**Hive challenge task 3**

Scenario Based questions:

**1)Will the reducer work or not if you use “Limit 1” in any HiveQL query?**

For example

If your query is a simple select query then no reducers are called.

If your query has something like aggregation along with group by or order by and lets say you are using MR as your execution engine then reducers will be called.

If you are using Tez as your execution engine then reducers will most likely not be called because Tez uses a cost bases query optimizer which will try to do things in the map phase itself if the dataset is small and pertaining that there are no joins.

You can always use the explain key word to see what map phases and reduce phase are involved in the query. That’s the best way to determine what goes on in each phase.

**2)Suppose I have installed Apache Hive on top of my Hadoop cluster using default metastore configuration.**

**Then, what will happen if we have multiple clients trying to access Hive at the same time?**

The default metastore configuration allows only one Hive session to be opened at a time for accessing the metastore. Therefore, if multiple clients try to access the metastore at the same time, they will get an error. One has to use a standalone metastore, i.e. Local or remote metastore configuration in Apache Hive for allowing access to multiple clients concurrently.

Following are the steps to configure MySQL database as the local metastore in Apache Hive:

One should make the following changes in hive-site.xml:

javax.jdo.option.ConnectionURL property should be set to jdbc:mysql://host/dbname?createDataba

seIfNotExist=true.

javax.jdo.option.ConnectionDriverName property should be set to com.mysql.jdbc.Driver.

One should also set the username and password as:

javax.jdo.option.ConnectionUserName is set to desired username.

javax.jdo.option.ConnectionPassword is set to the desired password.

The JDBC driver JAR file for MySQL must be on the Hive's classpath, i.e. The jar file should be copied into the Hive's lib directory.

Now, after restarting the Hive shell, it will automatically connect to the MySQL database which is running as a standalone metastore.

**3)Suppose, I create a table that contains details of all the transactions done by the customers:**

**CREATE TABLE transaction\_details (cust\_id INT, amount FLOAT, month STRING, country STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;**

**Now, after inserting 50,000 records in this table, I want to know the total revenue generated for each month.**

**But, Hive is taking too much time in processing this query. How will you solve this problem and list the steps that I will be taking in order to do so?**

We can solve this problem of query latency by partitioning the table according to each month. So, for each month we will be scanning only the partitioned data instead of whole data sets.

As we know, we can't partition an existing non-partitioned table directly. So, we will be taking following steps to solve the very problem:

Create a partitioned table, say partitioned\_transaction:

CREATE TABLE partitioned\_transaction (cust\_id INT, amount FLOAT, country STRING) PARTITIONED BY (month STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

Enable dynamic partitioning in Hive:

SET hive.exec.dynamic.partition = true;

SET hive.exec.dynamic.partition.mode = nonstrict;

Transfer the data from the non – partitioned table into the newly created partitioned table:

INSERT OVERWRITE TABLE partitioned\_transaction PARTITION (month) SELECT cust\_id, amount, country, month FROM transaction\_details;

Now, we can perform the query using each partition and therefore, decrease the query time.

**4)How can you add a new partition for the month December in the above partitioned table?**

I am inserting data into a table based on partitions dynamically.

But, I received an error – FAILED ERROR IN SEMANTIC ANALYSIS: Dynamic partition strict mode requires at least one static partition column. How will you remove this error?

For adding a new partition in the above table partitioned\_transaction, we will issue the command give below:

ALTER TABLE partitioned\_transaction ADD PARTITION (month='Dec') LOCATION '/partitioned\_transaction';

**5)Suppose, I have a CSV file – 'sample.csv' present in '/temp' directory with the following entries:**

**id first\_name last\_name email gender ip\_address**

**How will you consume this CSV file into the Hive warehouse using built-in SerDe?**

SerDe stands for serializer/deserializer. A SerDe allows us to convert the unstructured bytes into a record that we can process using Hive. SerDes are implemented using Java. Hive comes with several built-in SerDes and many other third-party SerDes are also available.

Hive provides a specific SerDe for working with CSV files. We can use this SerDe for the sample.csv by issuing following commands:

CREATE EXTERNAL TABLE sample

(id int, first\_name string,

last\_name string, email string,

gender string, ip\_address string)

ROW FORMAT SERDE 'org.apache.hadoop.hive.serde2.OpenCSVSerde'

STORED AS TEXTFILE LOCATION '/temp';

Now, we can perform any query on the table 'sample':

SELECT first\_name FROM sample WHERE gender = 'male';

**6)Suppose, I have a lot of small CSV files present in the input directory in HDFS and I want to create a single Hive table corresponding to these files.**

**The data in these files are in the format: {id, name, e-mail, country}. Now, as we know, Hadoop performance degrades when we use lots of small files.**

**So, how will you solve this problem where we want to create a single Hive table for lots of small files without degrading the performance of the system?**

One can use the SequenceFile format which will group these small files together to form a single sequence file. The steps that will be followed in doing so are as follows:

Create a temporary table:

CREATE TABLE temp\_table (id INT, name STRING, e-mail STRING, country STRING)

**7)LOAD DATA LOCAL INPATH 'Home/country/state/'**

**OVERWRITE INTO TABLE address;**

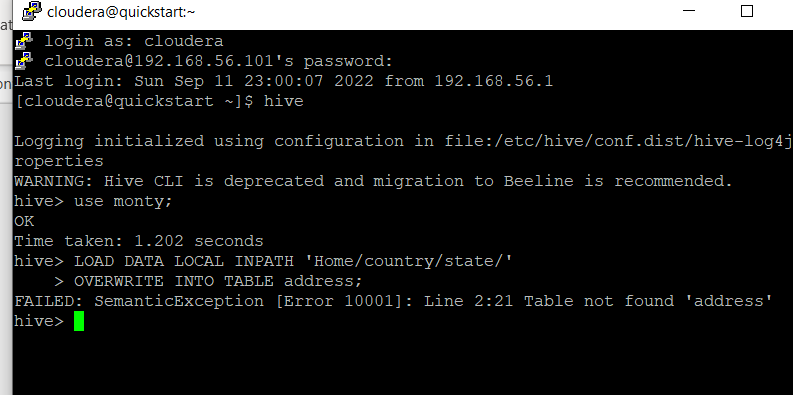
**The following statement failed to execute. What can be the cause?**

Complete file name missing

Correct syntax

LOAD DATA LOCAL INPATH 'Home/country/state/file.txt'

OVERWRITE INTO TABLE address;



**8)Is it possible to add 100 nodes when we already have 100 nodes in Hive? If yes, how?**

Yes, we can add the nodes by following the below steps:

1: Take a new system; create a new username and password

2: Install SSH and with the master node setup SSH connections

3: Add ssh public\_rsa id key to the authorized keys file

4: Add the new DataNode hostname, IP address, and other details in /etc/hosts slaves file:

192.168.1.102 slave3.in slave3

Step 5: Start the DataNode on a new node

Step 6: Login to the new node like suhadoop or:

ssh -X hadoop@192.168.1.103

Step 7: Start HDFS of the newly added slave node by using the following command:

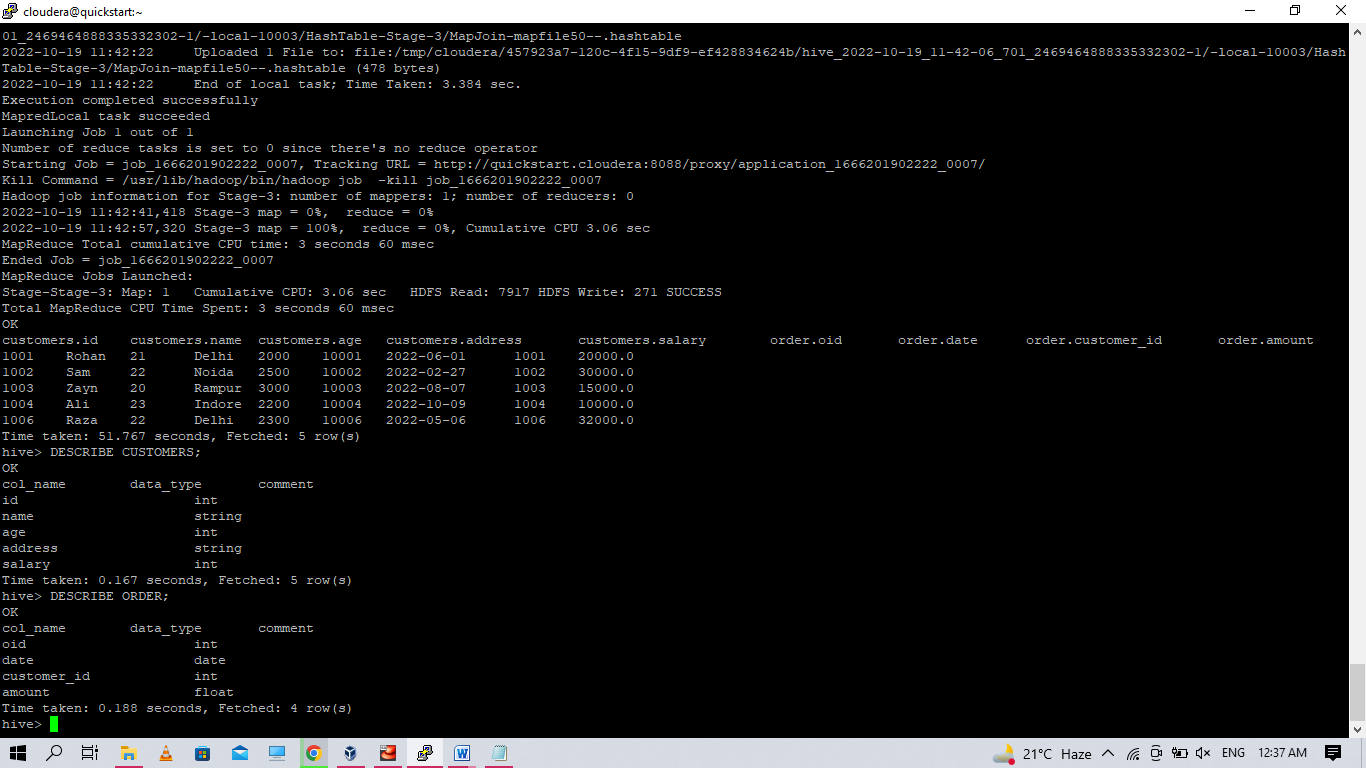
./bin/hadoop-daemon.sh start data node

**Hive Practical questions:**

Hive Join operations

Create a table named CUSTOMERS(ID | NAME | AGE | ADDRESS | SALARY)

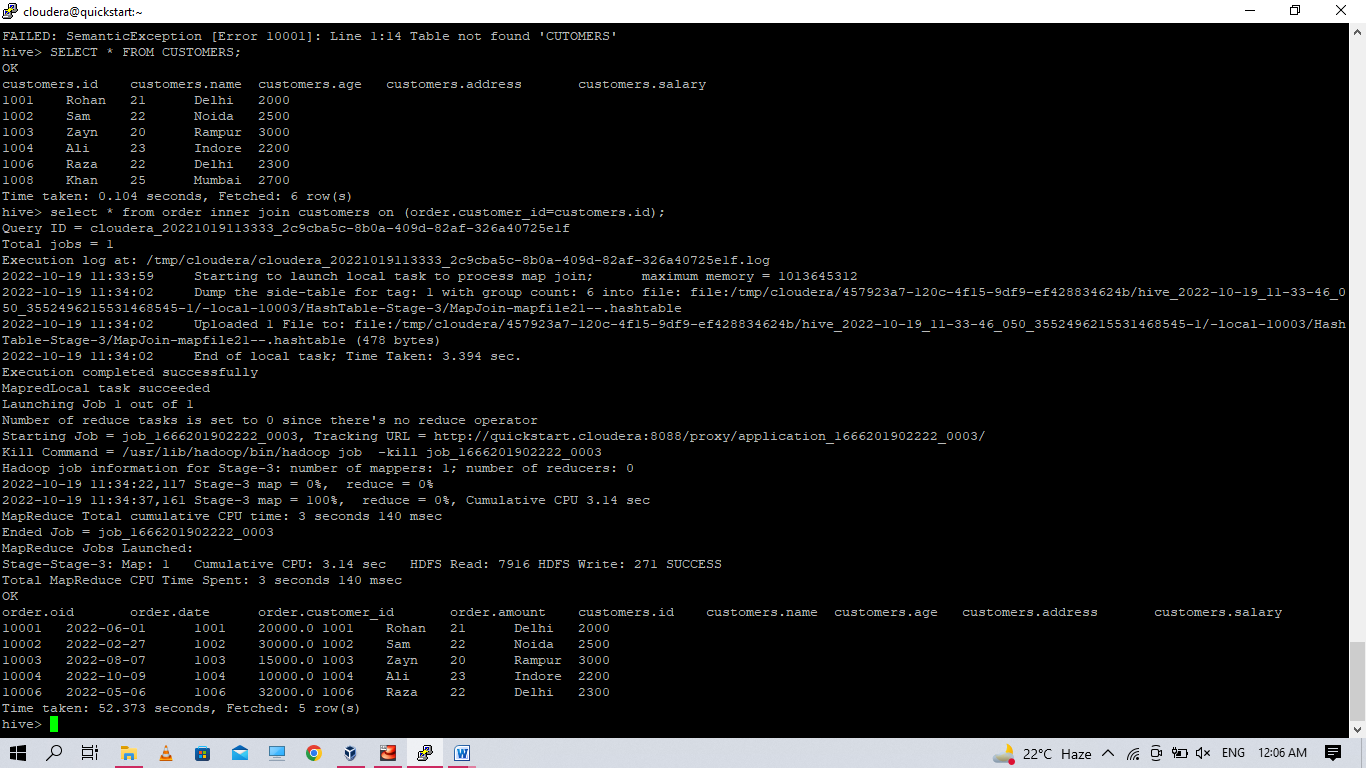
Create a Second table ORDER(OID | DATE | CUSTOMER\_ID | AMOUNT)



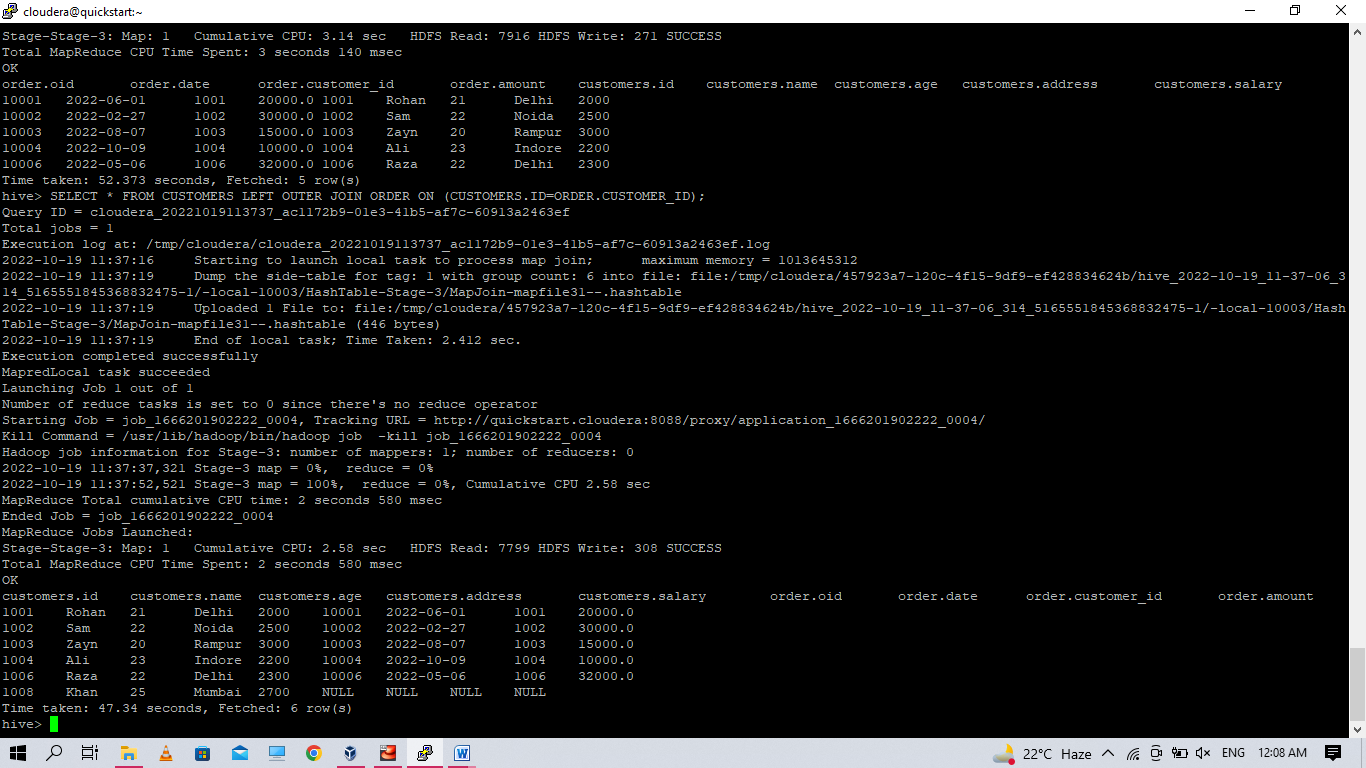
Now perform different joins operations on top of these tables

(Inner JOIN, LEFT OUTER JOIN ,RIGHT OUTER JOIN ,FULL OUTER JOIN)

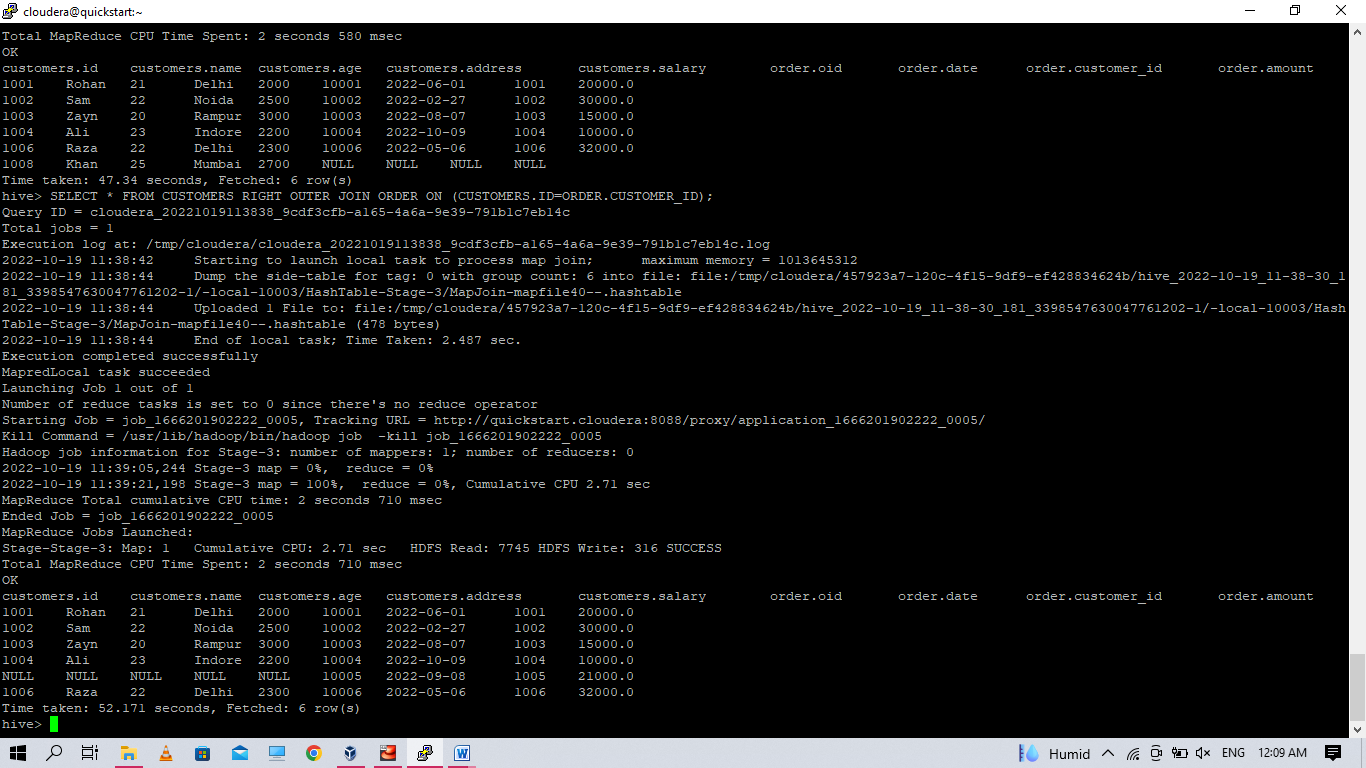
INNER JOIN



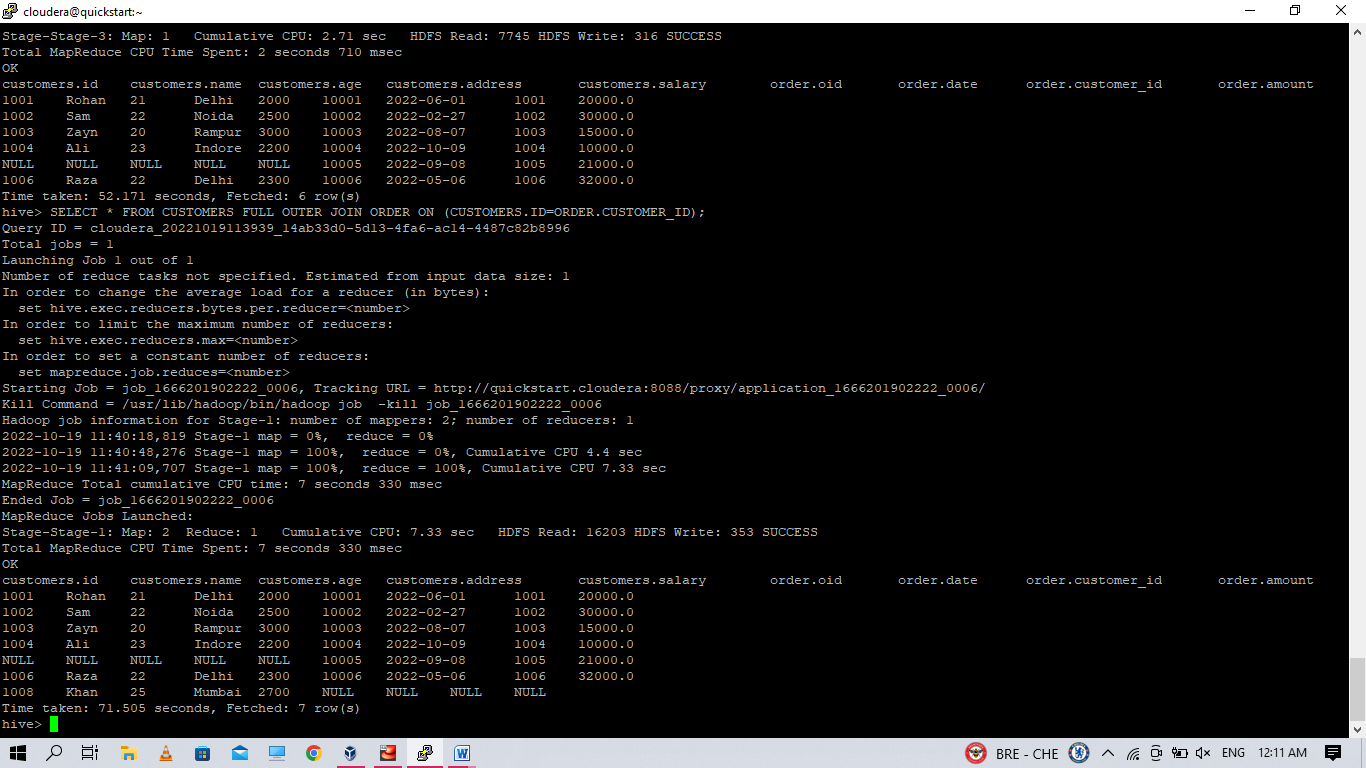
LEFT OUTER JOIN



RIGHT OUTER JOIN



FULL OUTER JOIN

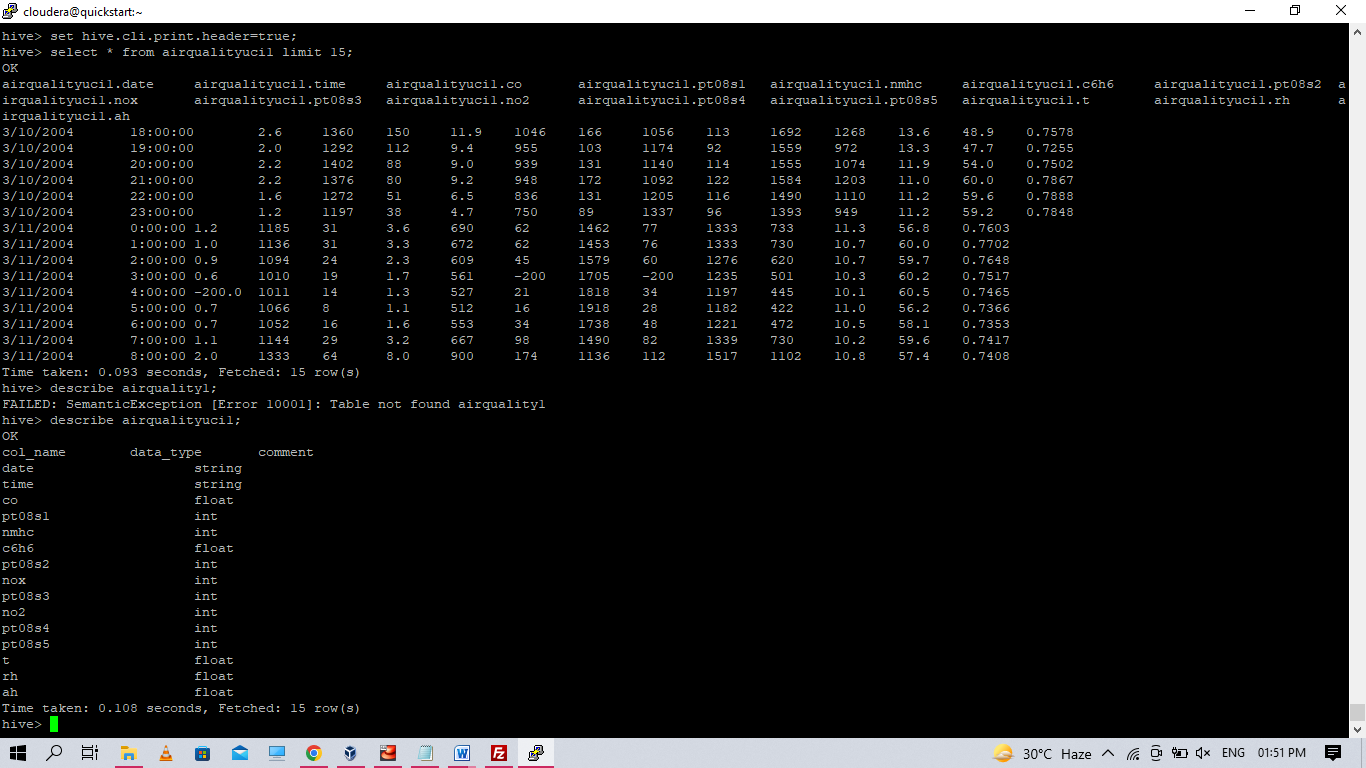


**BUILD A DATA PIPELINE WITH HIVE**

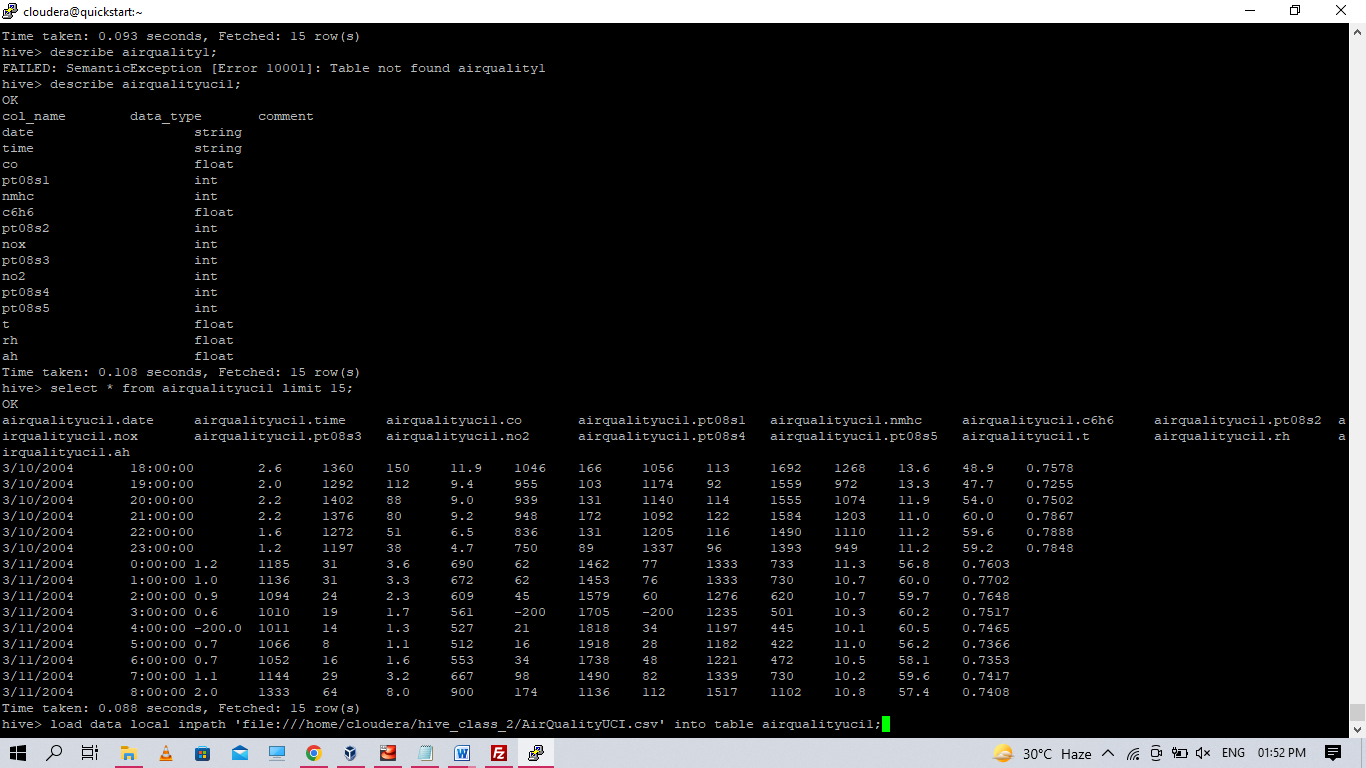
Download a data from the given location -

<https://archive.ics.uci.edu/ml/machine-learning-databases/00360/>

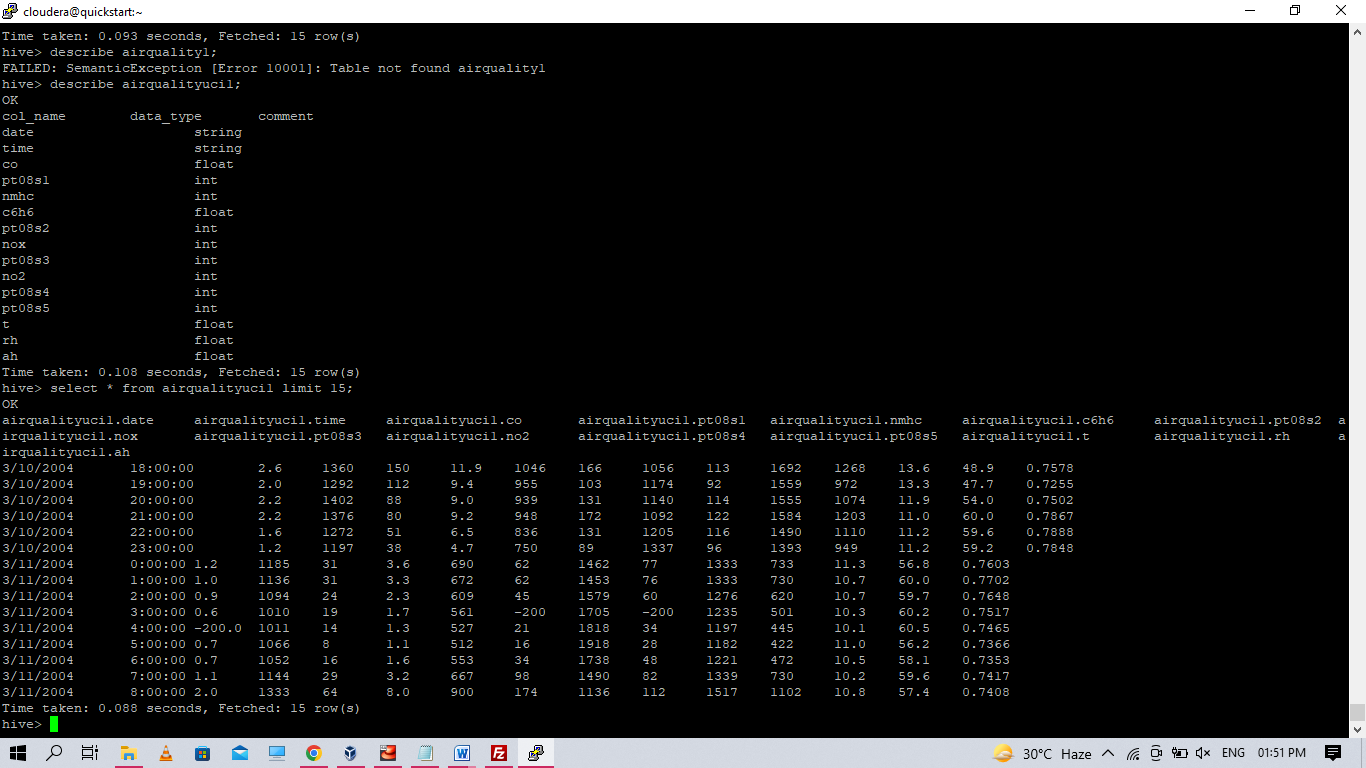
1. Create a hive table as per given schema in your dataset



1. try to place a data into table location

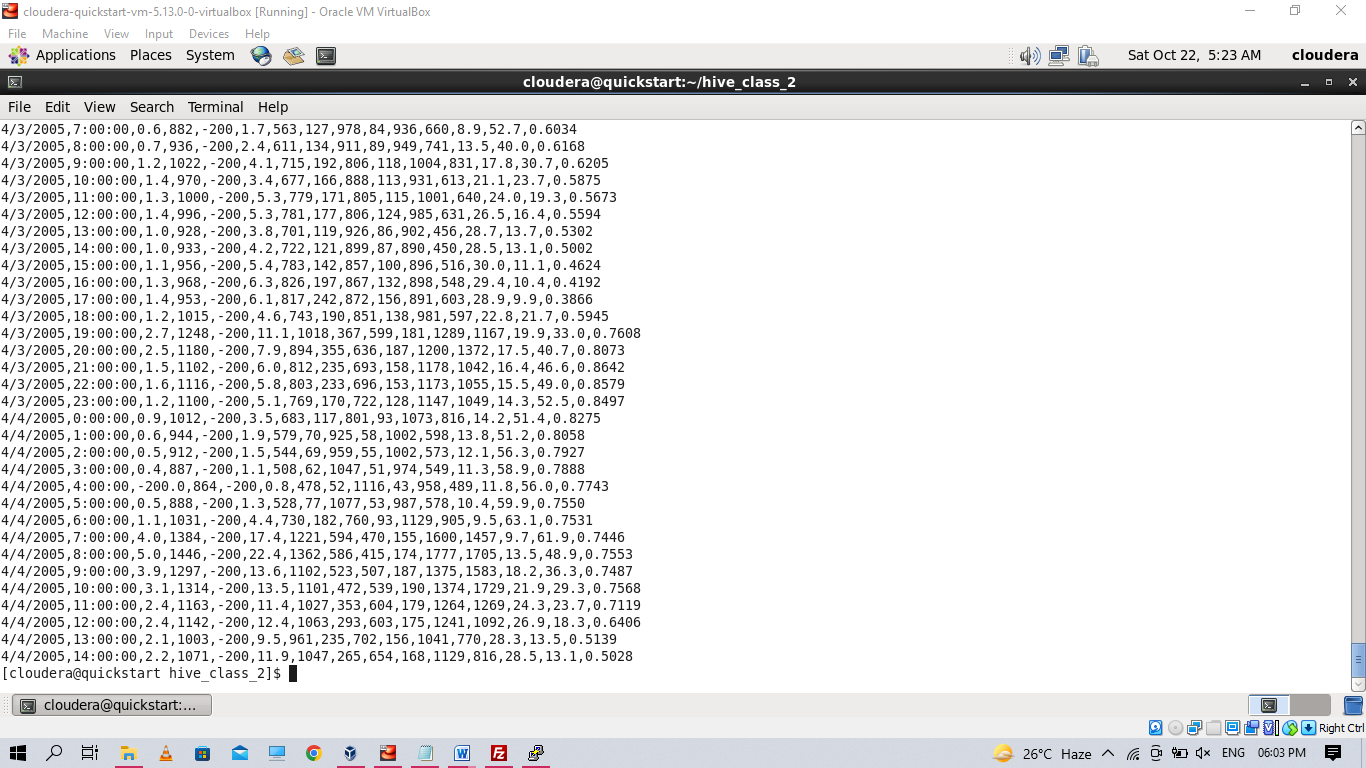


1. Perform a select operation

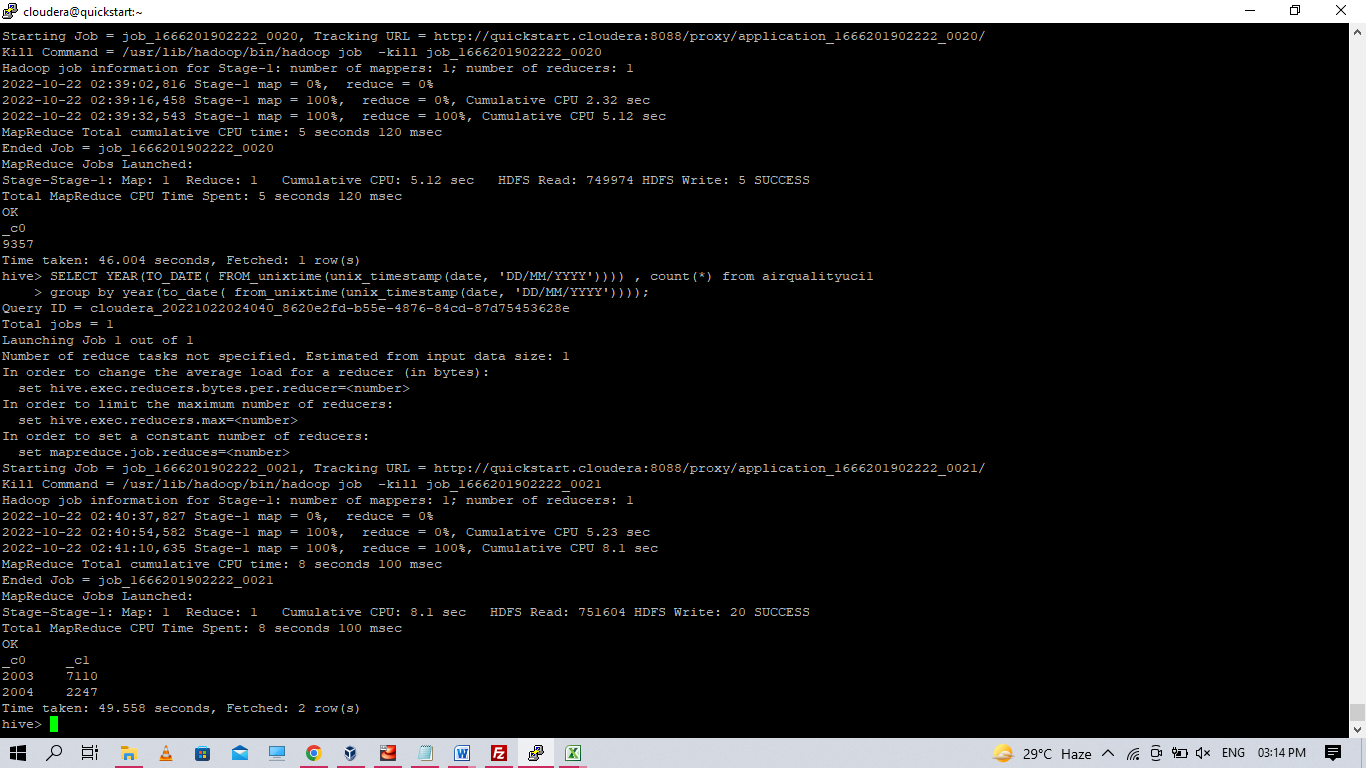


1. Fetch the result of the select operation in your local as a csv file .

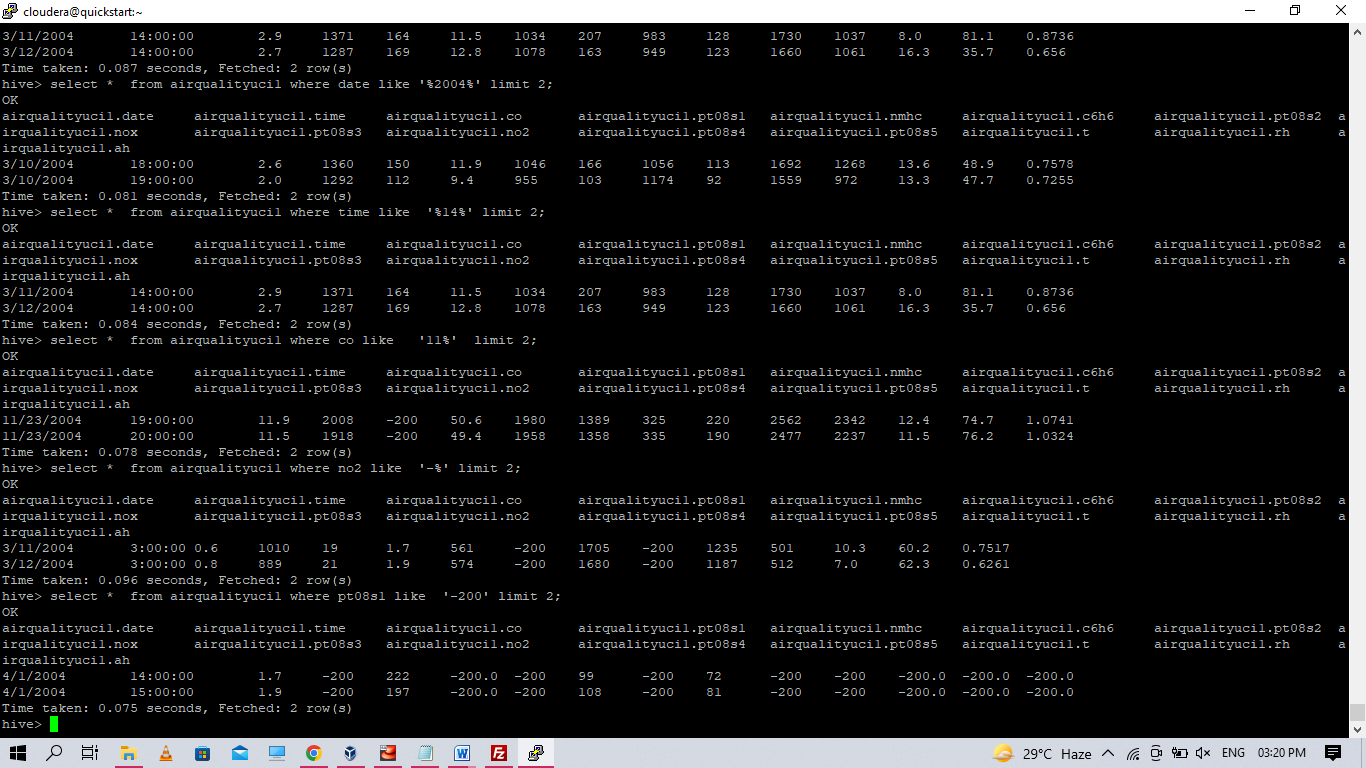
Cat AirQualityuci1.csv



1. Perform group by operation .



1. Perform filter operation at least 5 kinds of filter examples .



select \* from airqualityuci1 where date like '%2004%' limit 2;

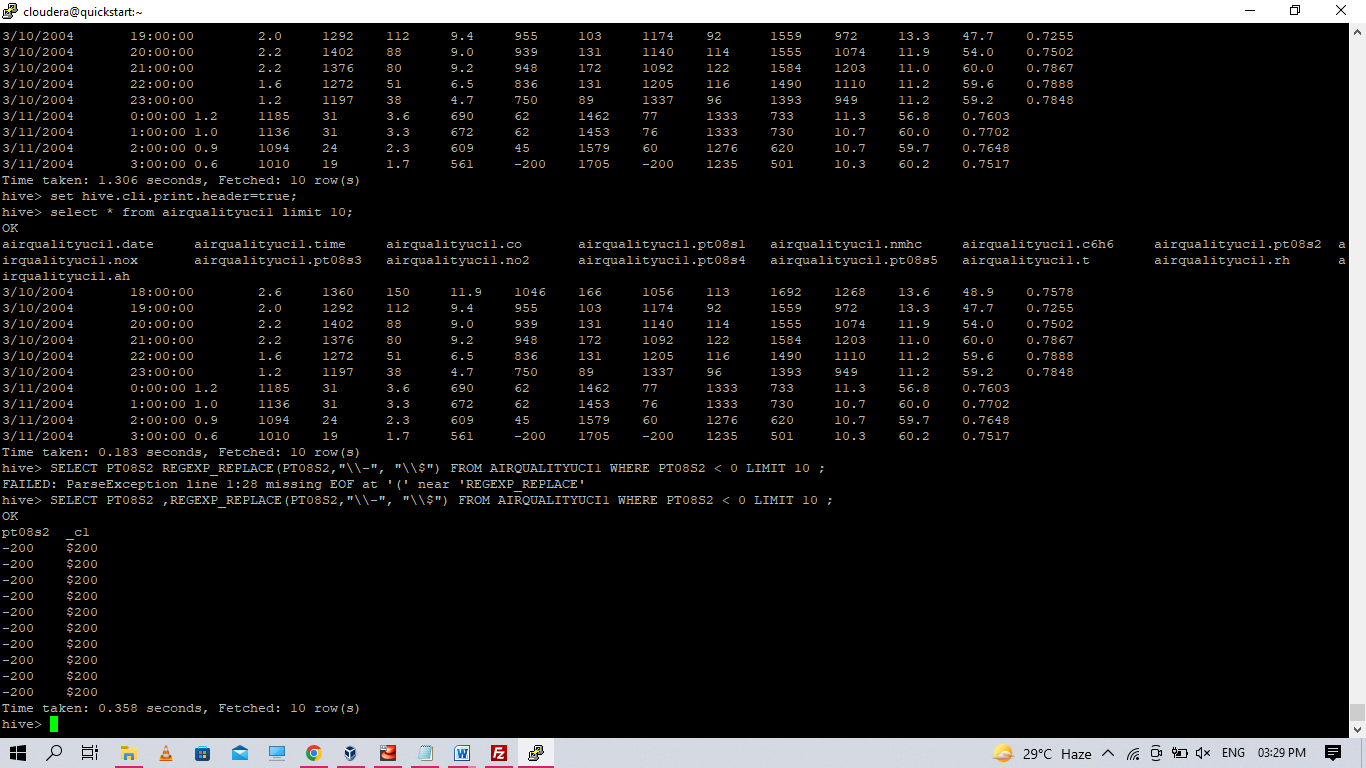
select \* from airqualityuci1 where time like '%14%' limit 2;

select \* from airqualityuci1 where co like '11%' limit 2;

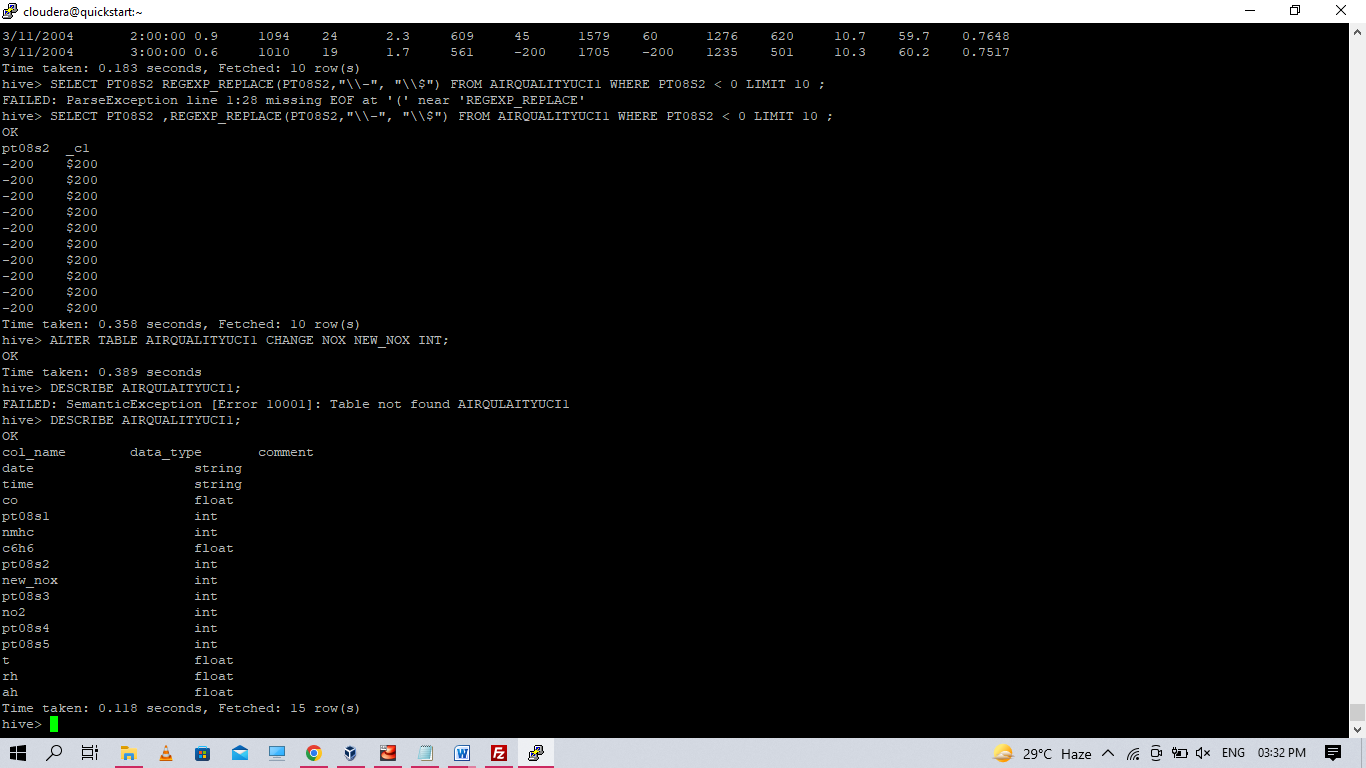
select \* from airqualityuci1 where no2 like '-%' limit 2;

select \* from airqualityuci1 where pt08s1 like '-200' limit 2;

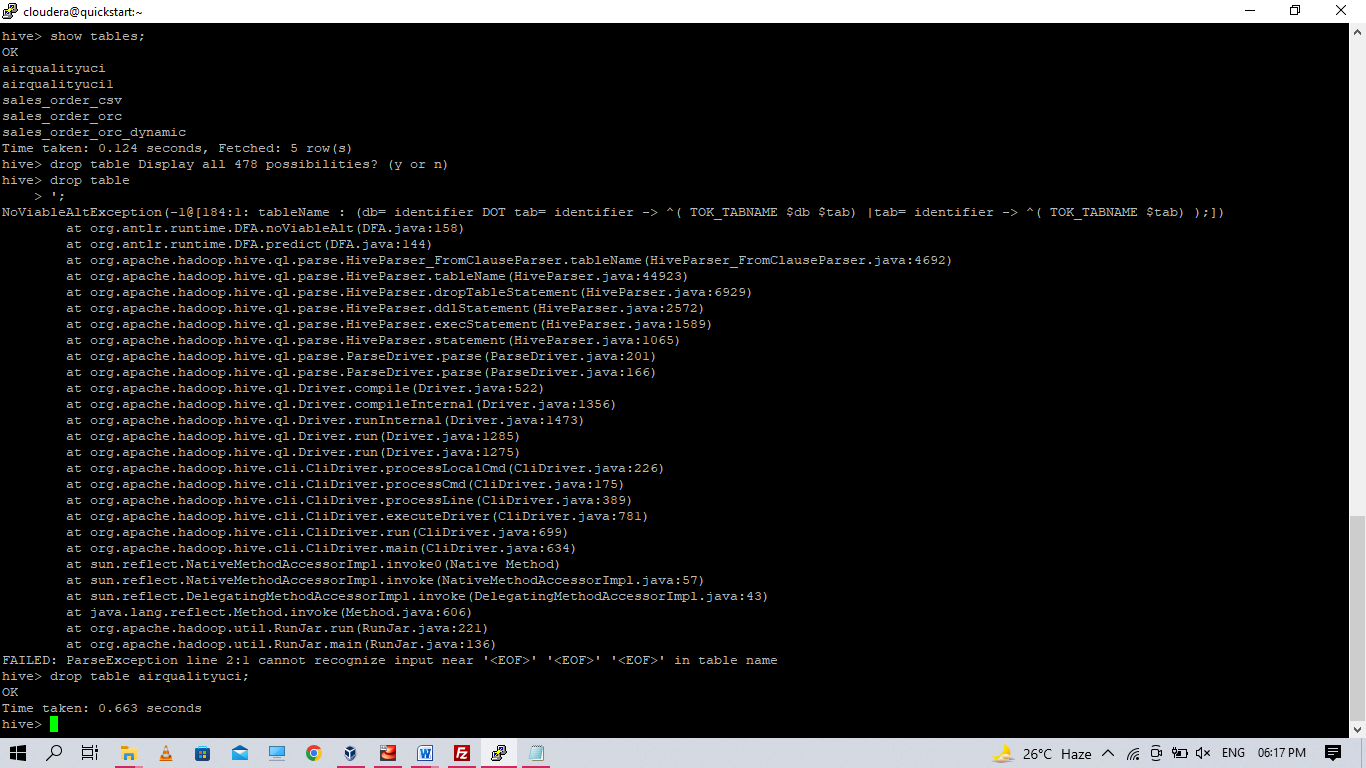
1. show and example of regex operation



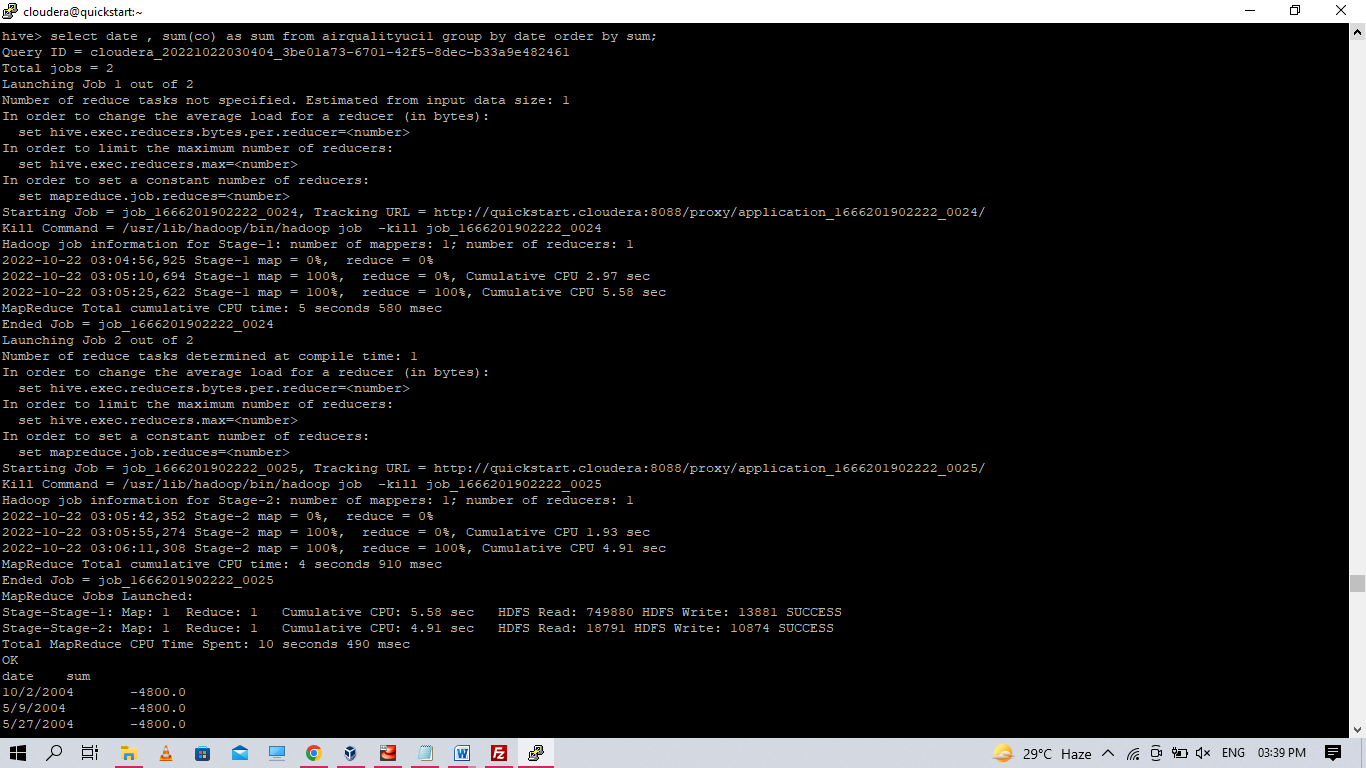
1. alter table operation



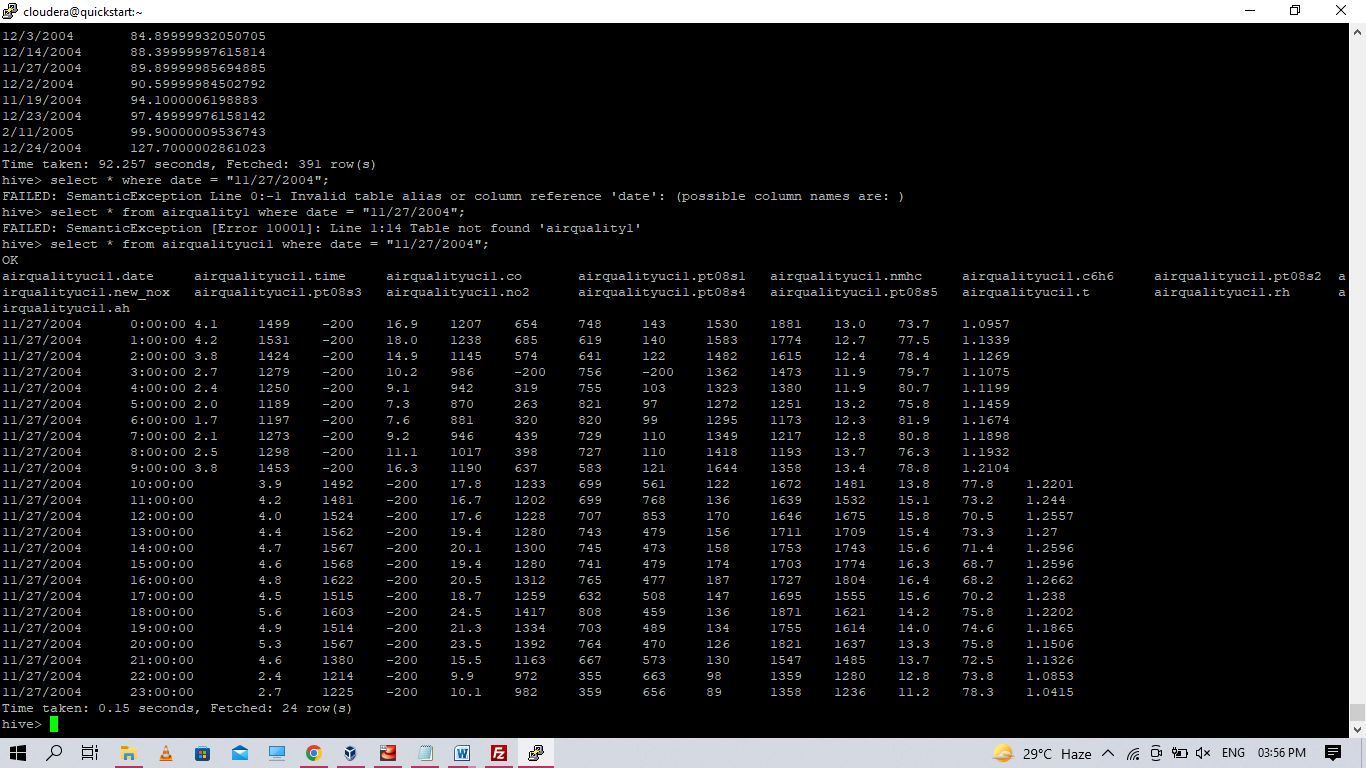
1. drop table operation



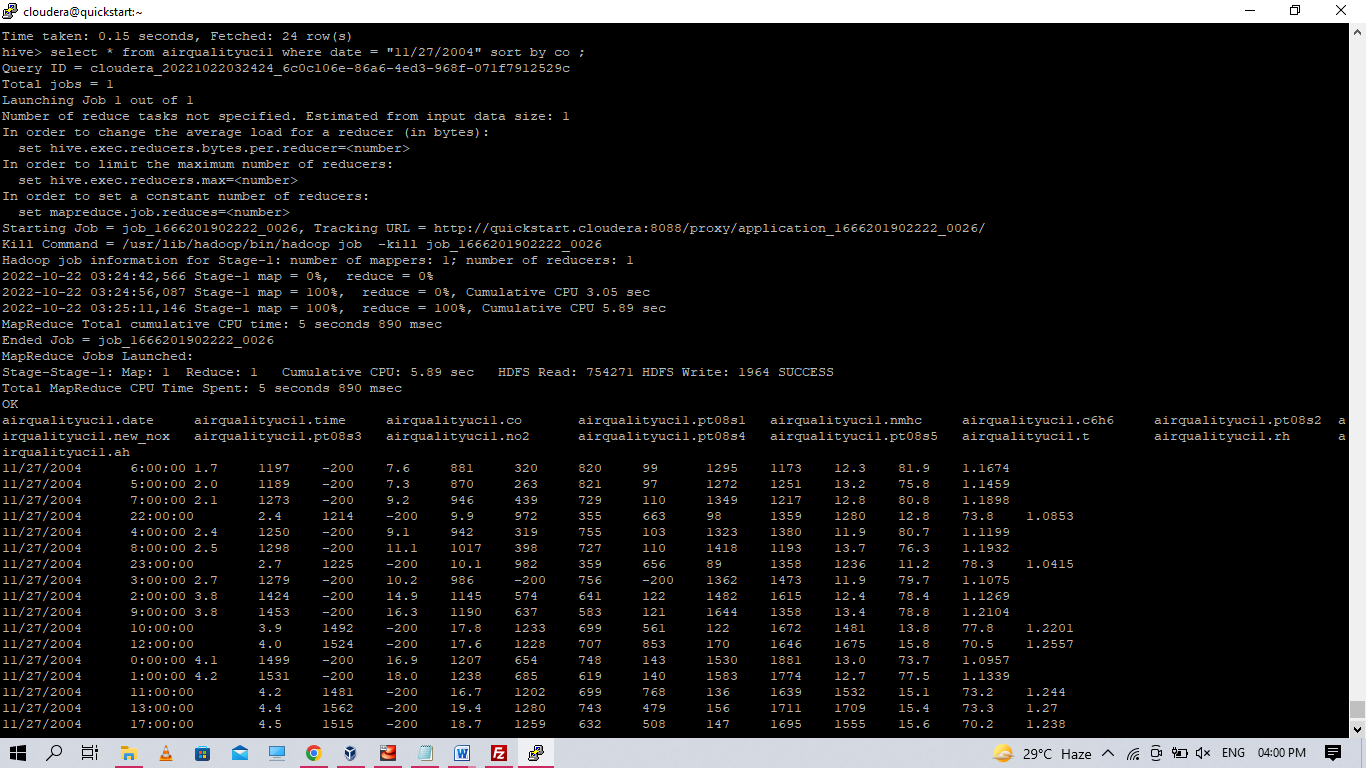
1. order by operation .



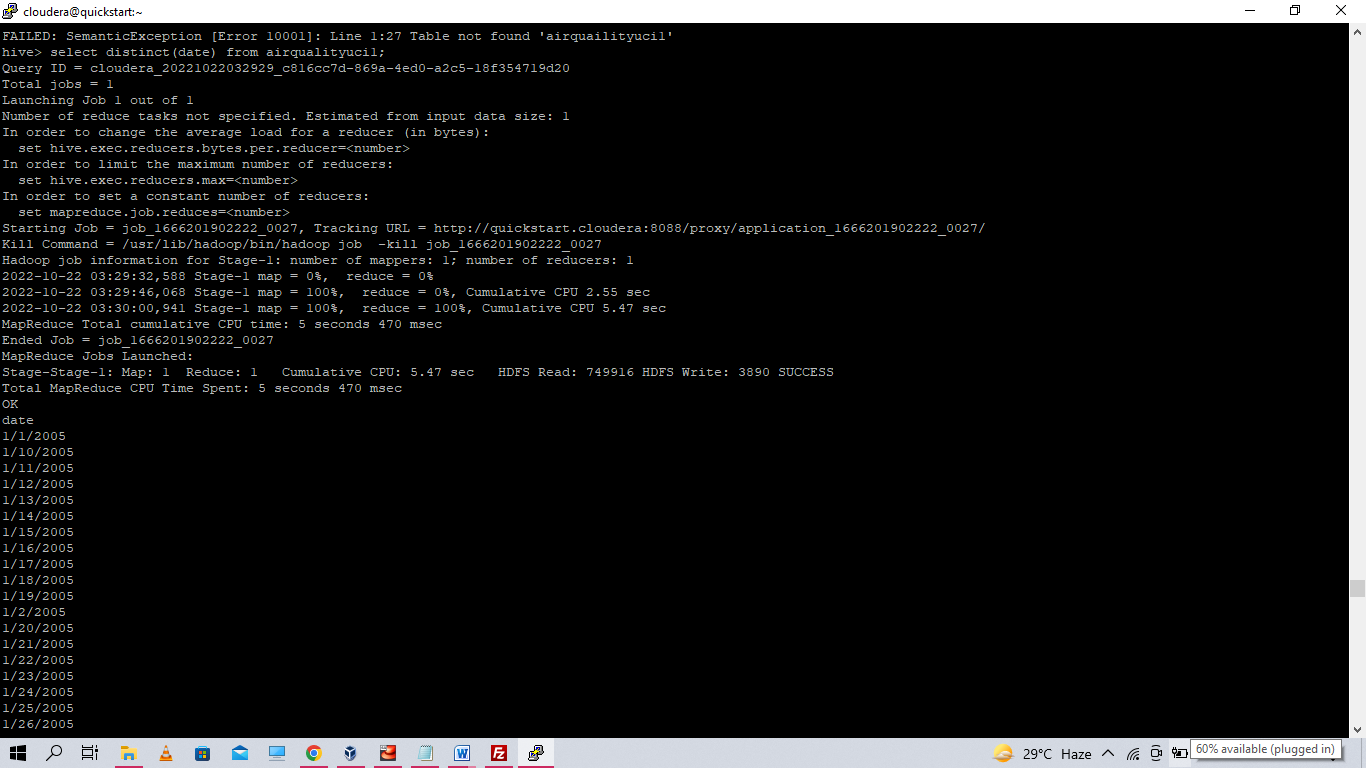
1. where clause operations you have to perform



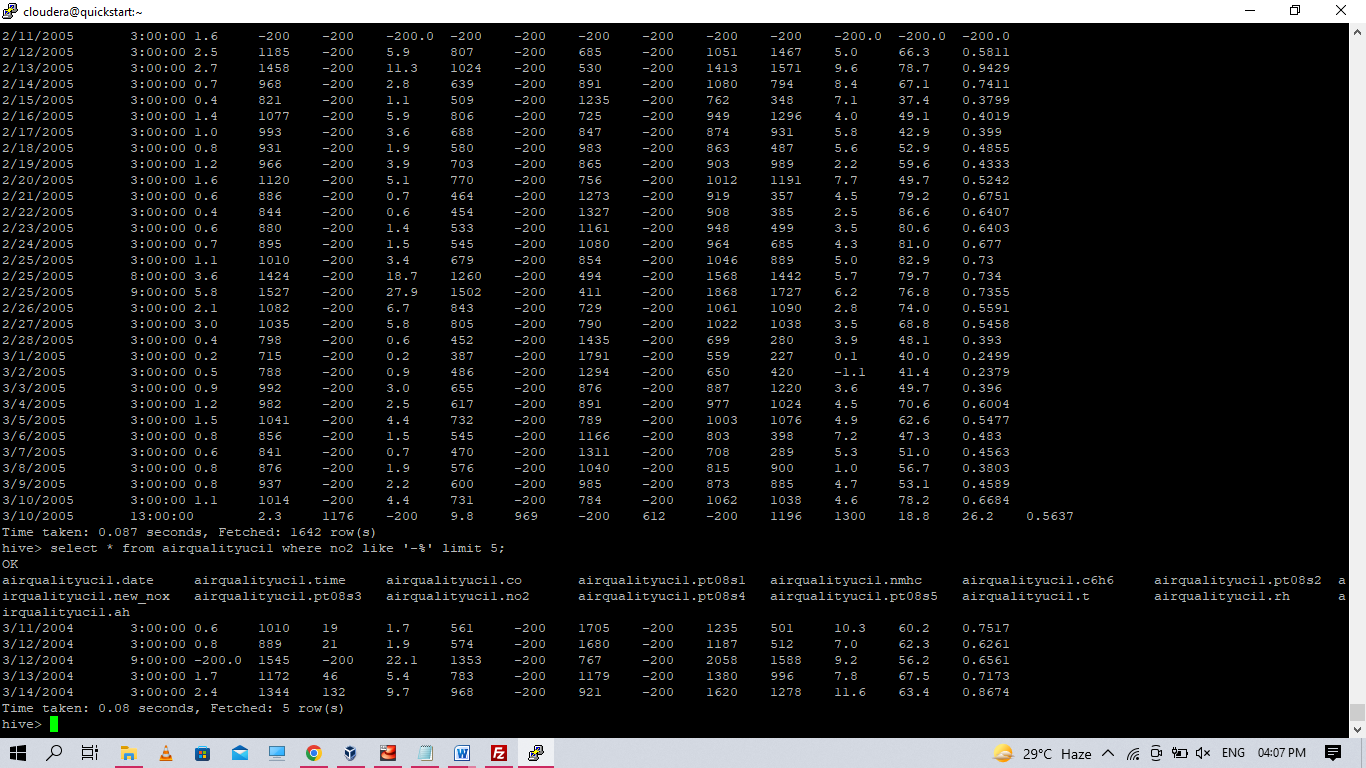
1. sorting operation you have to perform



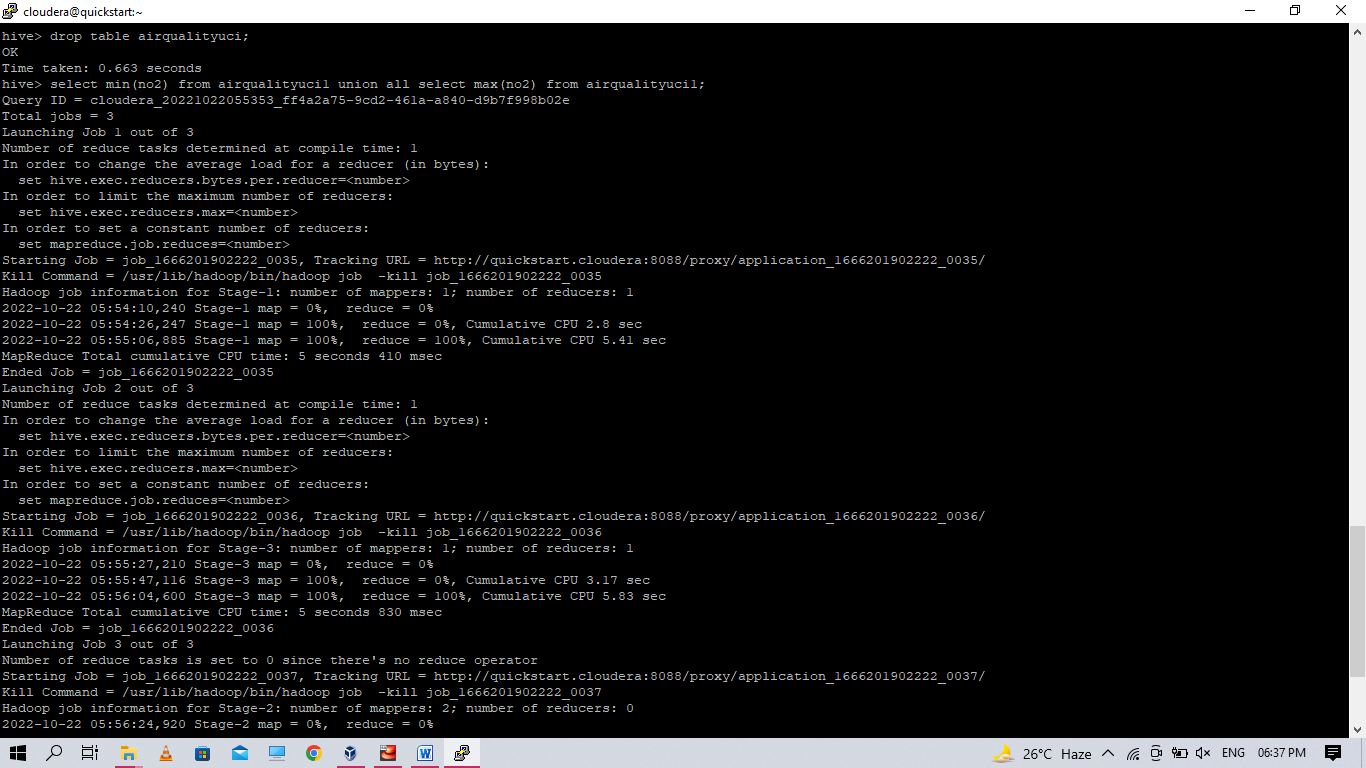
1. distinct operation you have to perform

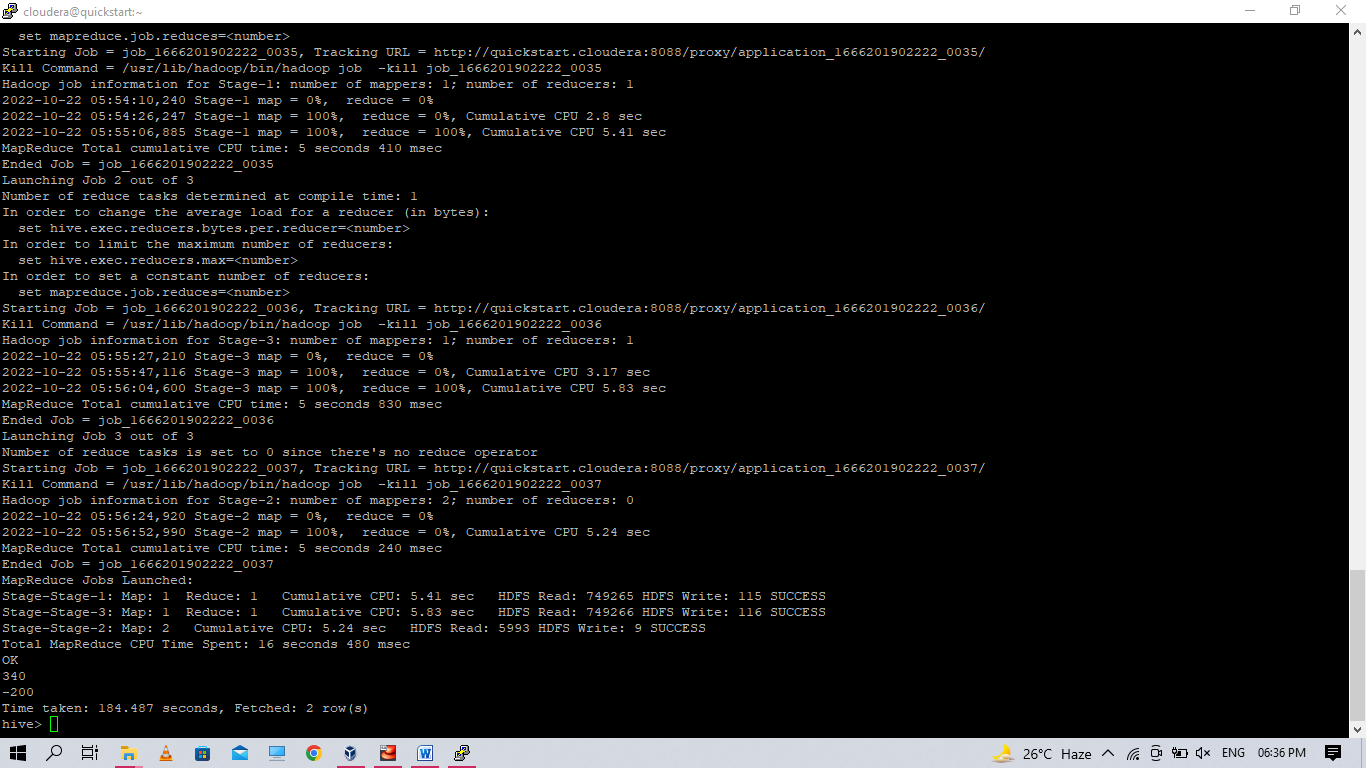


1. like an operation you have to perform .

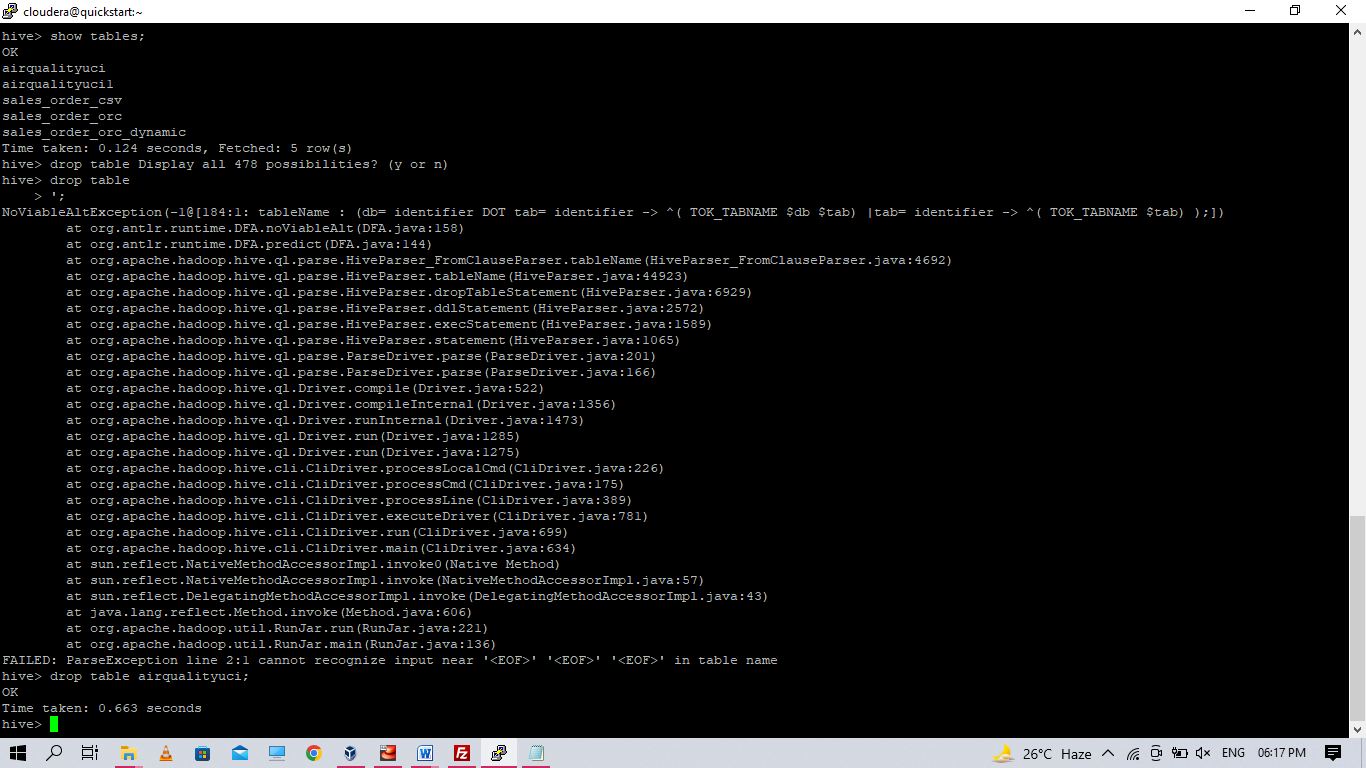


1. union operation you have to perform .





1. table view operation you have to perform .



hive operation with python

Create a python application that connects to the Hive database for extracting data, creating sub tables for data processing, drops temporary tables.fetch rows to python itself into a list of tuples and mimic the join or filter operations