Life Vest	2765543.314743042
8	403	Nike Men's CJ Elite 2 TD Football Cleat	2763977.4868011475
9	627	Under Armour Girls' Toddler Spine Surge Runni	1214896.220287323
10	565	adidas Youth Germany Black/Red Away Match Soc	63490

	count(*)	url	
1	1926	/department/apparel/category/cleats/product/Perfect%20Fitness%20Perfect%20Rip%20Deck	2nd
2	1793	/department/apparel/category/featured%20shops/product/adidas%20Kids'%20RG%20III%20Mid%20Football%20Cleat	MISSING???
3	1780	/department/golf/category/women's%20apparel/product/Nike%20Men's%20Dri-FIT%20Victory%20Golf%20Polo	5th
4	1757	/department/apparel/category/men's%20footwear/product/Nike%20Men's%20CJ%20Elite%202%20TD%20Football%20Cleat	8th
5	1104	/department/fan%20shop/category/water%20sports/product/Pelican%20Sunstream%20100%20Kayak	6th
6	1084	/department/fan%20shop/category/indoor/outdoor%20games/product/O'Brien%20Men's%20Neoprene%20Life%20Vest	7th
7	1059	/department/fan%20shop/category/camping%20&%20hiking/product/Diamondback%20Women's%20Serene%20Classic%20Comfort%20Bi	3rd
8	1028	/department/fan%20shop/category/fishing/product/Field%20&%20Stream%20Sportsman%2016%20Gun%20Fire%20Safe	1st
9	1004	/department/footwear/category/cardio%20equipment/product/Nike%20Men's%20Free%205.0+%20Running%20Shoe	4th
10	939	/department/footwear/category/fitness%20accessories/product/Under%20Armour%20Hustle%20Storm%20Medium%20Duffle%20Bag	> 10th

Well, in our example with DataCo, once these odd findings are presented to your manager, it is immediately escalated. Eventually, someone figures out that on that view page, where most visitors stopped, the sales path of the affected product had a typo in the price for the item. Once the typo was fixed, and a correct price was displayed, the sales for that SKU started to rapidly increase.

CONCLUSION

If you hadn’t had an efficient and interactive tool enabling analytics on high-volume semi-structured data, this loss of revenue would have been missed for a long time. There is risk of loss if an organization looks for answers within partial data. Correlating two data sets for the same business question showed value and being able to do so within the same platform made life easier for you and for the organization.