**Project 5200 – Iowa Liquor Sales Analysis : by Group 1**

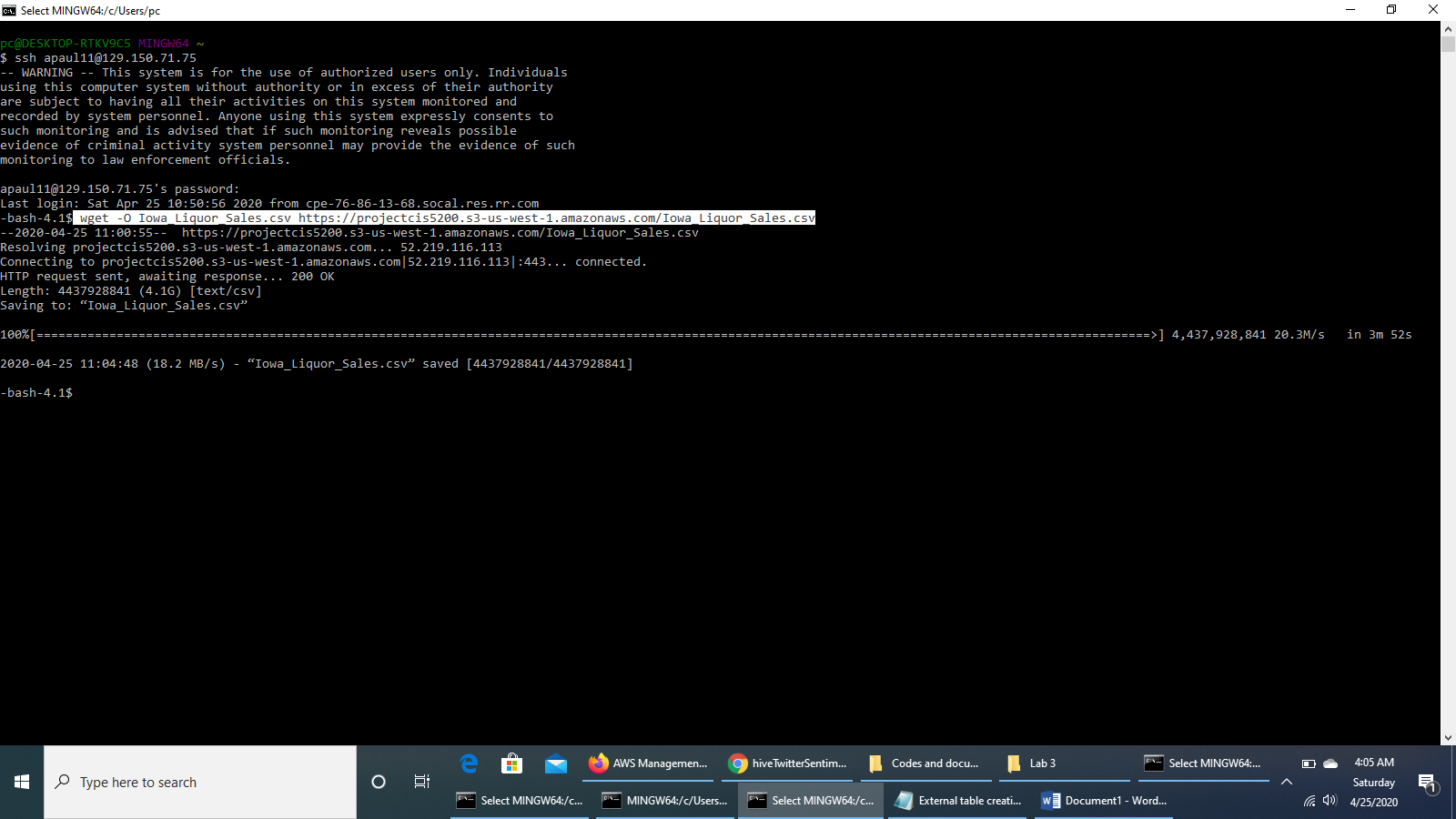
**(Codes and Screenshots)**

**Connect to Oracle Cloud:**

ssh [apaul11@129.150.71.75](mailto:apaul11@129.150.71.75)

**[Note:** We also used alternatively ssh [tkhan9@129.150.71.75](mailto:tkhan9@129.150.71.75) in the later part since apaul11’s account ran out of storage space.**]**

**Download the data file from Amazon S3:**

wget -O iowa\_Liquor\_Sales.csv https://projectcis5200.s3-us-west-1.amazonaws.com/iowa\_Liquor\_Sales.csv 

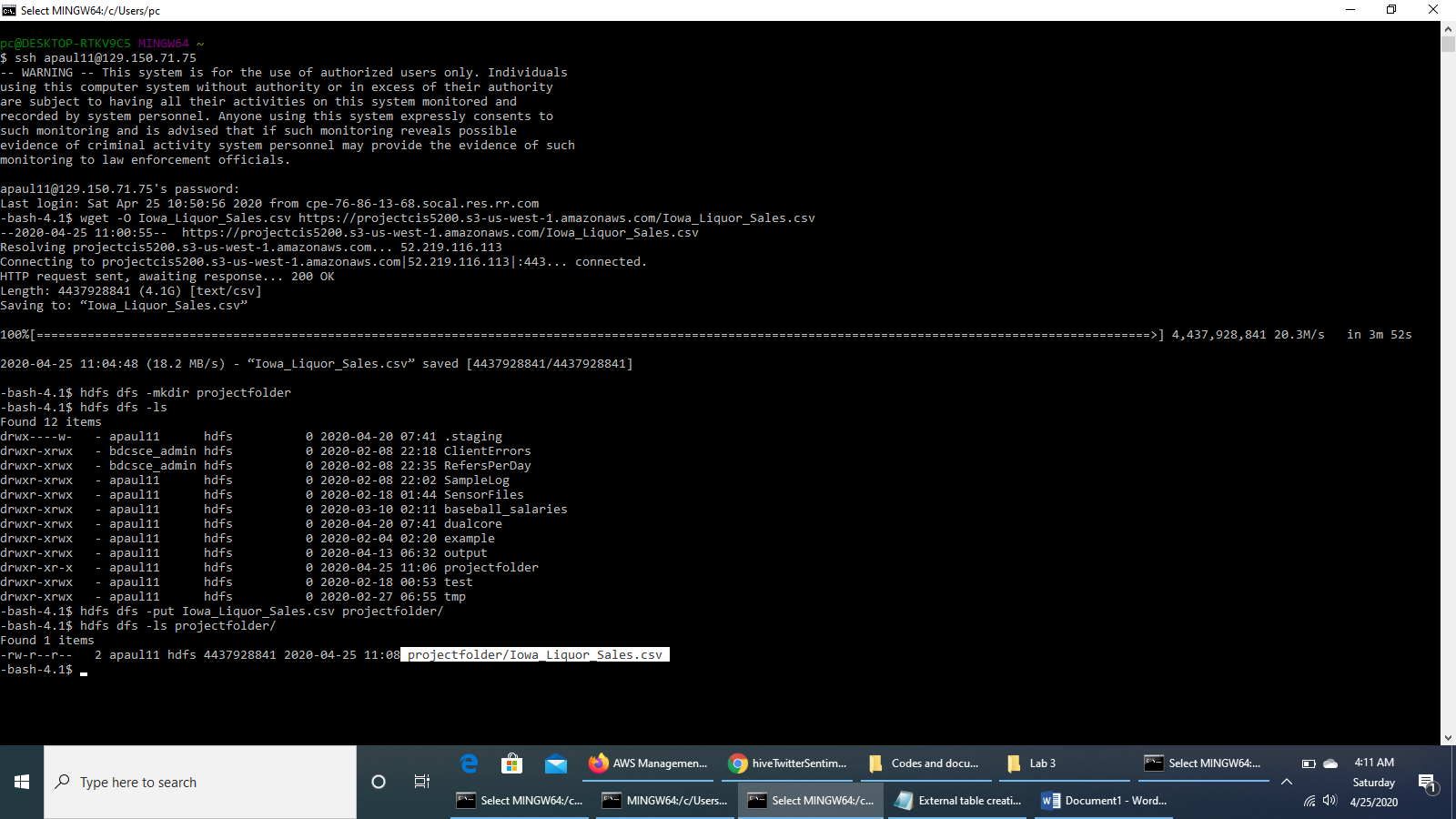
**Upload the Iowa\_Liquor\_Sales.csv file to *projectfolder* directory of HDFS**

hdfs dfs -mkdir projectfolder

hdfs dfs -ls

hdfs dfs -put Iowa\_Liquor\_Sales.csv projectfolder/

hdfs dfs -ls projectfolder/

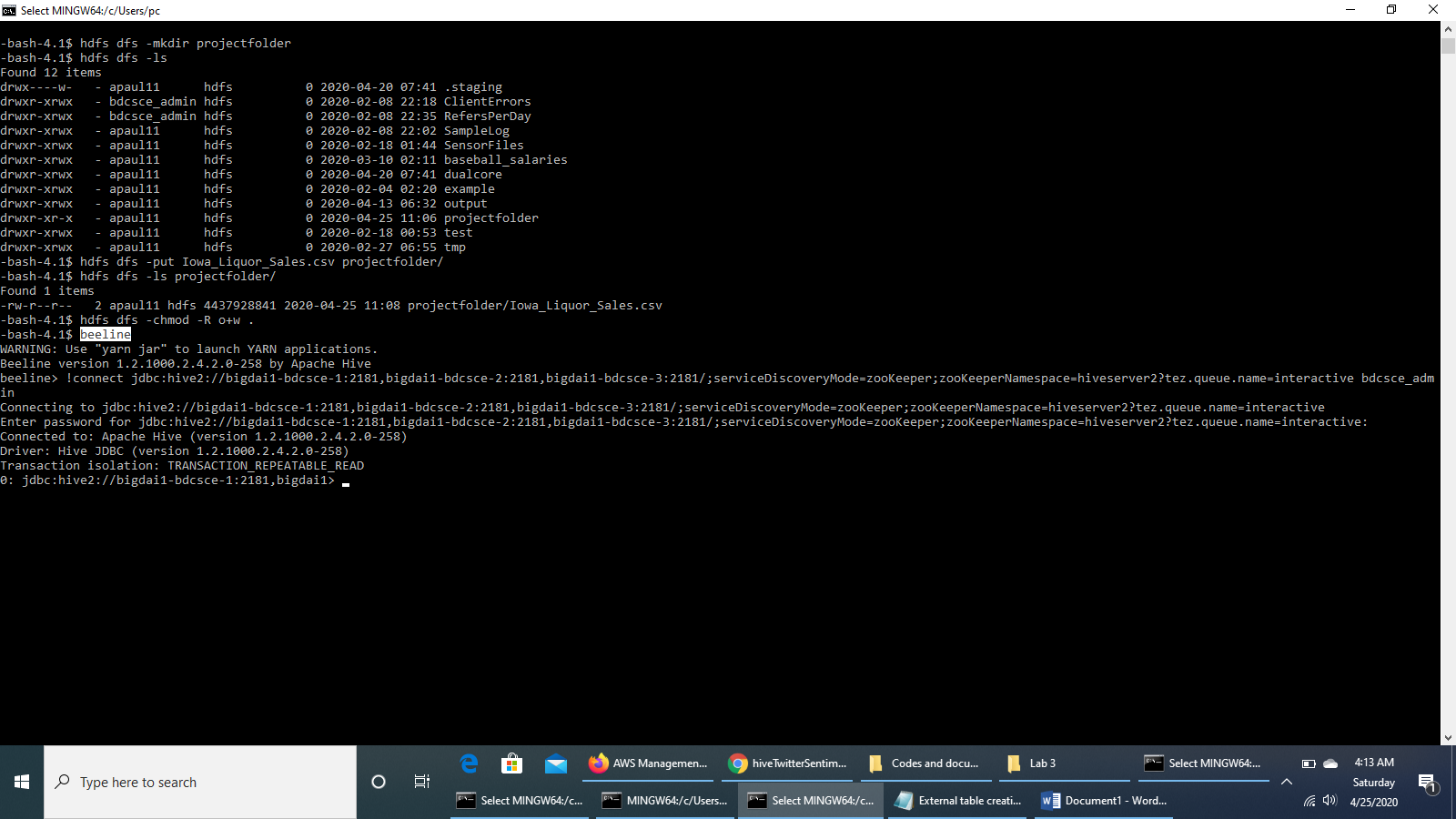


**Run the HDFS command to make beeline command work, open beeline CLI and copy and paste “!connect …” command**

hdfs dfs -chmod -R o+w .

beeline

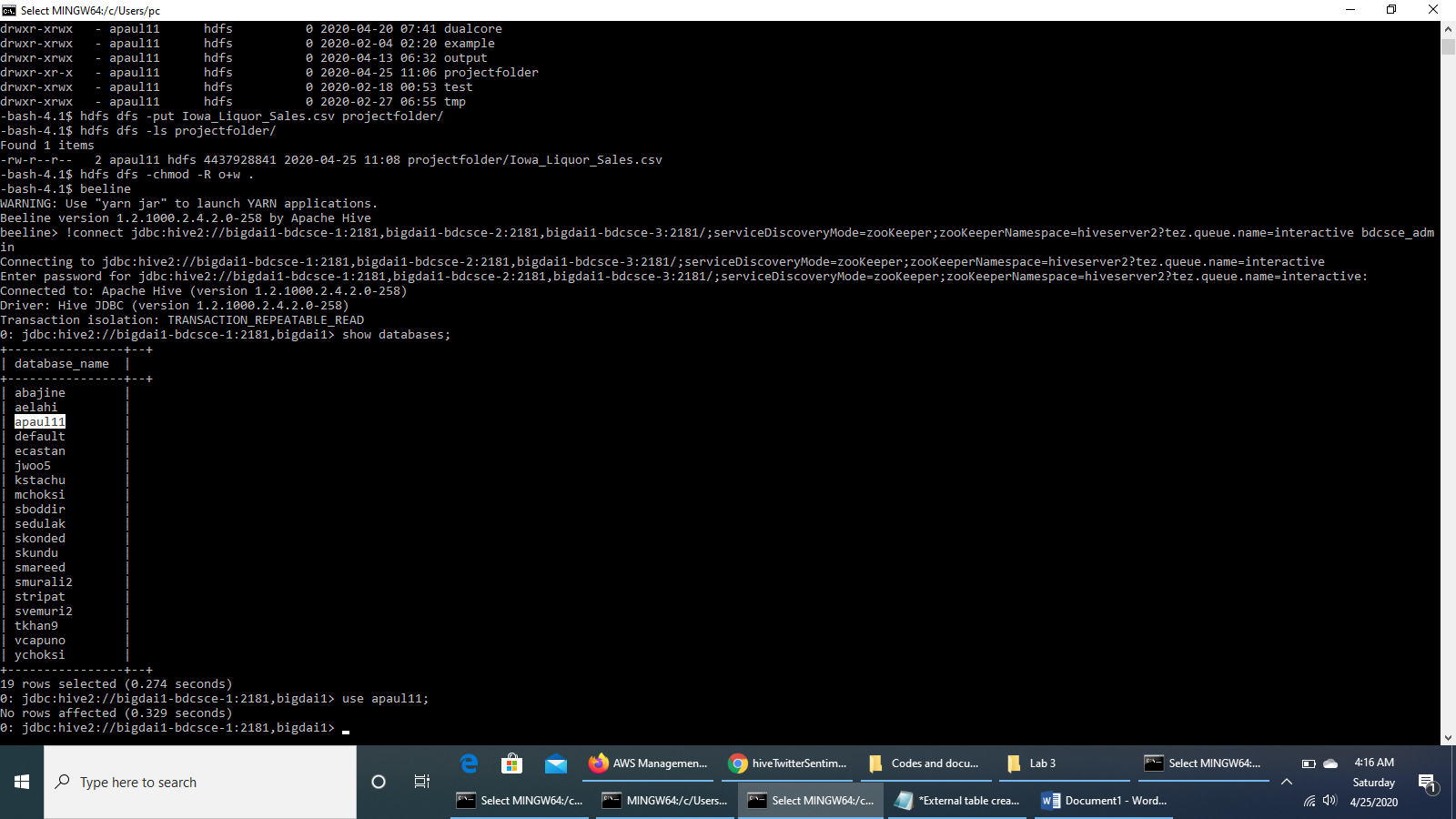
!connect jdbc:hive2://bigdai1-bdcsce-1:2181,bigdai1-bdcsce-2:2181,bigdai1-bdcsce-3:2181/;serviceDiscoveryMode=zooKeeper;zooKeeperNamespace=hiveserver2?tez.queue.name=interactive bdcsce\_admin



**See the list of databases already created and use our database to run further queries**

show databases;

use apaul11; [Alternatively: use tkhan9;]



**Run the HiveQL code to create an external table named *iowaliquorsales***

DROP TABLE IF EXISTS iowaLiquorSales;

CREATE EXTERNAL TABLE IF NOT EXISTS IowaLiquorSales(Invoice\_Item\_Number string, Sale\_Date

date, Store\_Number string, Store\_Name string, Address string,

City string, Zip\_Code string, Store\_Location string, County\_Number

string, County string,

Category string, Category\_Name string, Vendor\_Number string, Vendor\_Name

string, Item\_Number string,

Item\_Description string, Pack int, Bottle\_Volume\_ml int, State\_Bottle\_Cost

double, State\_Bottle\_Retail double, Bottles\_Sold int, Sale\_in\_Dollars double,

Volume\_Sold\_in\_Litres double, Volume\_Sold\_in\_Gallons double )

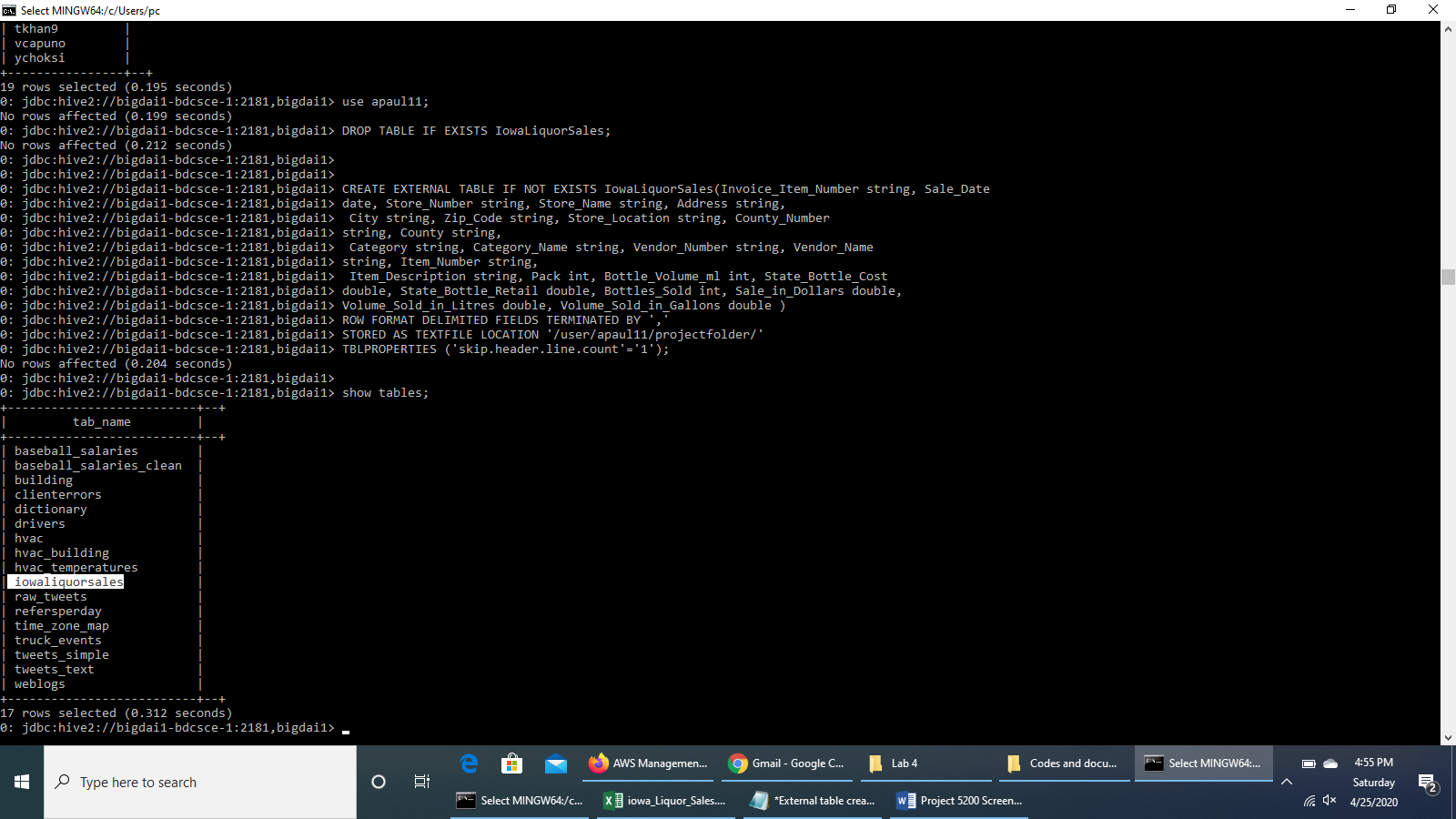
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/tkhan9/projectfolder/'

TBLPROPERTIES ('skip.header.line.count'='1');

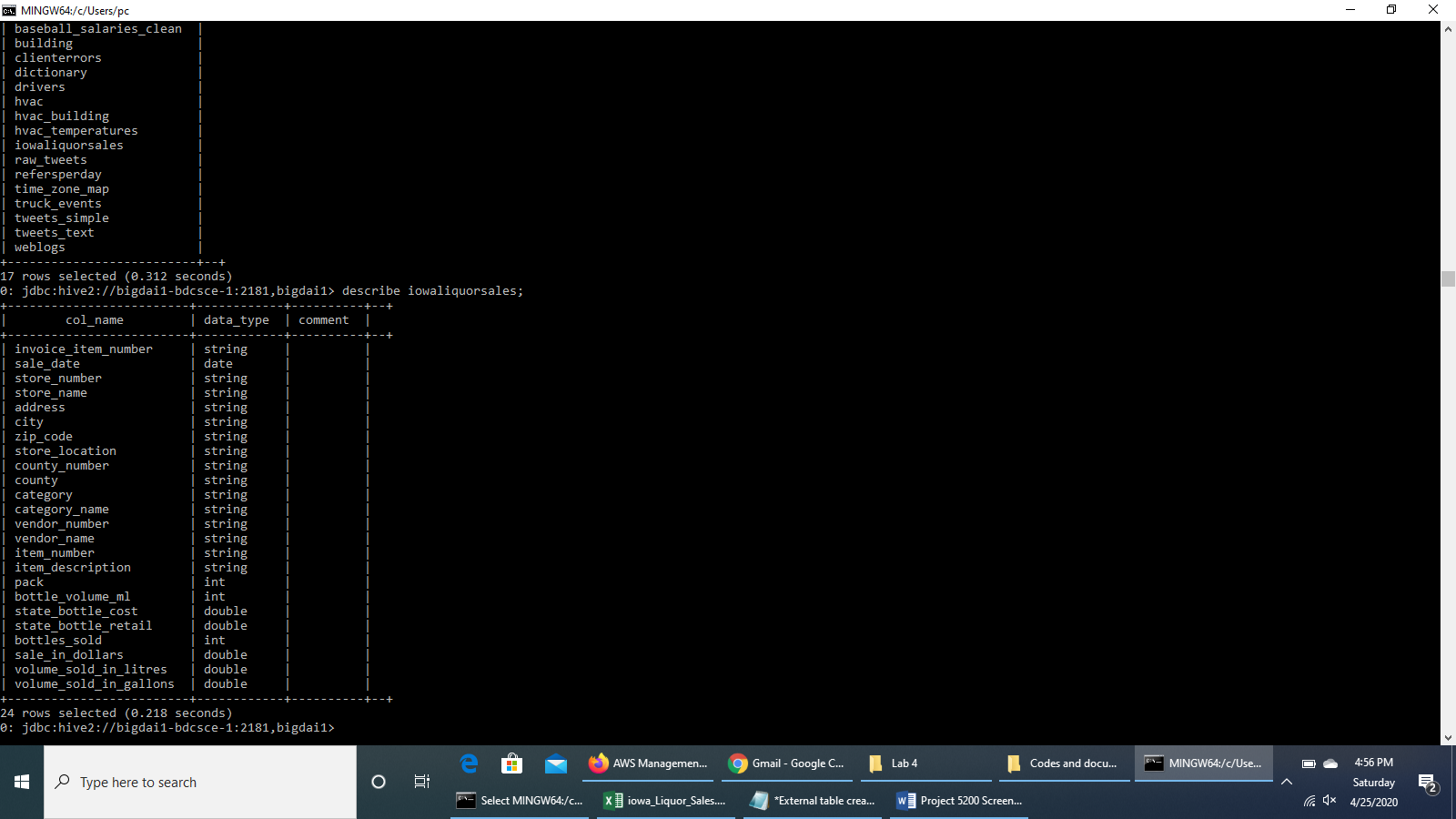
**Check that the table *iowaliquorsales* has been created**

show tables;



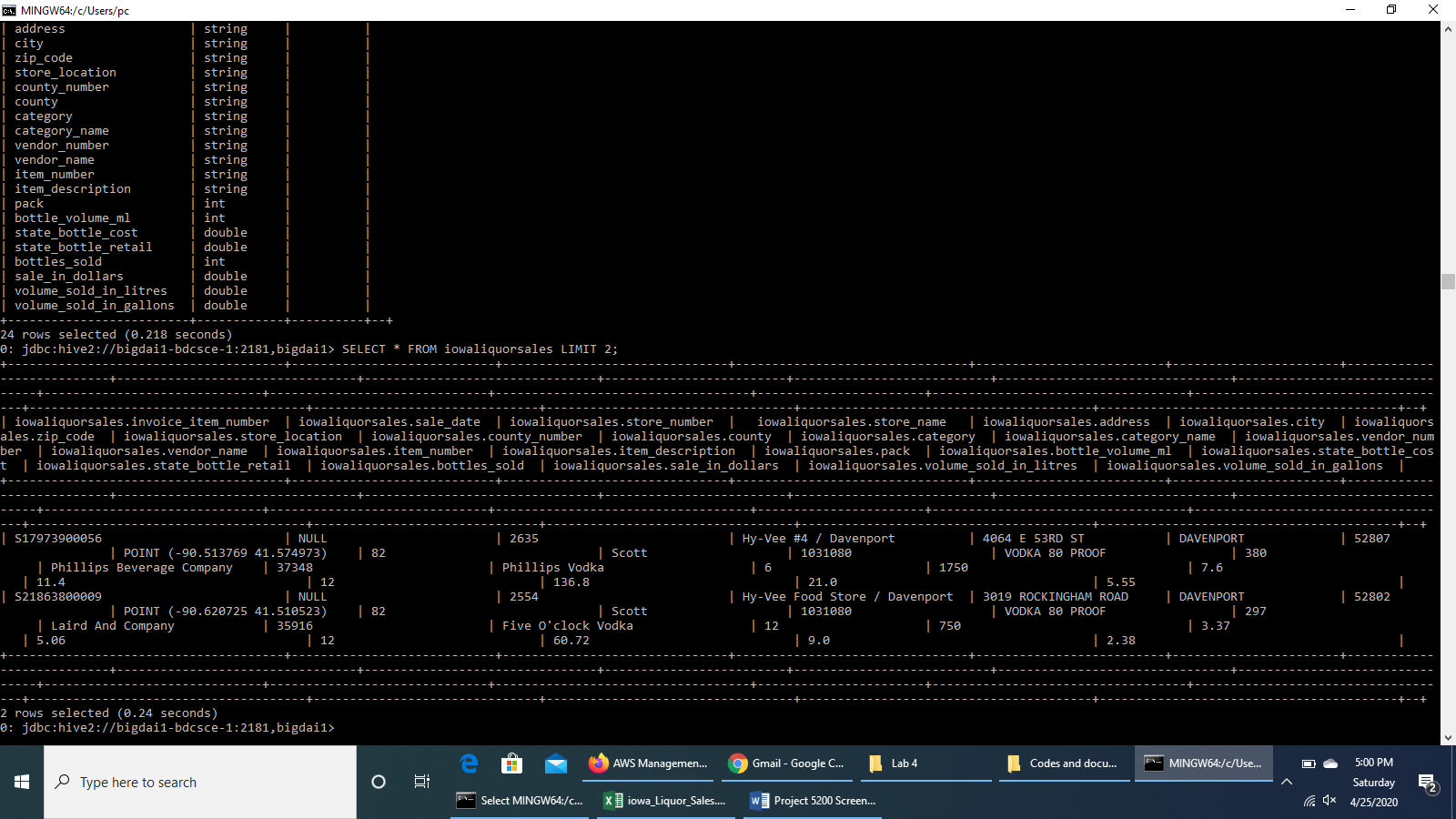
**Check the detailed structure of the table *iowaliquorsales***

describe iowaliquorsales;



**Check if the table *iowaliquorsales* has the value once it is well created**

SELECT \* FROM iowaliquorsales LIMIT 2



**Alter columns & Change Date Format**

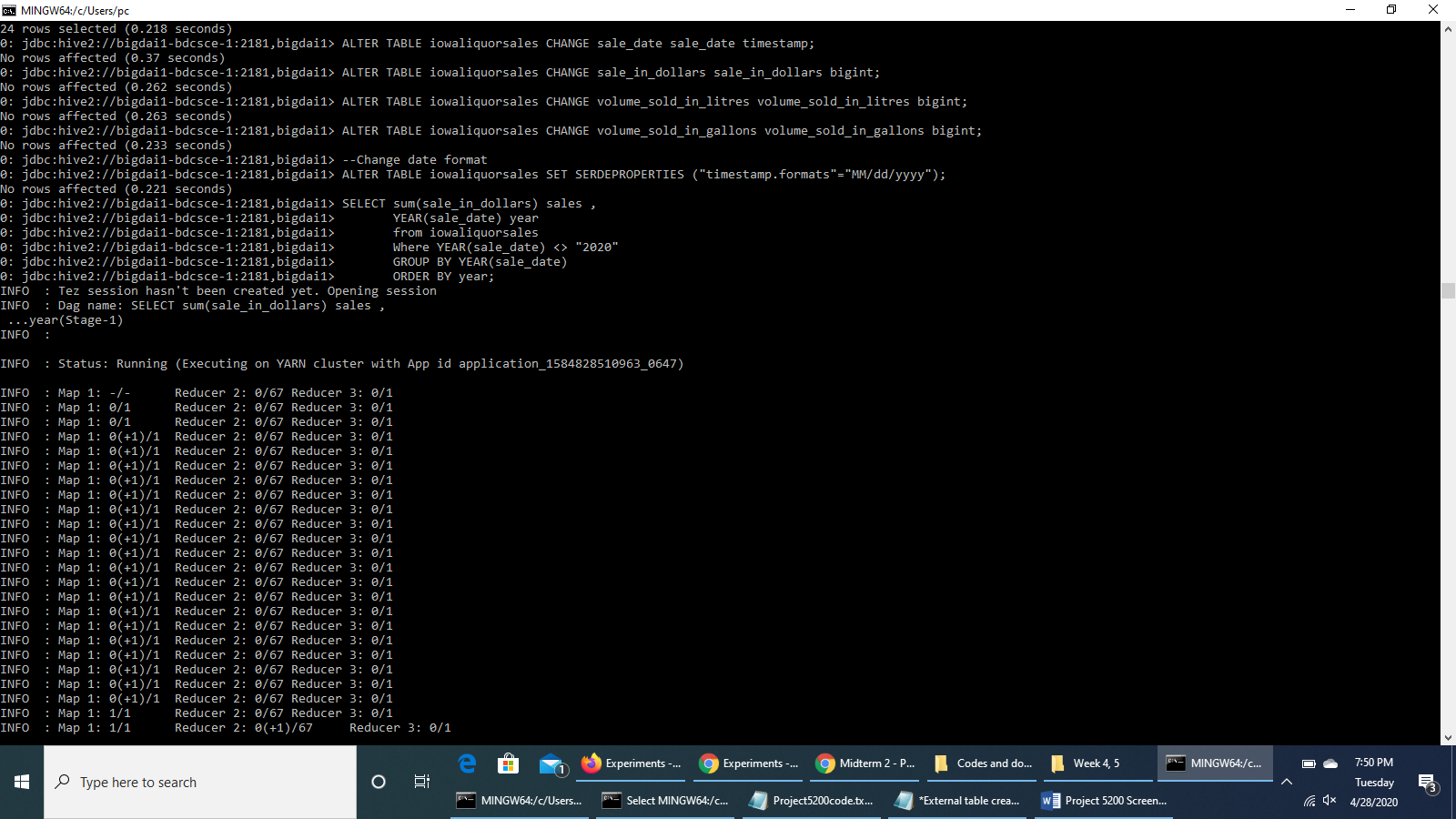
ALTER TABLE iowaliquorsales CHANGE sale\_date sale\_date timestamp;

ALTER TABLE iowaliquorsales CHANGE sale\_in\_dollars sale\_in\_dollars bigint;

ALTER TABLE iowaliquorsales CHANGE volume\_sold\_in\_litres volume\_sold\_in\_litres bigint;

ALTER TABLE iowaliquorsales CHANGE volume\_sold\_in\_gallons volume\_sold\_in\_gallons bigint;

ALTER TABLE iowaliquorsales SET SERDEPROPERTIES ("timestamp.formats"="MM/dd/yyyy");



**Run Hive queries to analyze Sales trend of liquor over the years**

SELECT sum(sale\_in\_dollars) sales ,

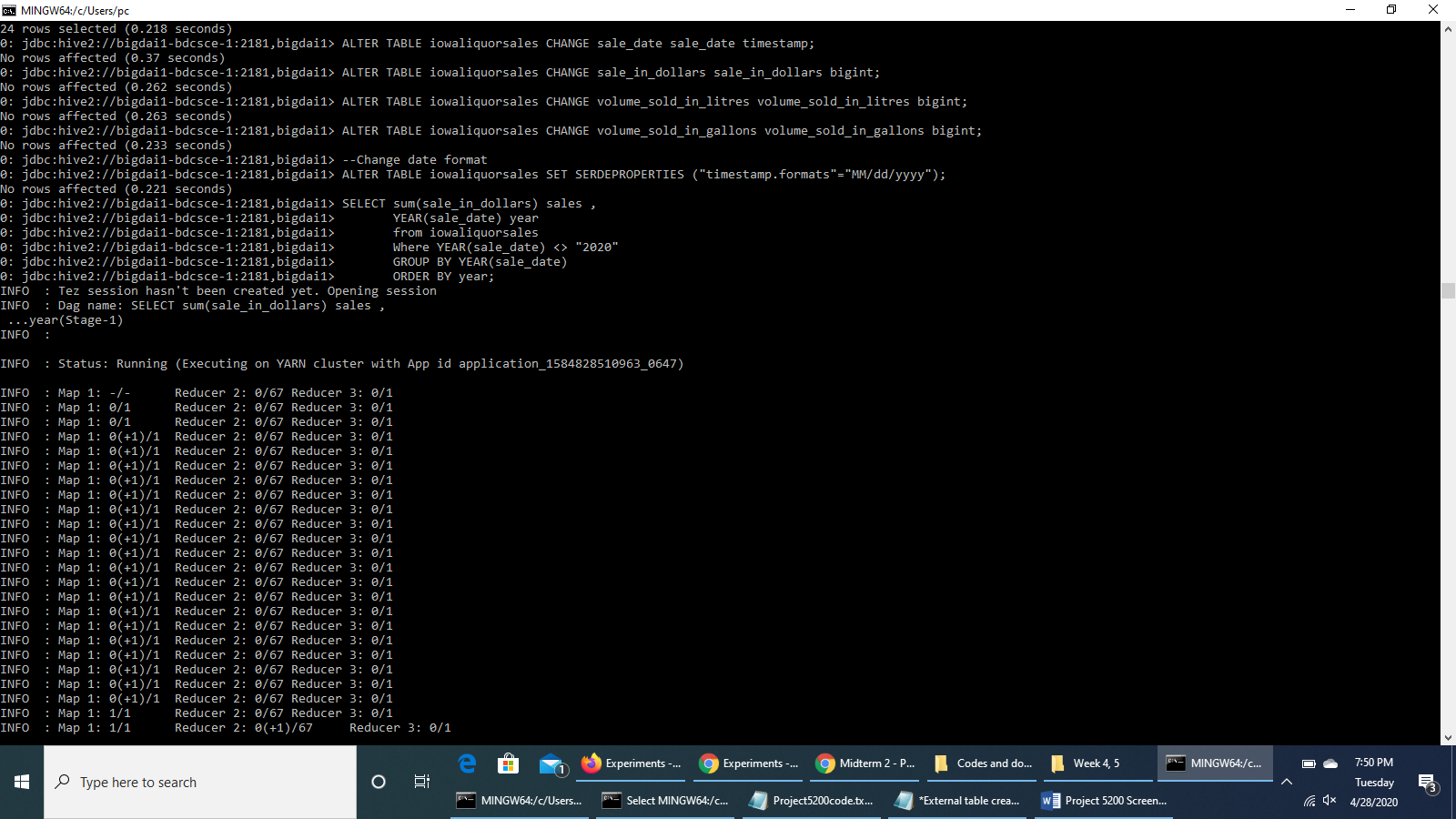
YEAR(sale\_date) year

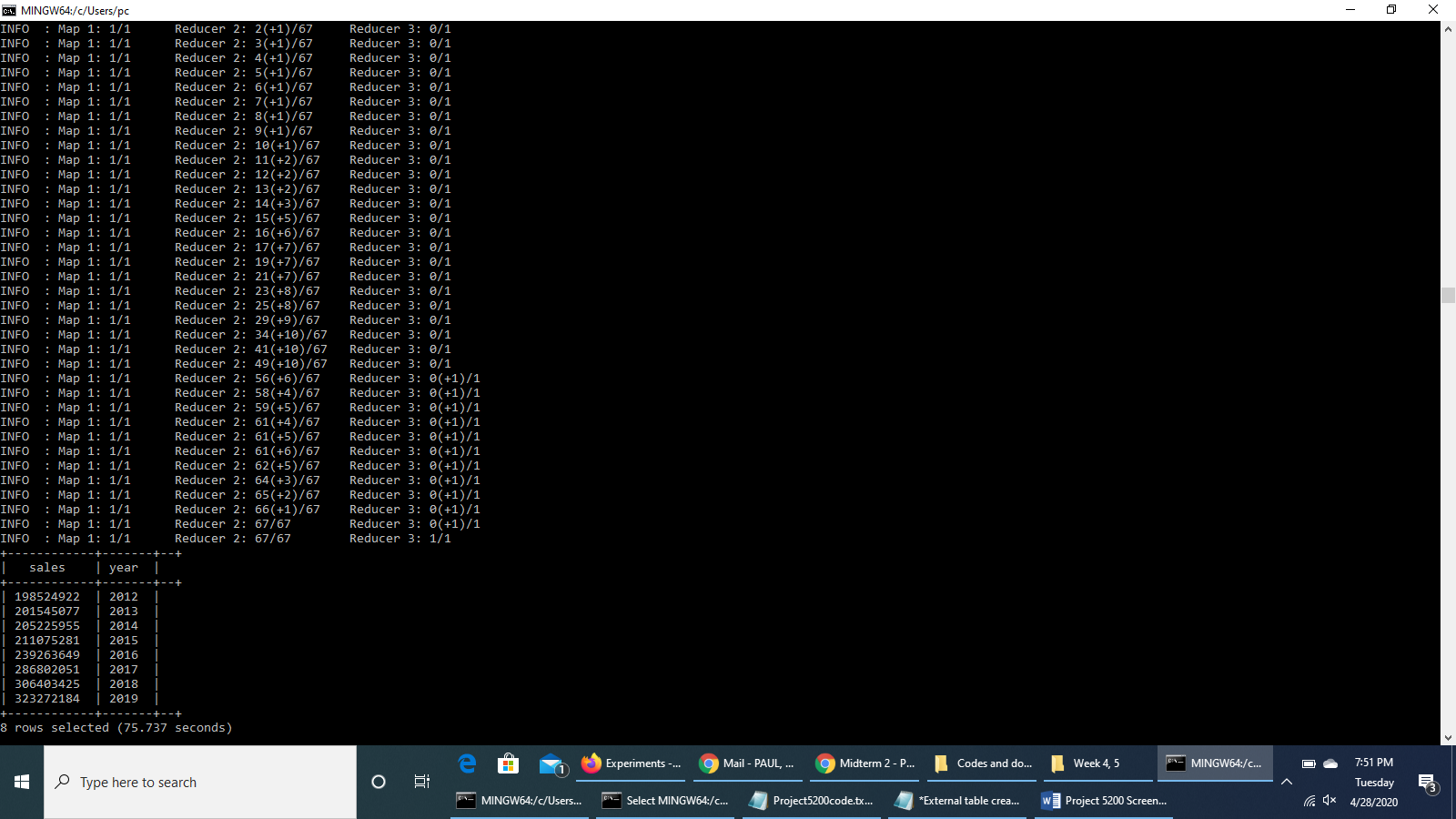
from iowaliquorsales

Where YEAR(sale\_date) <> "2020"

GROUP BY YEAR(sale\_date)

ORDER BY year;





**Run Hive queries to find out the Top 10 counties in terms of sale amount in Dollars**

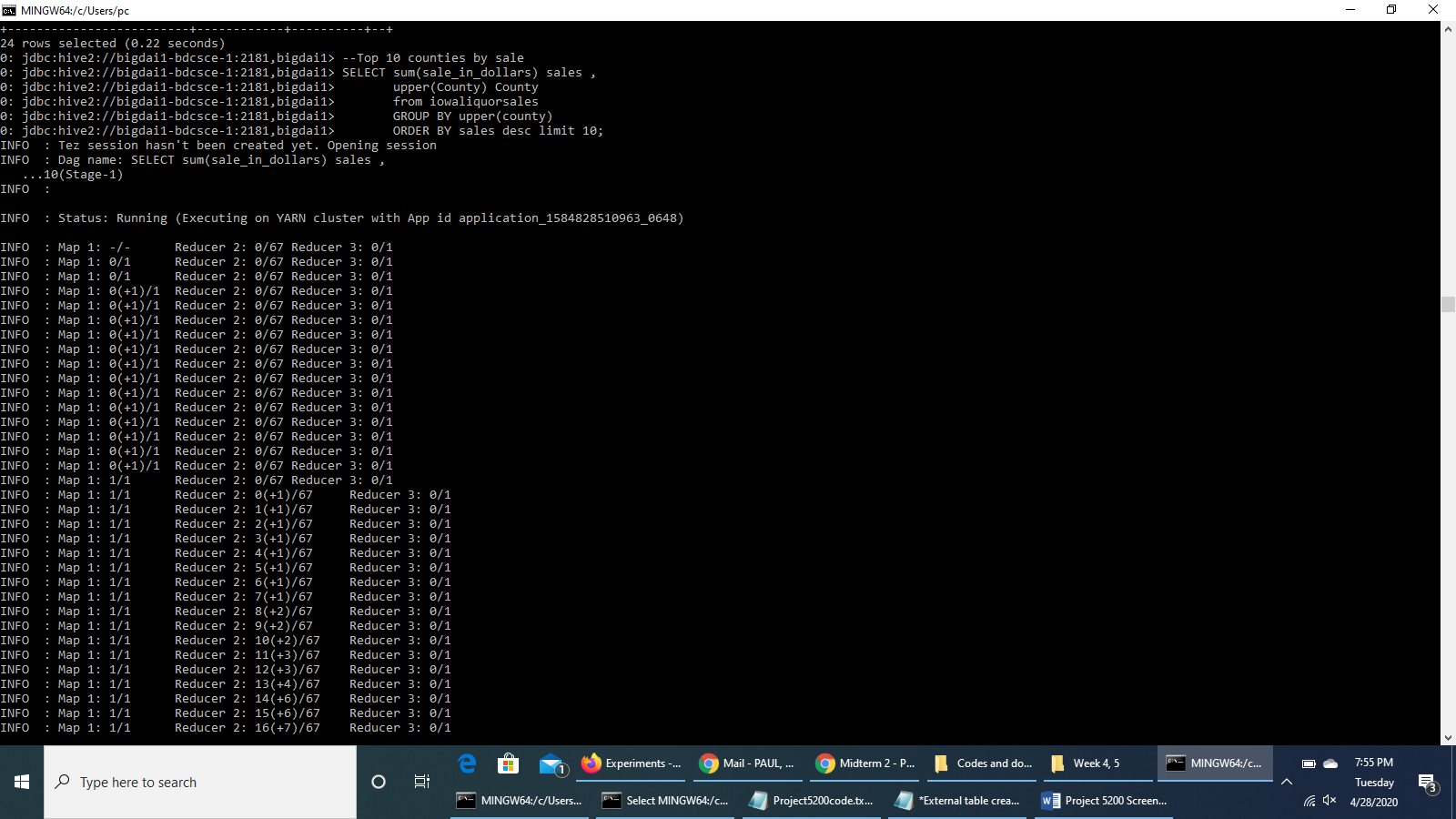
SELECT sum(sale\_in\_dollars) sales ,

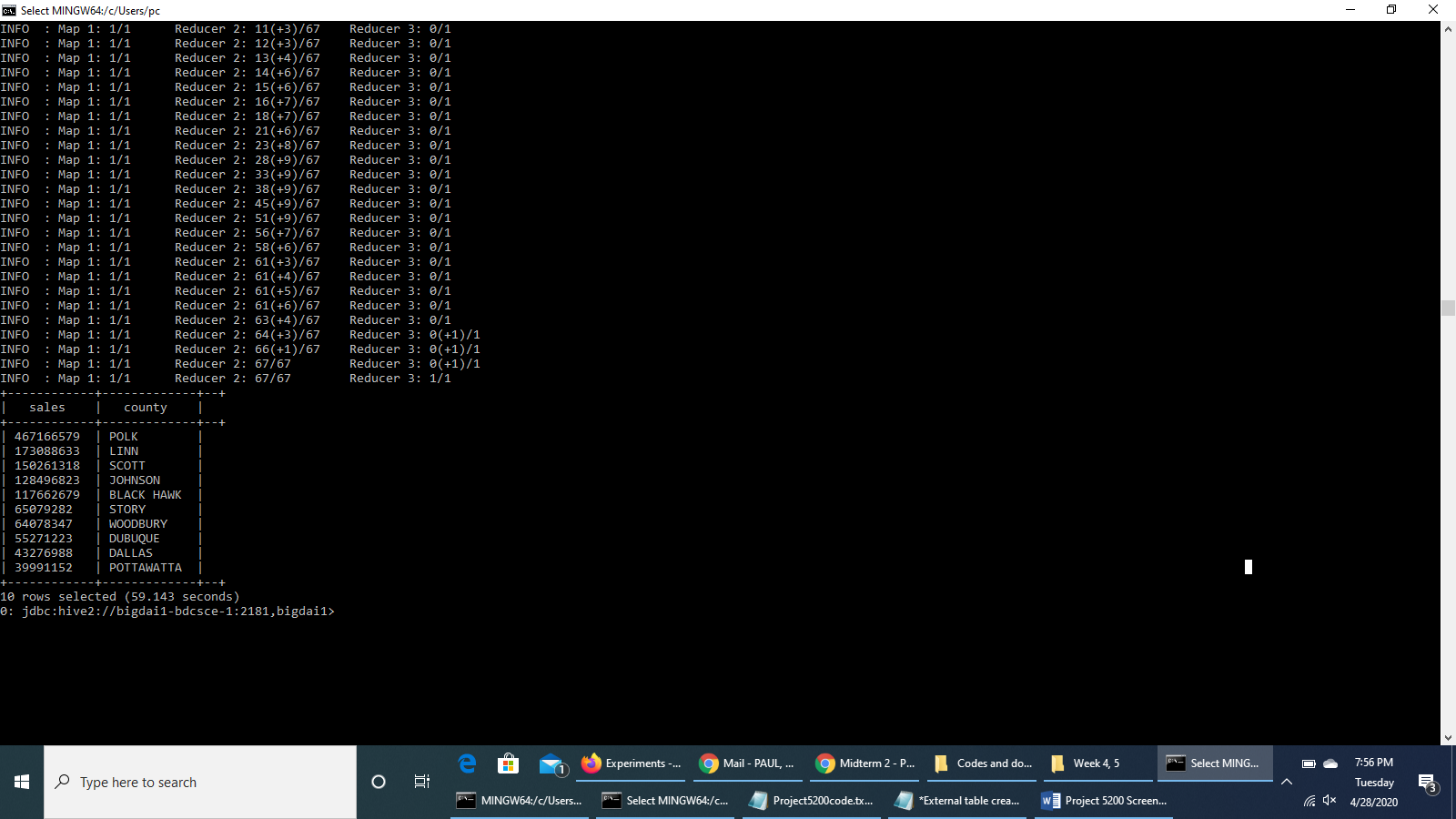
upper(County) County

from iowaliquorsales

GROUP BY upper(county)

ORDER BY sales DESC limit 10;





**Run Hive queries to find out the Top 5 categories of liquors sold with highest volume in Litres**

SELECT SUM(Volume\_Sold\_in\_Litres) volume,

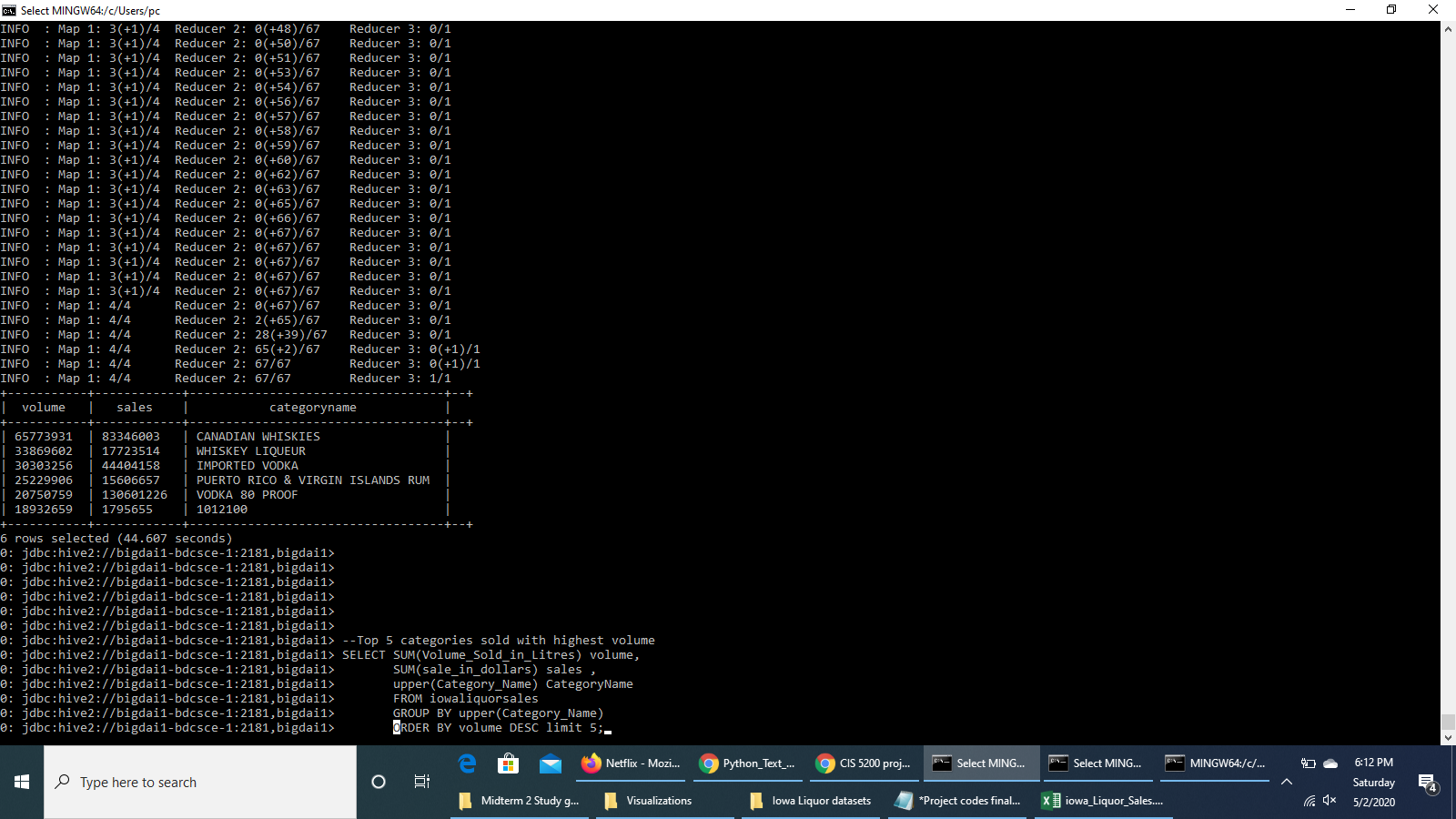
SUM(sale\_in\_dollars) sales ,

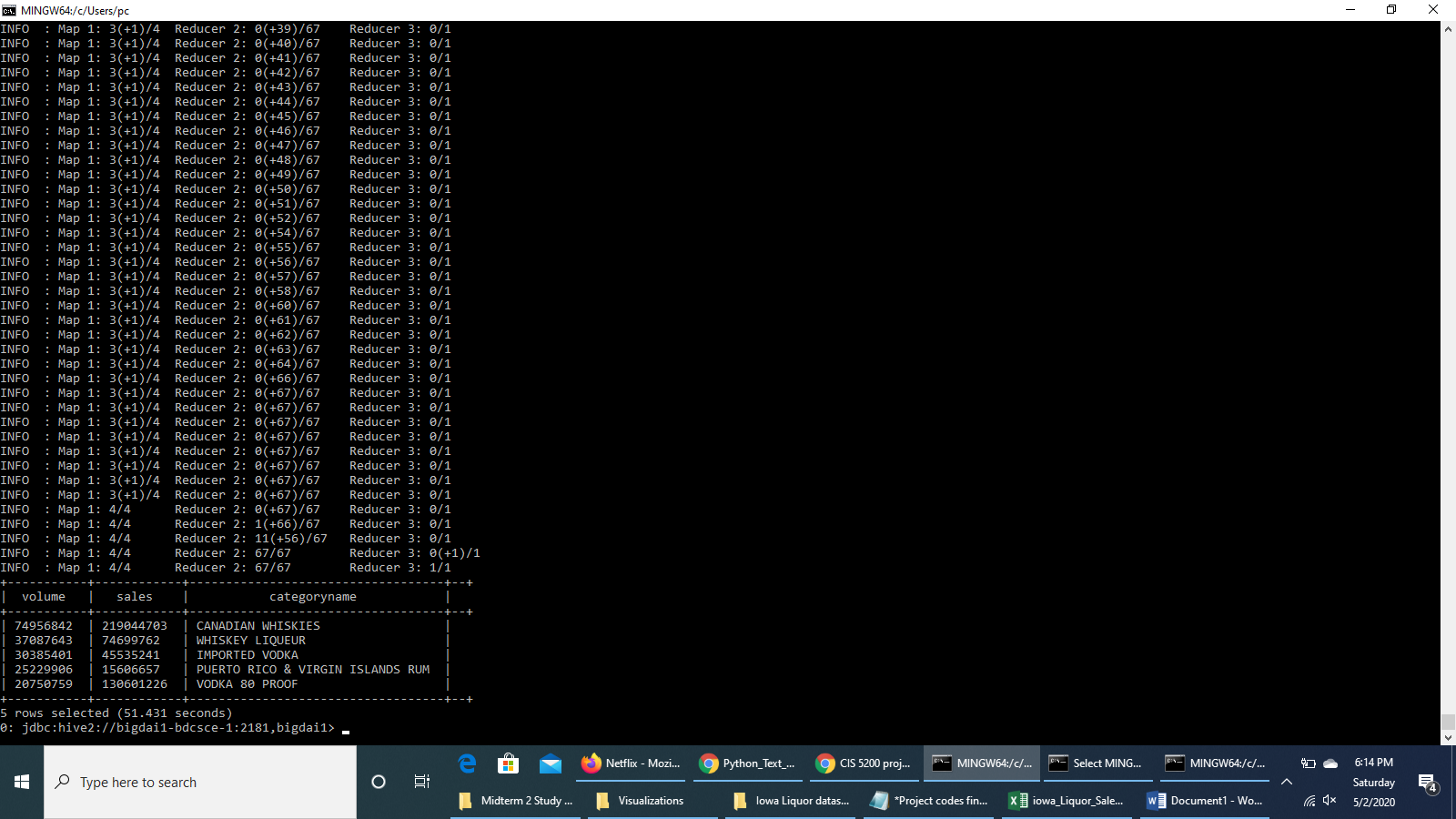
upper(Category\_Name) CategoryName

FROM iowaliquorsales

GROUP BY upper(Category\_Name)

ORDER BY volume DESC limit 5;





**Run Hive queries to find out the Top 10 cities with highest sales in Dollars**

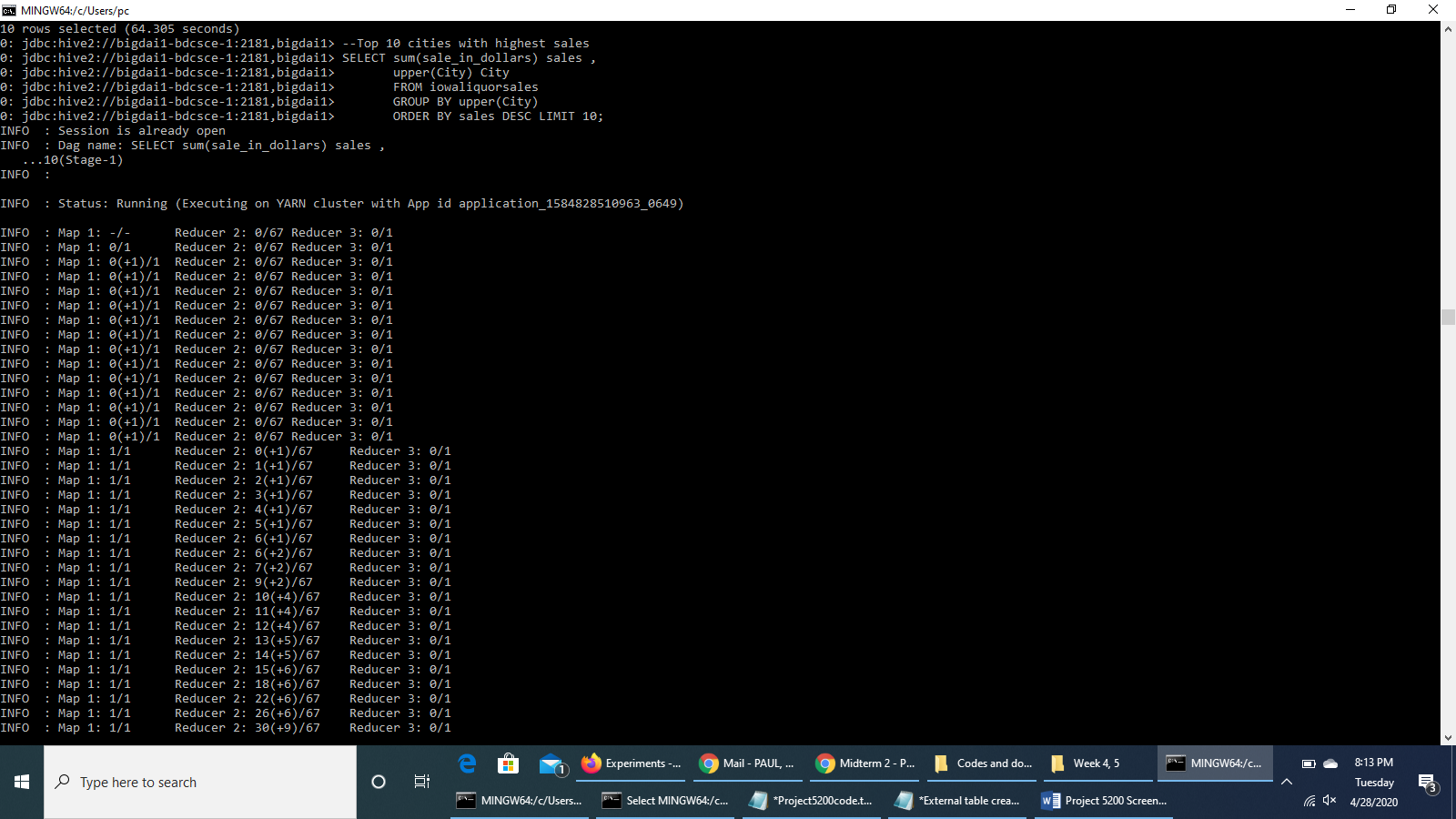
SELECT sum(sale\_in\_dollars) sales ,

upper(City) City

FROM iowaliquorsales

GROUP BY upper(City)

ORDER BY sales DESC LIMIT 10;



**Run Hive queries to find out Sale of Canadian Whiskies across counties for the year 2019**

SELECT Invoice\_Item\_Number,

sale\_in\_dollars sales,

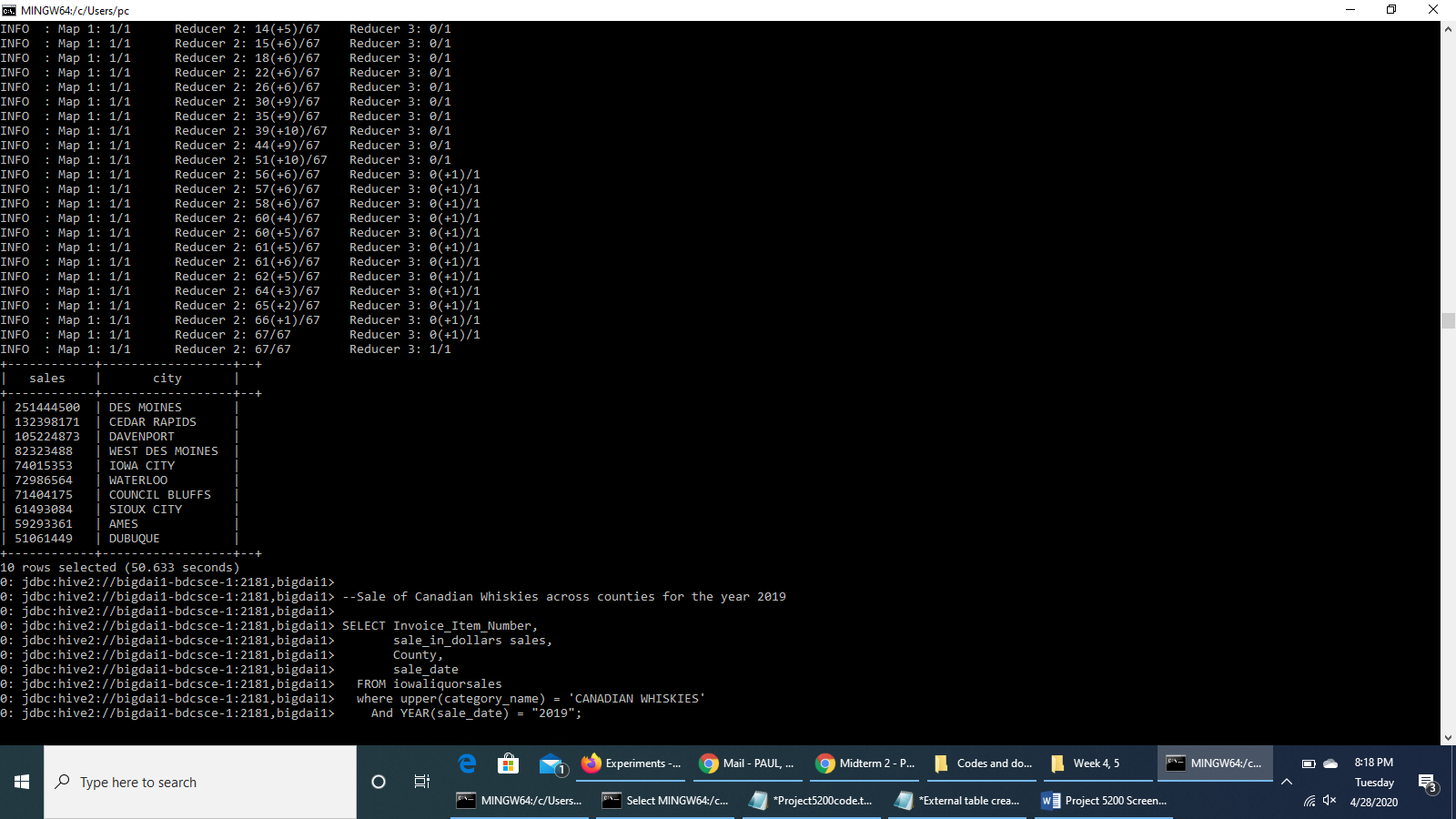
County,

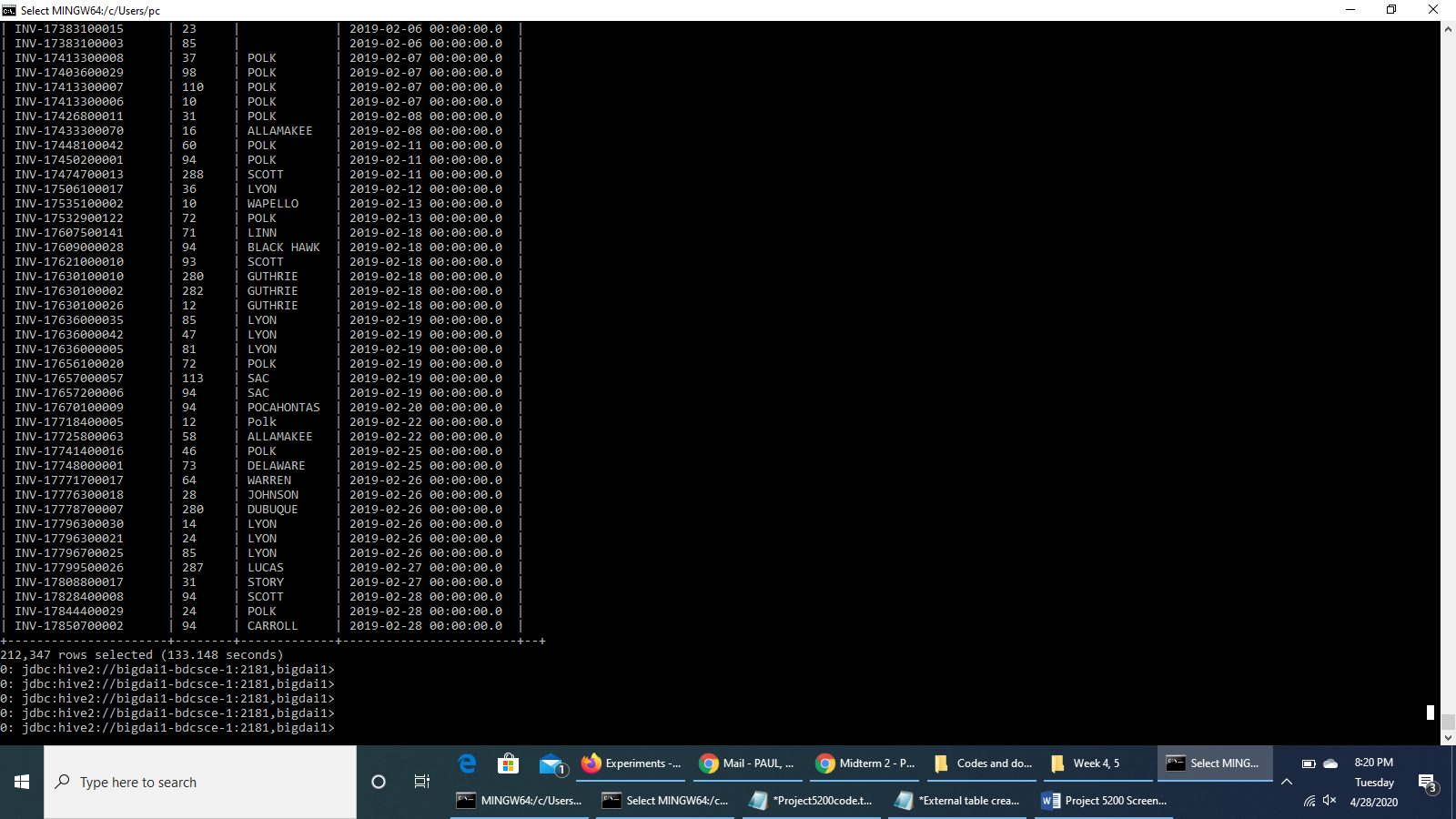
sale\_date

FROM iowaliquorsales

where upper(category\_name) = 'CANADIAN WHISKIES'

And YEAR(sale\_date) = "2019";





**We run HiveQL code to create a table *SaleCanadianWhiskies* to store the data of the sale of Canadian Whiskies across counties for the year 2019 for extraction**

DROP TABLE IF EXISTS SaleCanadianWhiskies;

CREATE TABLE IF NOT EXISTS SaleCanadianWhiskies

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/tkhan9/projectfolder/SaleCanadianWhiskies/'

AS

SELECT Invoice\_Item\_Number,

sale\_in\_dollars sales,

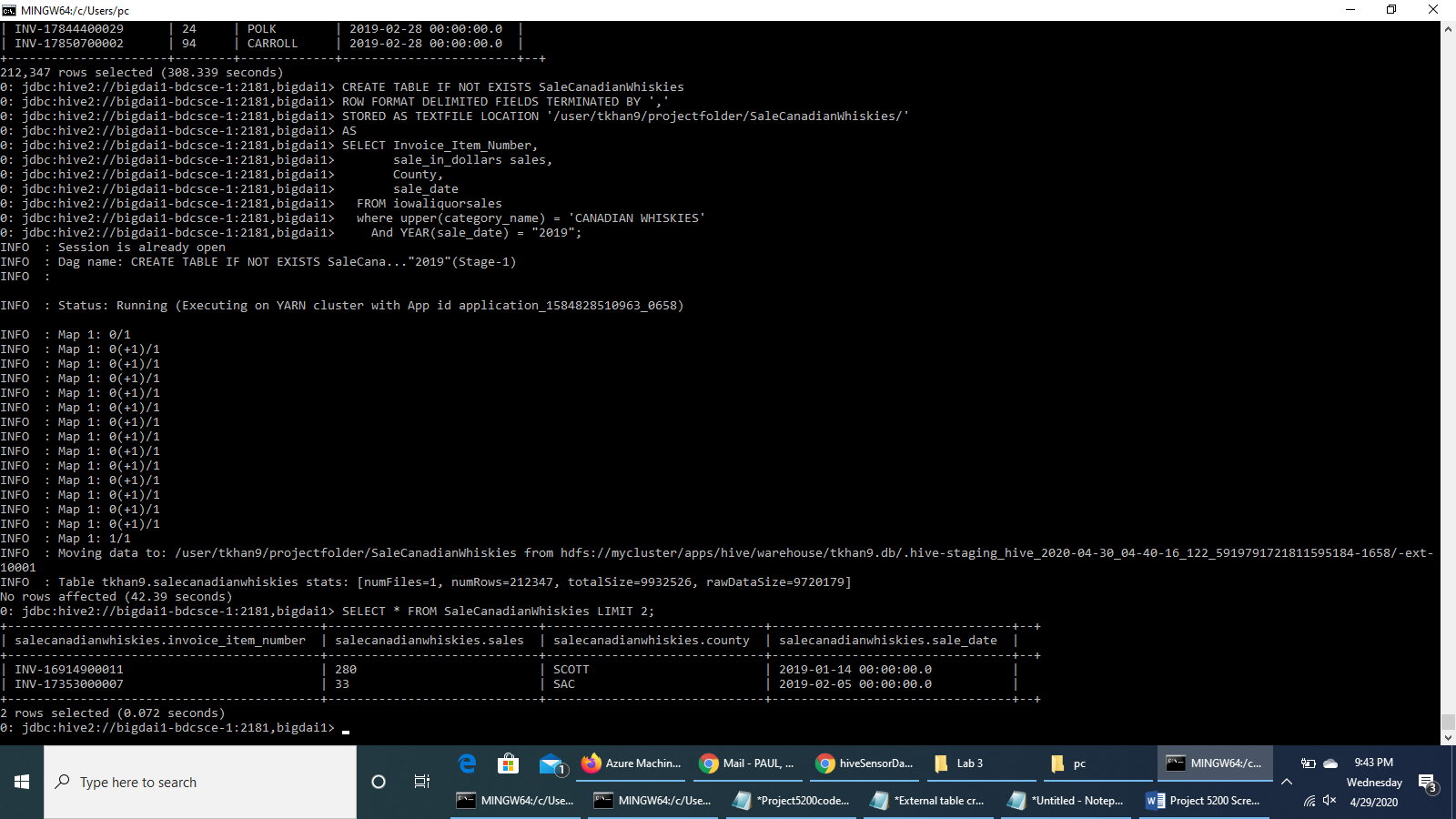
County,

sale\_date

FROM iowaliquorsales

where upper(category\_name) = 'CANADIAN WHISKIES'

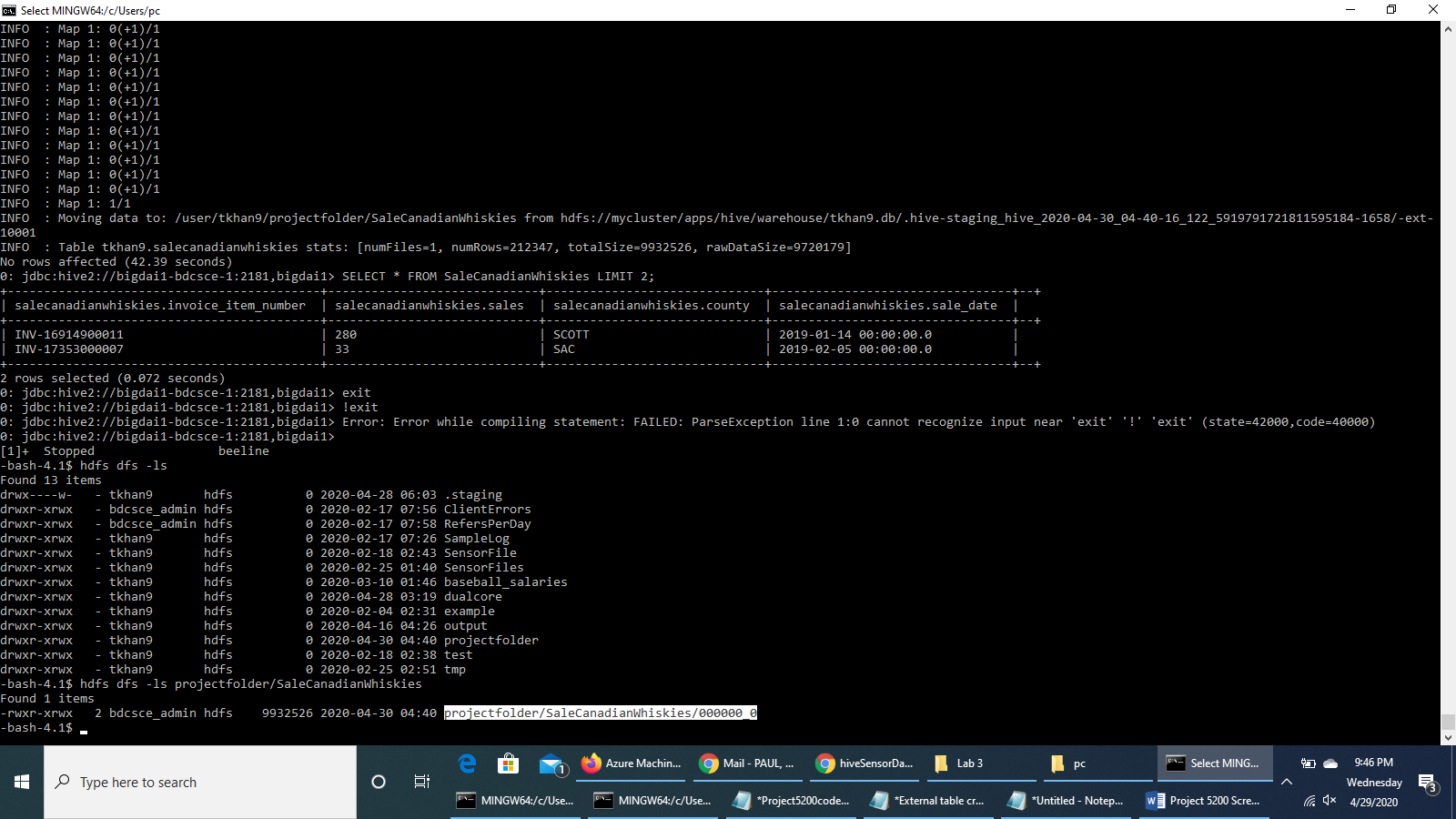
And YEAR(sale\_date) = "2019";



**Press CTRL+Z to exit Beeline**

**Check HDFS path of the output file 000000\_0**

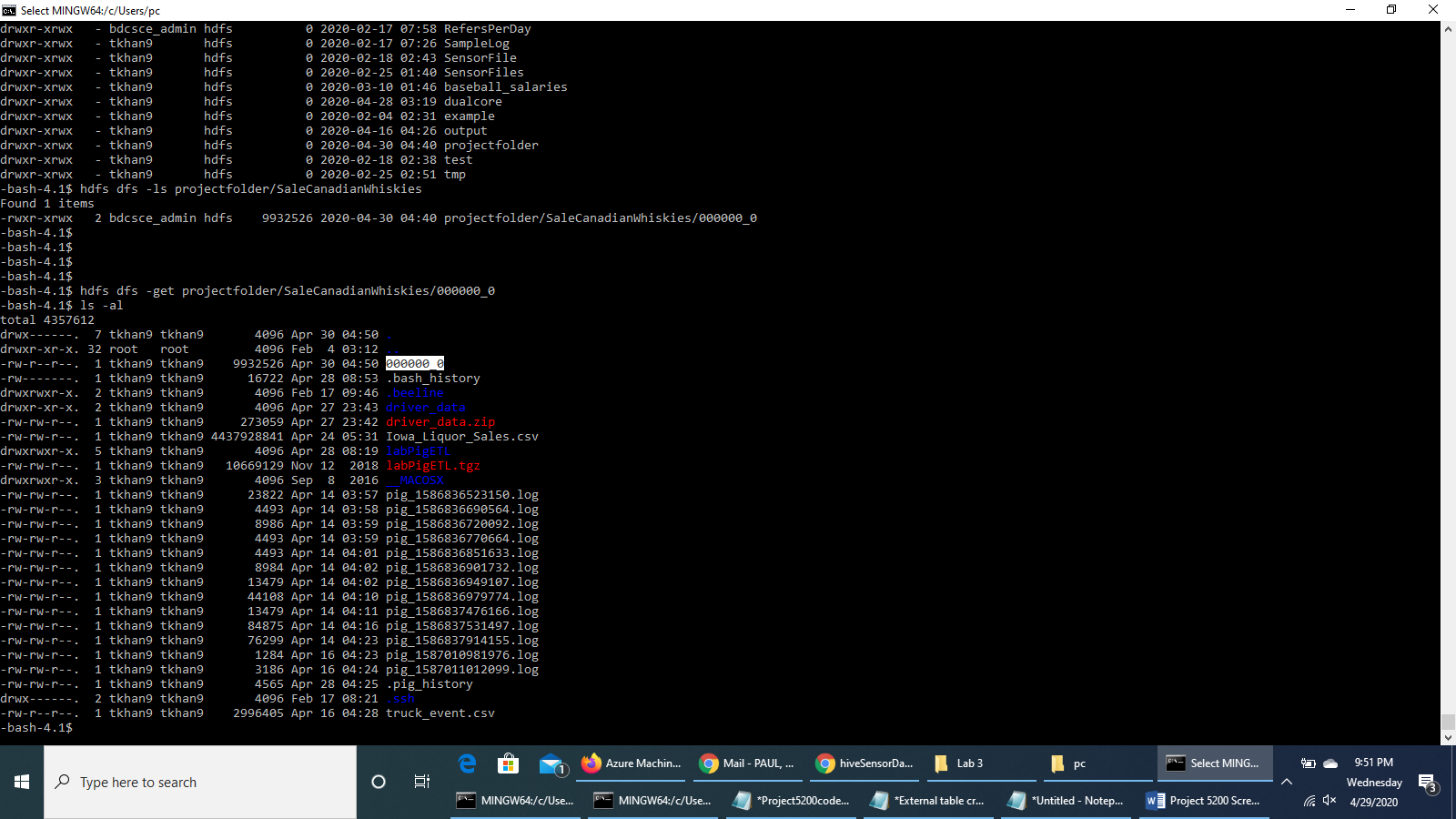
hdfs dfs -ls projectfolder/SaleCanadianWhiskies



**To download the output file to the local file systems**

hdfs dfs -get projectfolder/SaleCanadianWhiskies/000000\_0

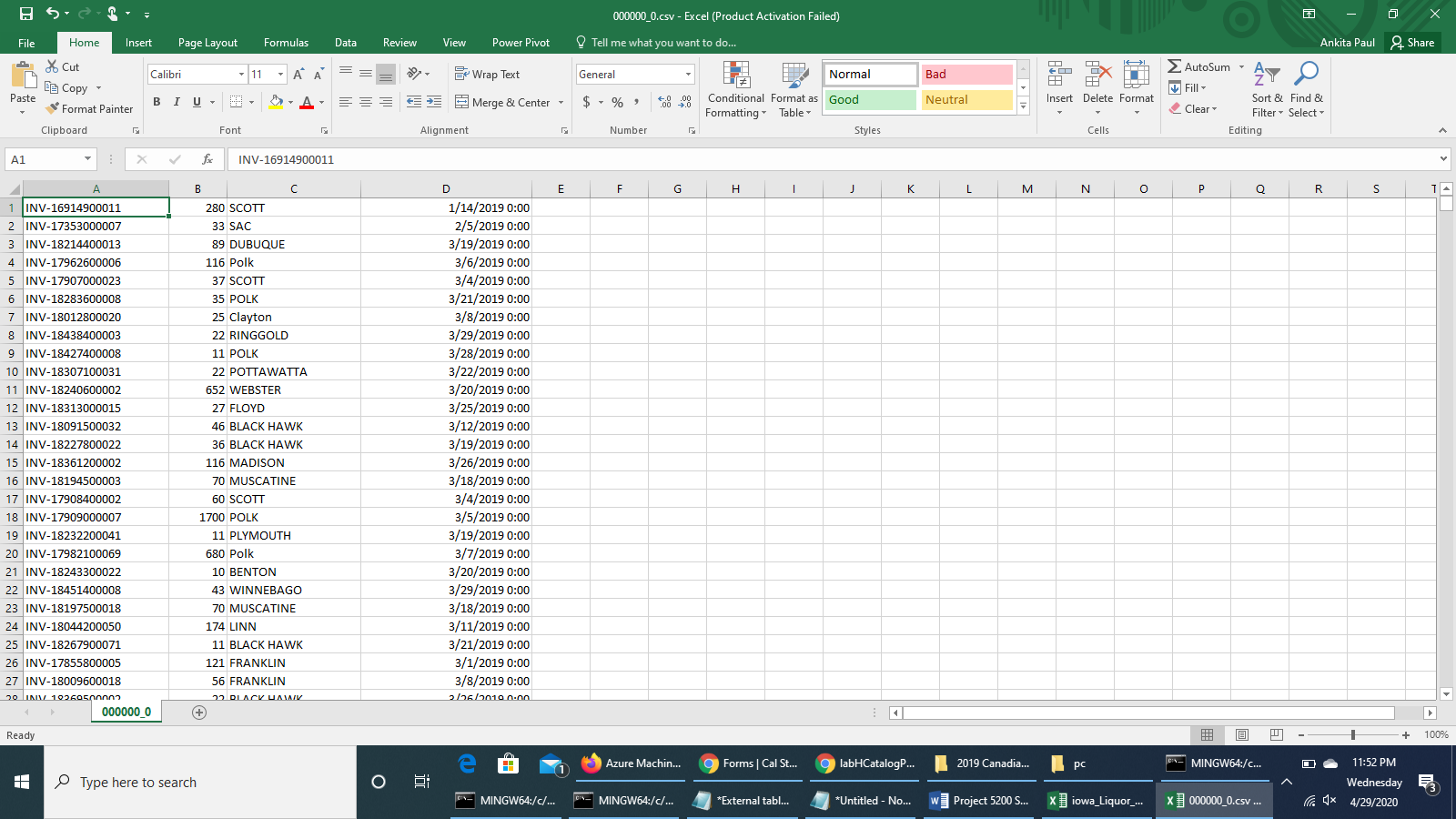
ls -al



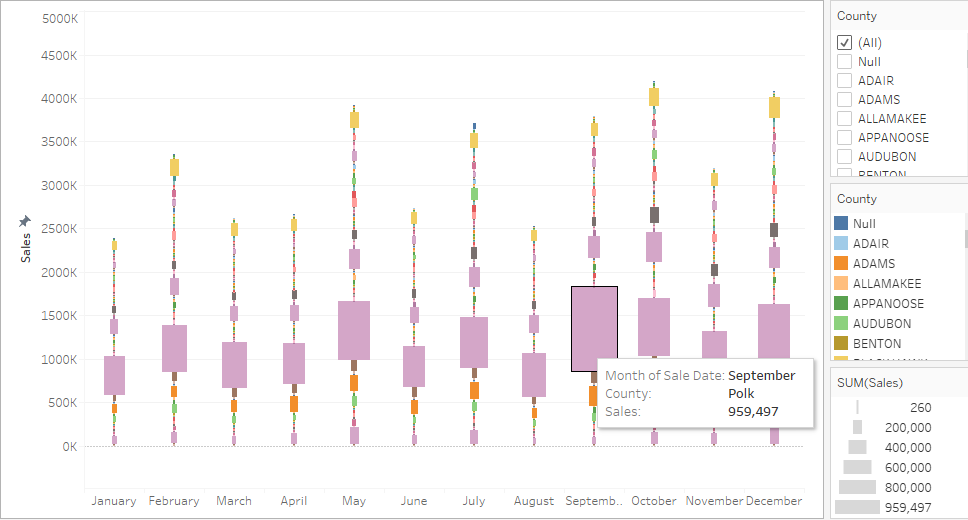
**Open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your computer (or your PC/Laptop).**

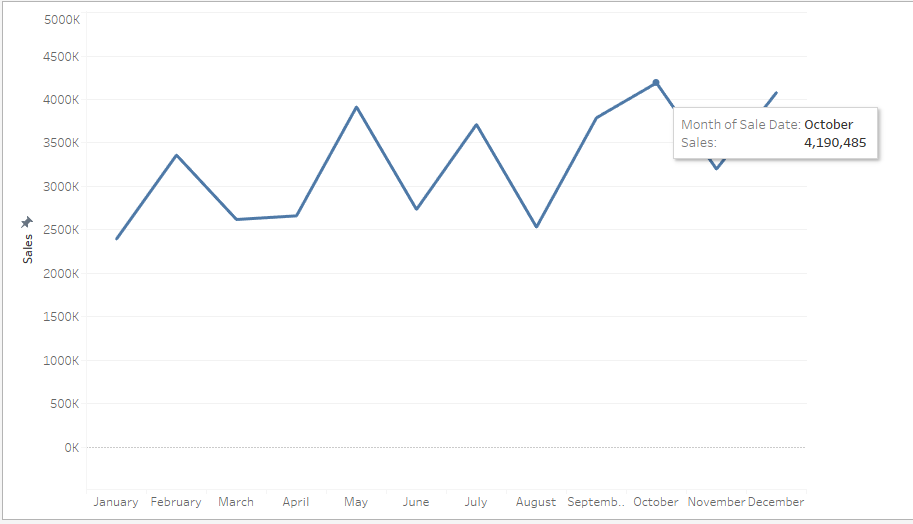
scp tkhan9@129.150.71.75:/home/tkhan9/000000\_0 000000\_0.csv

**Screenshot of the extracted csv file:**



**Screenshot of the Temporal analysis visualization of the Sale of Canadian Whiskies in 2019 on *MS Excel Pivot Chart***





**Tempo-spatial analysis - Sale of Canadian Whiskies across counties**

We used a cleaned dataset “Cleaned\_Location\_data.csv” by splitting the “Store Location” column from the orginal dataset into two separate columns of “Longitude” and “Latitude”.

**Before proceeding with the next part of analysis, we will remove the previous output data file.**

rm 000000\_0

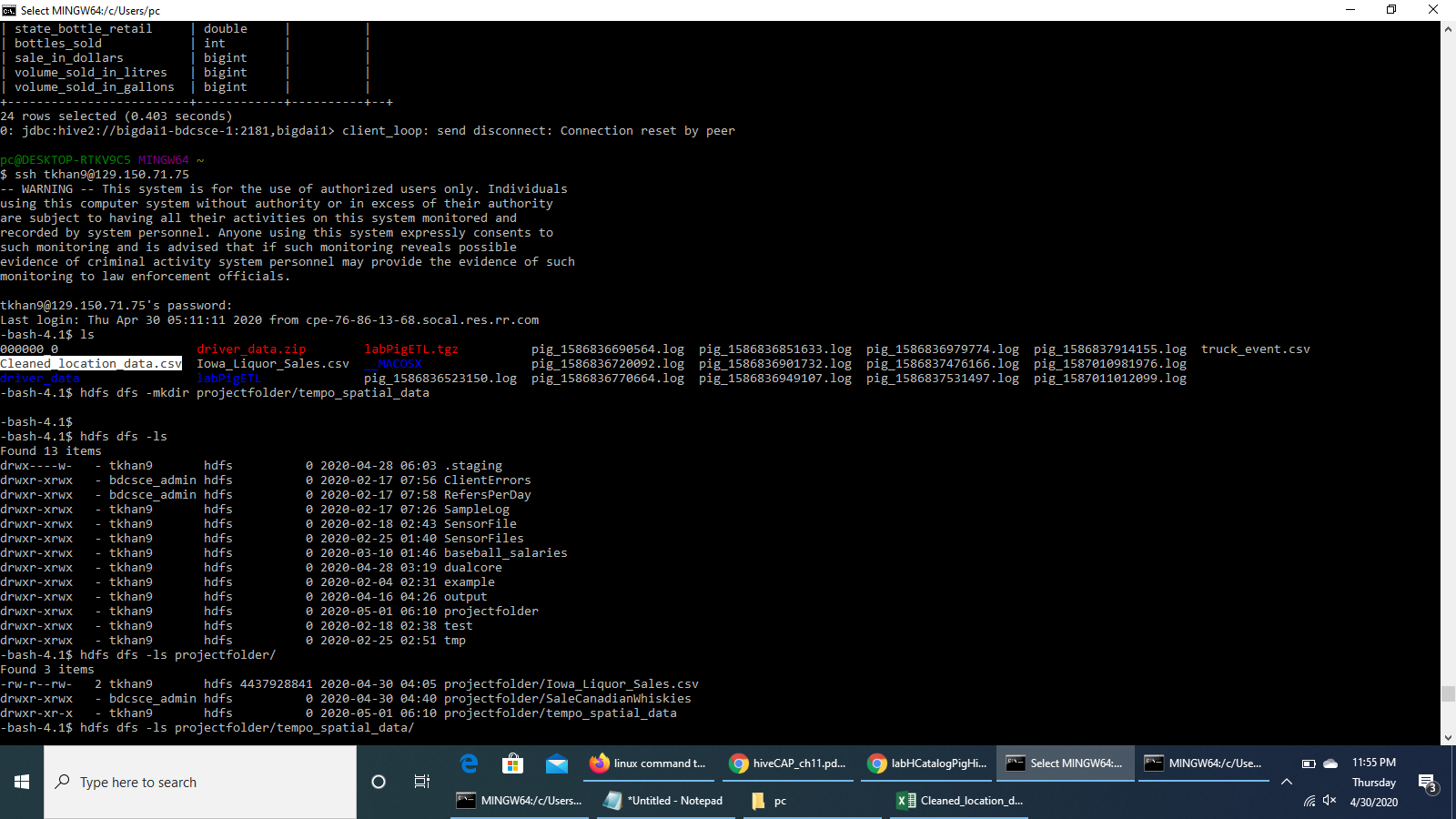
**Download the new data input file from Amazon S3**

wget –O Cleaned\_location\_data.csv <https://projectcis5200.s3-us-west-1.amazonaws.com/Cleaned_location_data.csv>

**Put the CSV input file in *tempo\_spatial\_data* directory in HDFS**

1. hdfs dfs -mkdir projectfolder/tempo\_spatial\_data

2. hdfs dfs -put Cleaned\_location\_data.csv projectfolder/tempo\_spatial\_data/



**Run the earlier commands to connect to Beeline and use the database tkhan9 by running the Hive command in Beeline CLI.**

**Create an external table *sale\_location\_data***

DROP TABLE IF EXISTS sale\_location\_data;

CREATE EXTERNAL TABLE IF NOT EXISTS sale\_location\_data(Invoice\_or\_Item\_Number string, Date\_of\_sale

date, StoreNumber string, StoreName string,

City string, Zip\_Code string, Longitude double, Latitude double, County string,

Category\_Name string, Sale\_Amount double,

Volume\_Sold\_litres double)

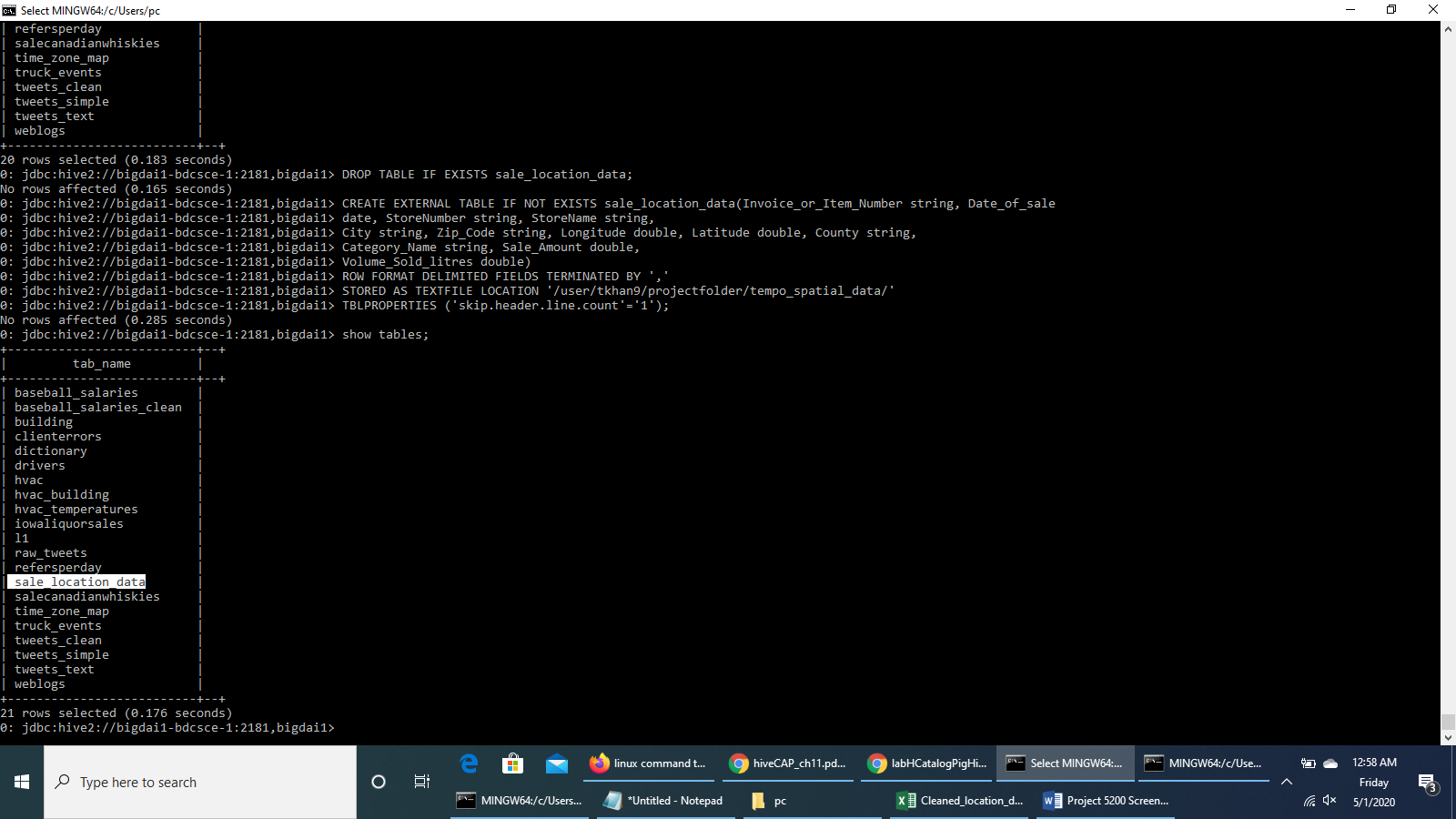
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/tkhan9/projectfolder/tempo\_spatial\_data/'

TBLPROPERTIES ('skip.header.line.count'='1');

**Check that the table *sale\_location\_data* has been created**

show tables;



**Alter columns & Change Date Format**

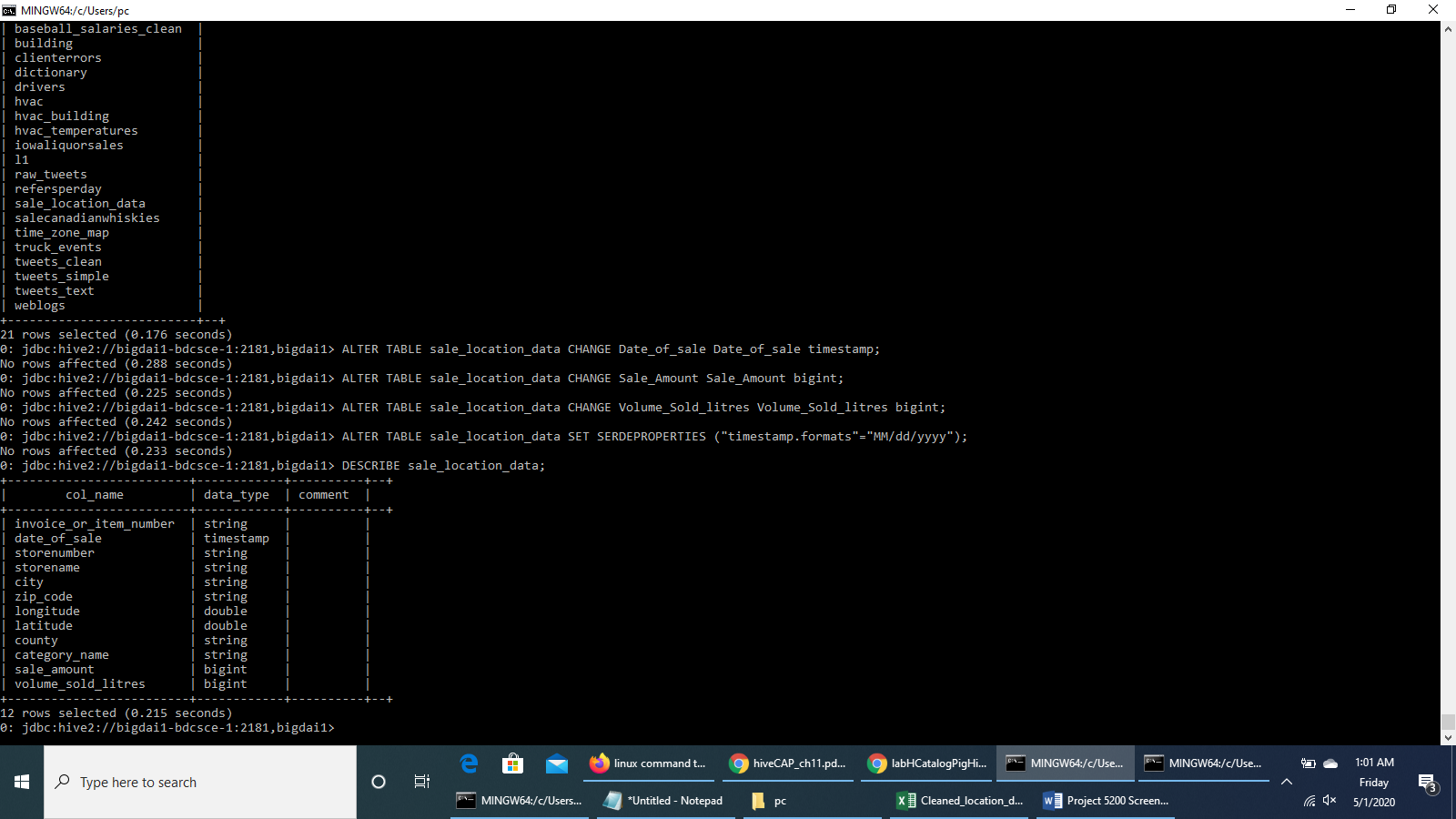
ALTER TABLE sale\_location\_data CHANGE Date\_of\_sale Date\_of\_sale timestamp;

ALTER TABLE sale\_location\_data CHANGE Sale\_Amount Sale\_Amount bigint;

ALTER TABLE sale\_location\_data CHANGE Volume\_Sold\_litres Volume\_Sold\_litres bigint;

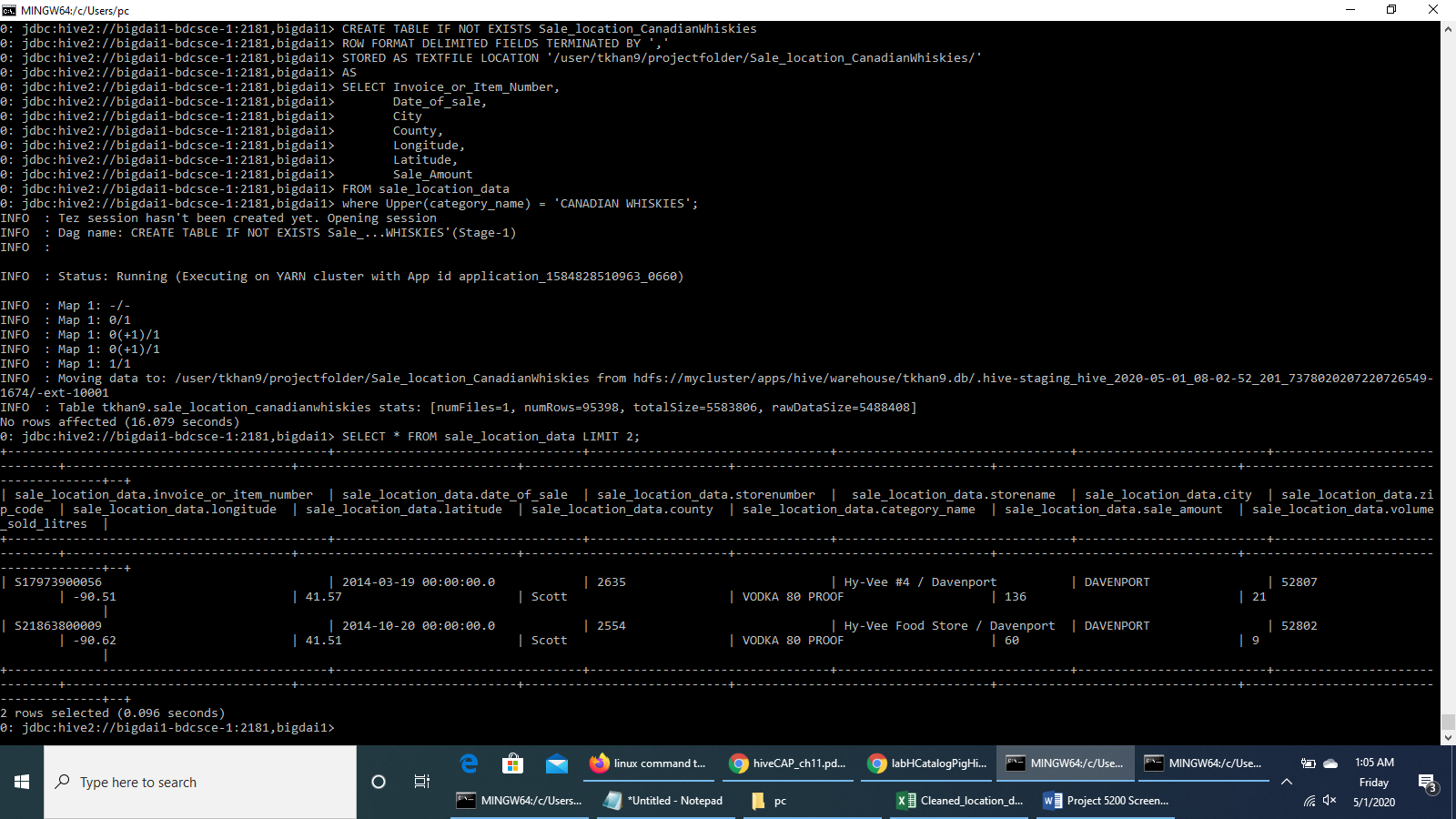
ALTER TABLE sale\_location\_data SET SERDEPROPERTIES ("timestamp.formats"="MM/dd/yyyy");

DESCRIBE sale\_location\_data;



**Check if the table *sale\_location\_data* has the value once it is well created**

SELECT \* FROM sale\_location\_data LIMIT 2;



**We run HiveQL code to create a table *Sale\_location\_CanadianWhiskies* to store the data of the sale of Canadian Whiskies across counties over the years.**

DROP TABLE IF EXISTS Sale\_location\_CanadianWhiskies;

CREATE TABLE IF NOT EXISTS Sale\_location\_CanadianWhiskies

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

STORED AS TEXTFILE LOCATION '/user/tkhan9/projectfolder/Sale\_location\_CanadianWhiskies/'

AS

SELECT Invoice\_or\_Item\_Number,

Date\_of\_sale,

County,

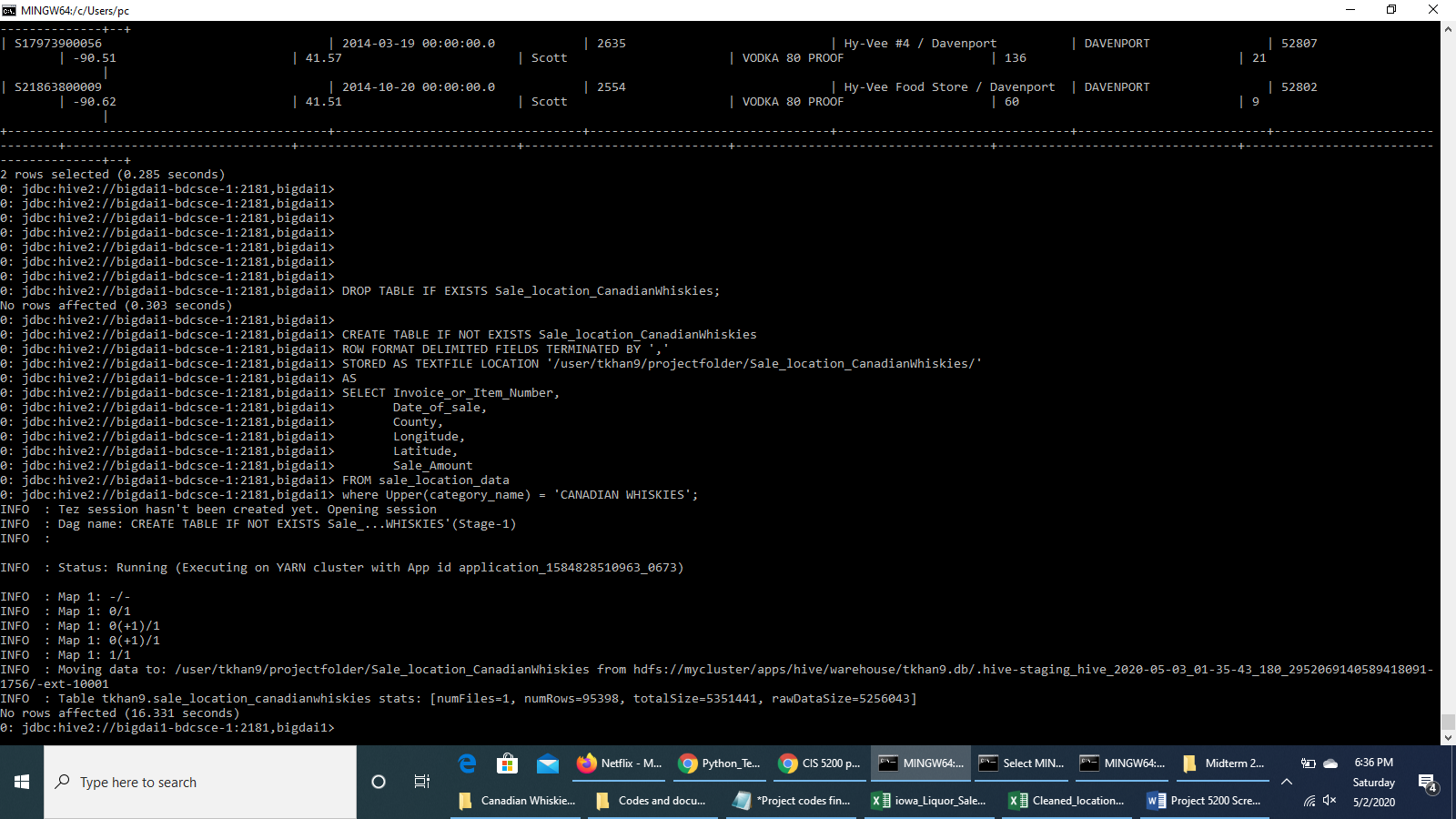
Longitude,

Latitude,

Sale\_Amount

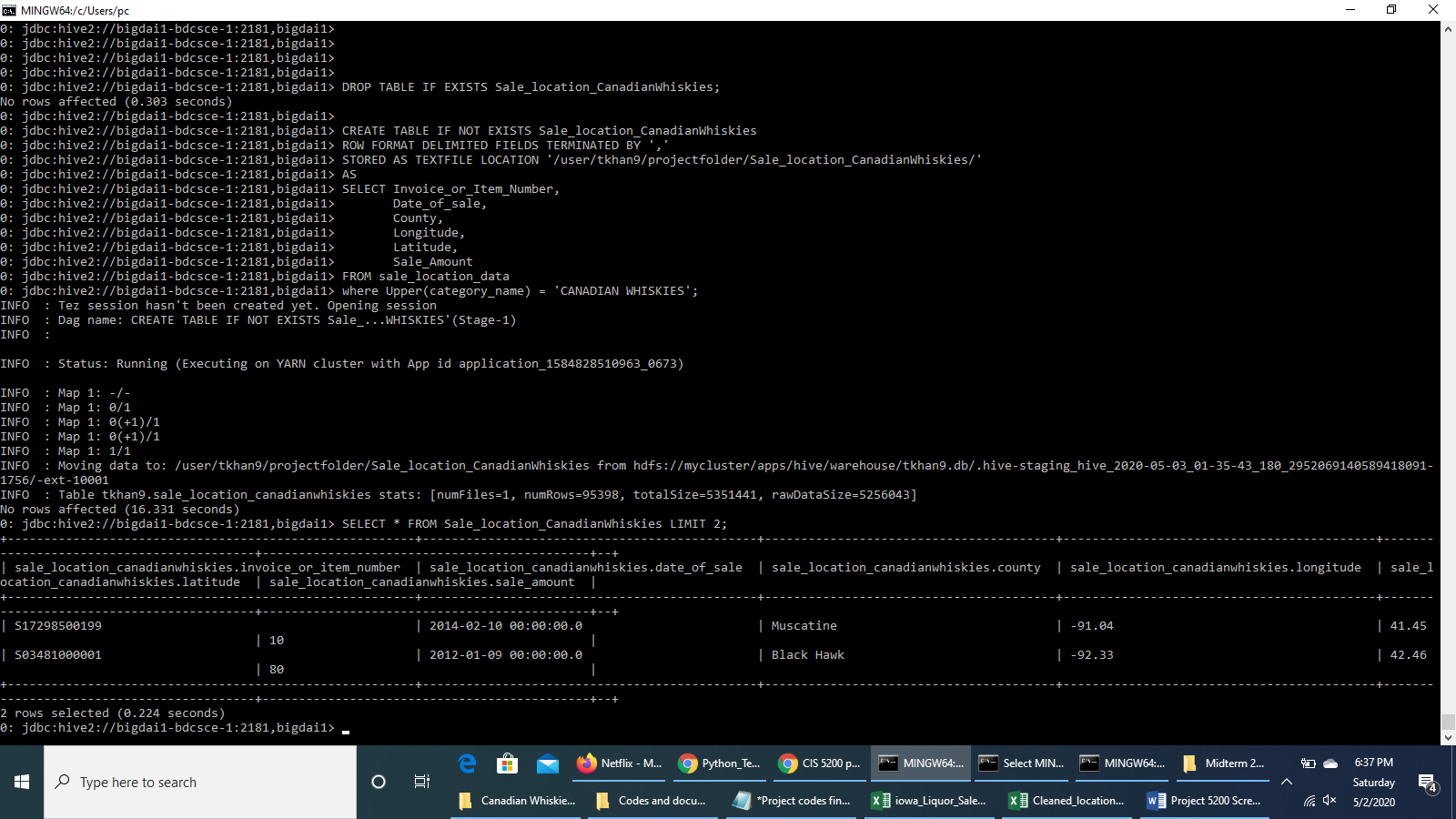
FROM sale\_location\_data

where Upper(category\_name) = 'CANADIAN WHISKIES';



**We run a query to see the resulting data upto 2 rows**

SELECT \* FROM Sale\_location\_CanadianWhiskies LIMIT 2;



**Press CTRL+Z to exit Beeline**

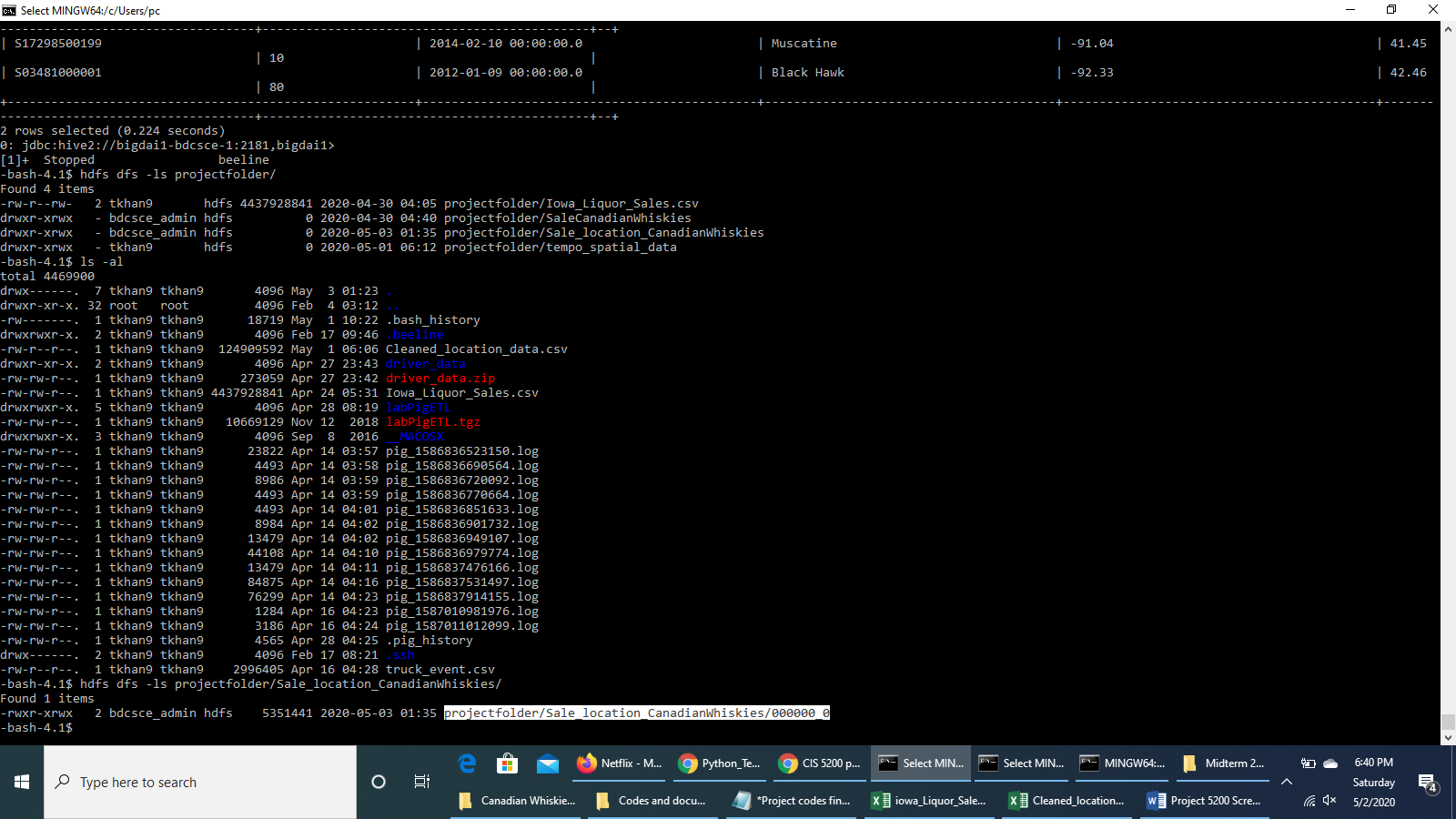
**Check HDFS path of the new output file 000000\_0**

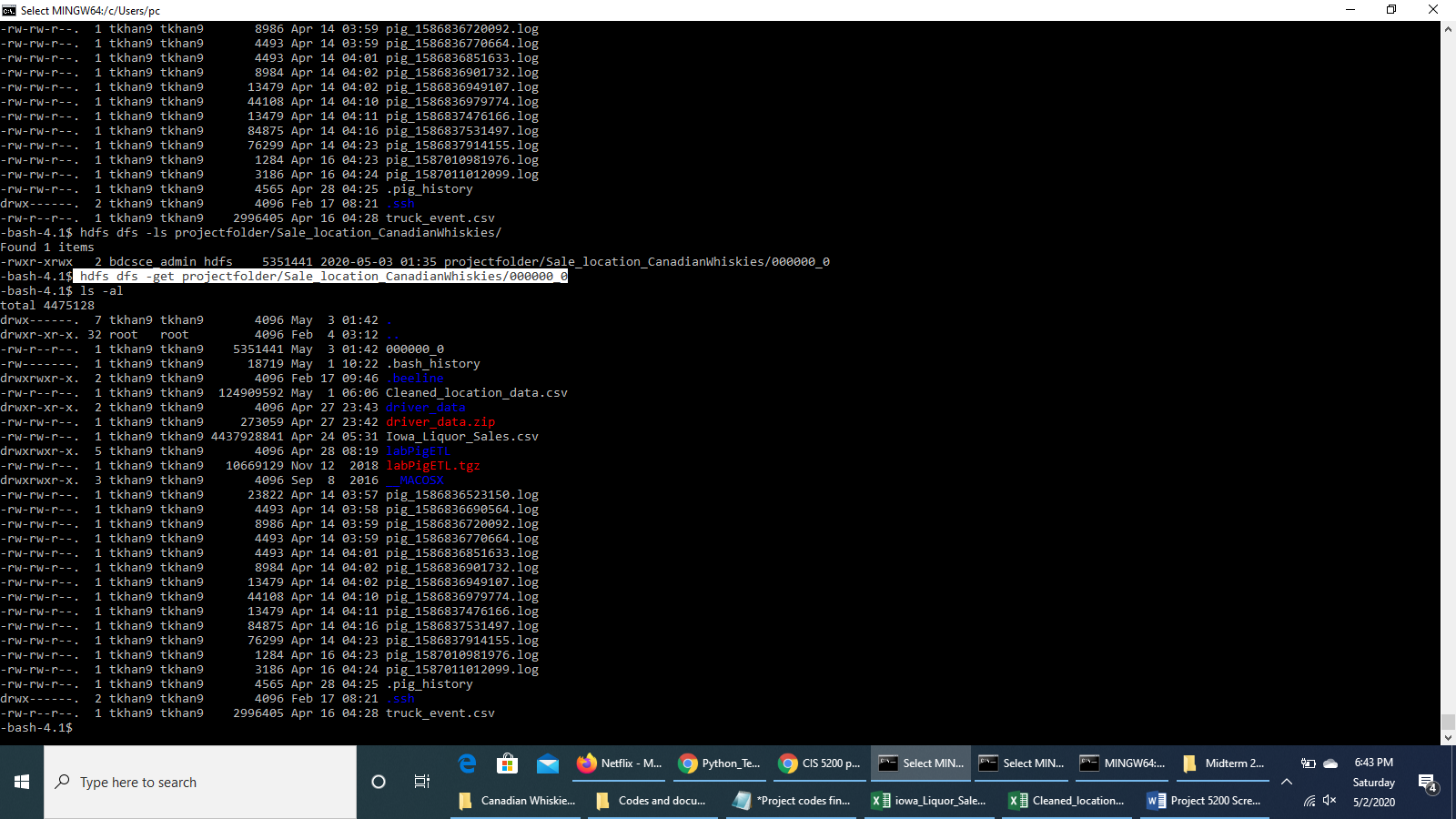
hdfs dfs -ls projectfolder/Sale\_Location\_CanadianWhiskies/

**To download the output file to the local file systems**

hdfs dfs -get projectfolder/Sale\_location\_CanadianWhiskies/000000\_0

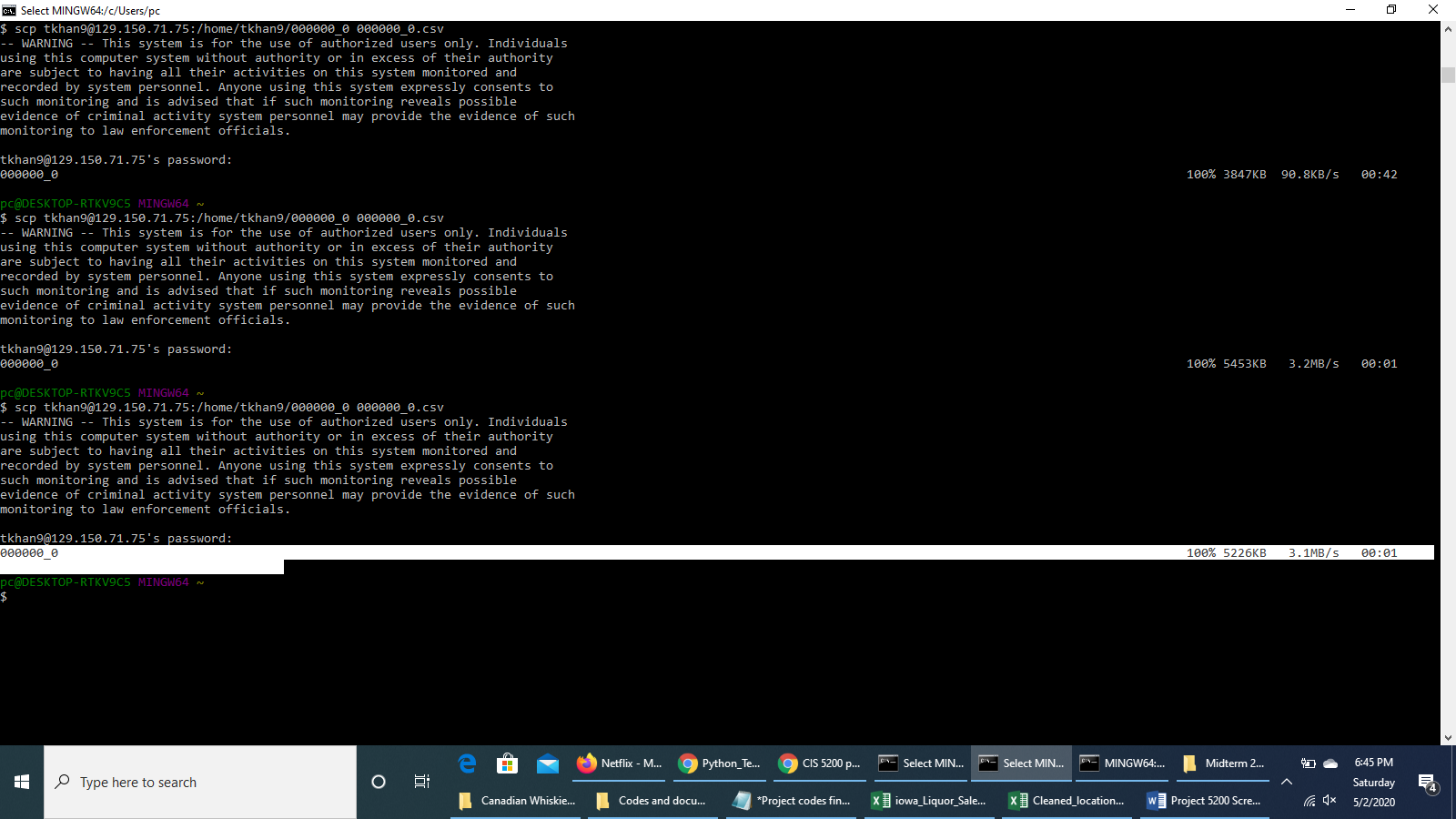
ls -al





**Open another terminal with git bash, minty, or putty in order to read/import the output file using your lab computer (or your PC/Laptop) - you have to download the file to your computer (or your PC/Laptop).**

scp [tkhan9@129.150.71.75:/home/tkhan9/000000\_0 000000\_0.csv](mailto:tkhan9@129.150.71.75:/home/tkhan9/000000_0%20000000_0.csv)



**Visualization on Excel 3D Map**

We opened the csv output file, inserted column headers as Invoice\_or\_Item\_Number, Date\_of\_sale, Longitude, Latitude and Sale\_Amount. We saved files in .xlsx format and renamed it to *Canadian\_Whiskies\_Sale\_results*. We opened the 3D Map and selected the fields as follows:-

**Location :** Latitude

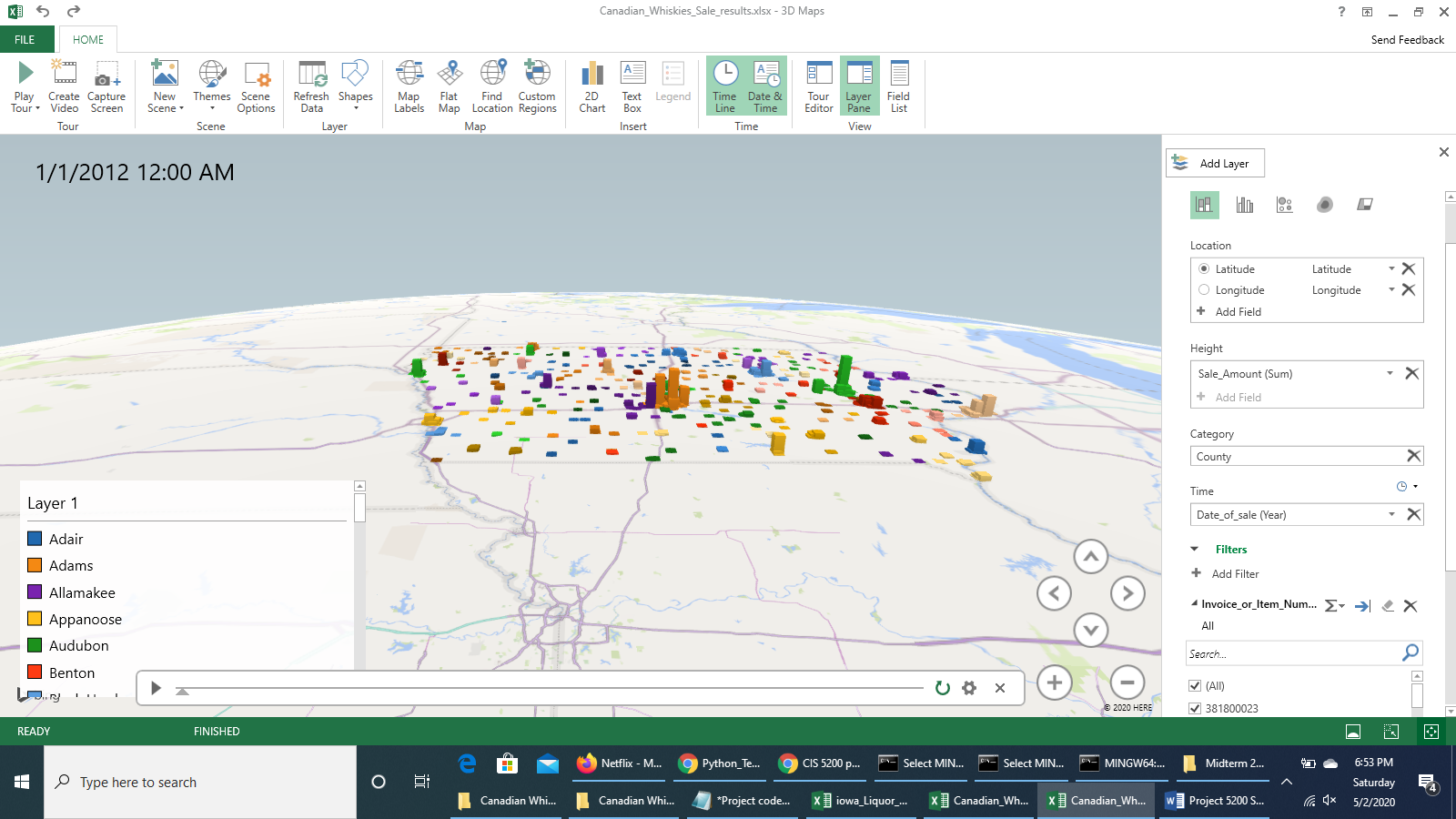
**Height :**  Sale\_Amount (Sum)

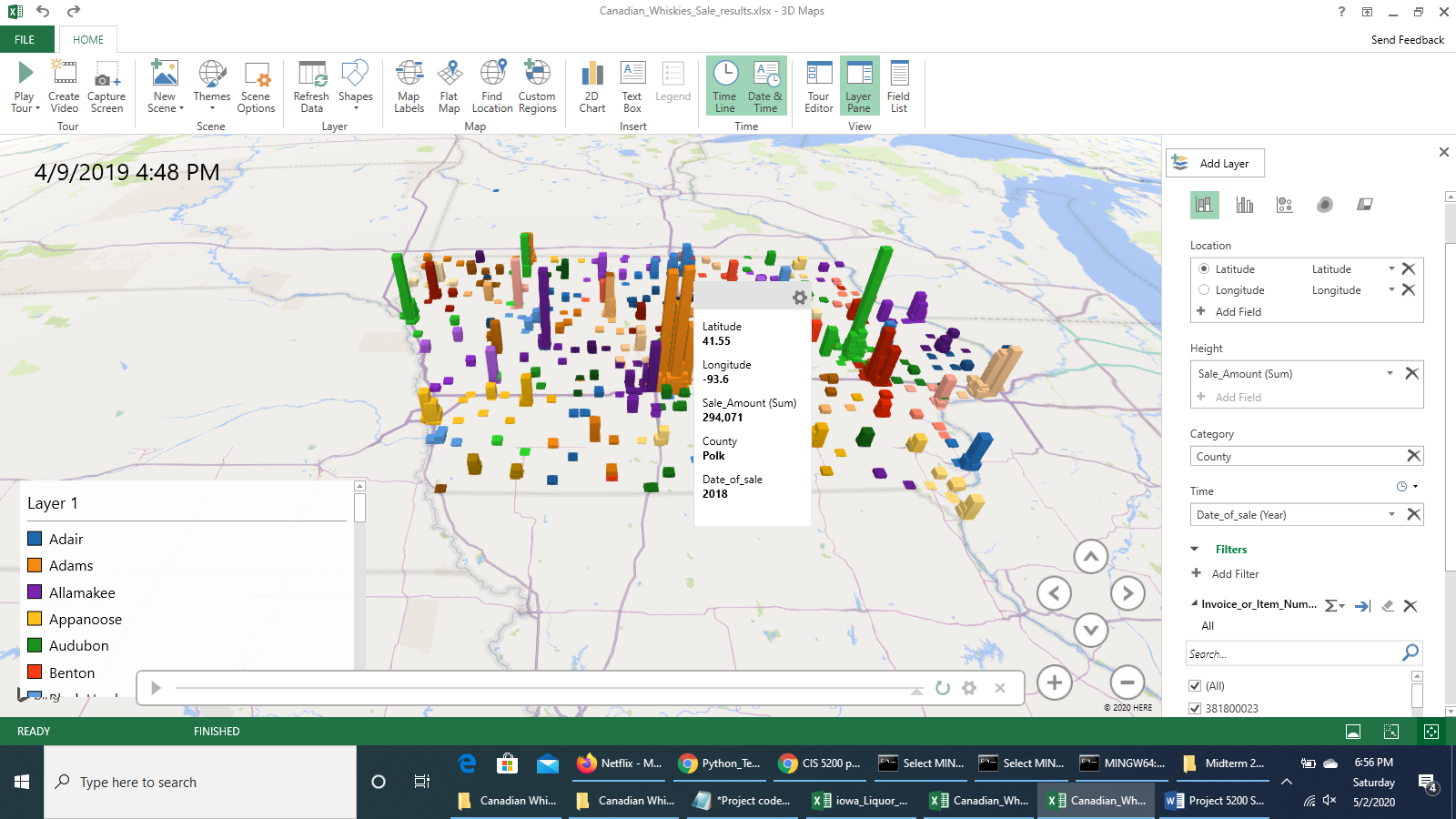
**Category :** County

**Time :** Date\_of\_sale (Year)

**Filter > Add filter :** Invoice\_or\_Item\_Number (All)

Selected ***Stacked Column*** type data visualization.

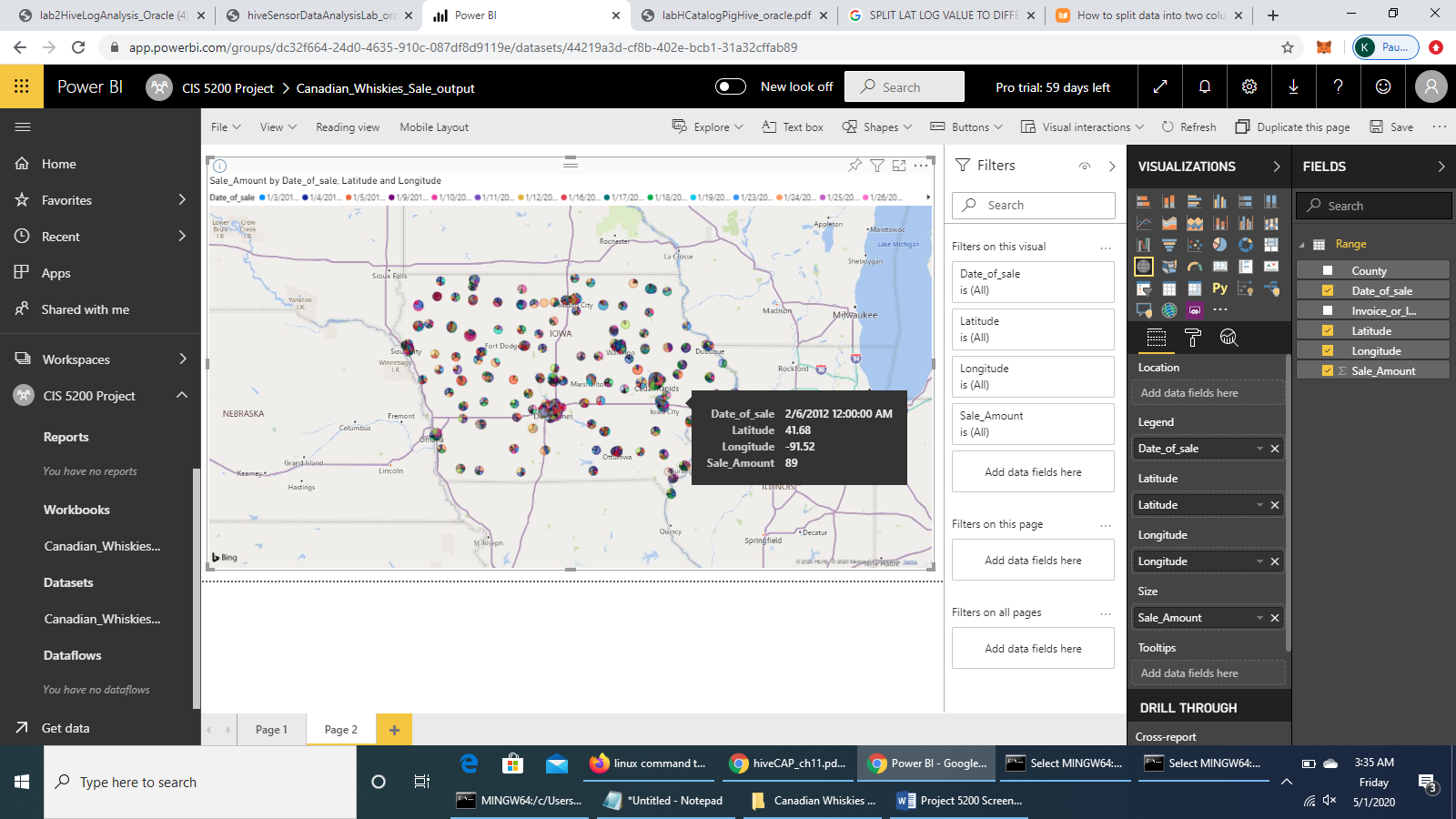




**Youtube Link to 3D Map Video capture:**

<https://youtu.be/nYV3qjIAUEI>

**Visualization on Power BI**



**Link of this report on Power BI**

<https://app.powerbi.com/groups/dc32f664-24d0-4635-910c-087df8d9119e/reports/bfa9f9a7-e08a-4518-9f9b-6e587f1f5a2e?ctid=ce8a2002-448f-4f58-82b1-d86f73e3afdd>