Hive Assignment -1

**1. Download vehicle sales data -> https://github.com/shashank-mishra219/Hive-Class/blob/main/sales\_order\_data.csv**

**2. Store raw data into hdfs location**

hadoop fs -copyFromlocal "sales\_order\_data.csv" /tmp/hive\_data

**3. Create a internal hive table "sales\_order\_csv" which will store csv data sales\_order\_csv .. make sure to skip header row while creating table**

create table sales(

ORDERNUMBER int,

QUANTITYORDERED int,

PRICEEACH float,

ORDERLINENUMBER int,

SALES float,

STATUS string,

QTR\_ID int,

MONTH\_ID int,

YEAR\_ID int,

PRODUCTLINE string,

MSRP int,

PRODUCTCODE string,

PHONE string,

CITY string,

STATE string,

POSTALCODE string,

COUNTRY string,

TERRITORY string,

CONTACTLASTNAME string,

CONTACTFIRSTNAME string,

DEALSIZE string)

row format delimited

fields terminated by ','

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

**4. Load data from hdfs path into "sales\_order\_csv"**

Load data inpath '/tmp/hive\_data/sales\_order\_data.csv' into table sales;

**5. Create an internal hive table which will store data in ORC format "sales\_order\_orc"**

create table sales\_ORC(

ORDERNUMBER int,

QUANTITYORDERED int,

PRICEEACH float,

ORDERLINENUMBER int,

SALES float,

STATUS string,

QTR\_ID int,

MONTH\_ID int,

YEAR\_ID int,

PRODUCTLINE string,

MSRP int,

PRODUCTCODE string,

PHONE string,

CITY string,

STATE string,

POSTALCODE string,

COUNTRY string,

TERRITORY string,

CONTACTLASTNAME string,

CONTACTFIRSTNAME string,

DEALSIZE string)

row format delimited

fields terminated by ','

stored as ORC;

**6. Load data from "sales\_order\_csv" into "sales\_order\_orc"**

Insert overwrite table sales\_orc select \* from sales

**Perform below menioned queries on "sales\_order\_orc" table :**

* **Calculate total sales per year**

select round(sum(sales),2) as total\_sales , year\_id from sales\_orc group by year\_id;

* **Find a product for which maximum orders were placed**

select productline, sum(quantityordered) as max\_quantity from sales\_orc group by productline order by max\_quantity desc limit 1;

* **Calculate the total sales for each quarter**

select sum(sales) as total\_Sales , qtr\_id from sales\_orc group by qtr\_id;

* **In which quarter sales was minimum**

select sum(sales) as min\_sales, qtr\_id from sales\_orc group by qtr\_id order by min\_sales desc limit 1;

* **In which country sales was maximum and in which country sales was minimum**

select country, sum(sales) as maximum\_sales from sales\_orc group by country order by maximum\_sales desc limit 1;

select country, sum(sales) as minimum\_sales from sales\_orc group by country order by minimum\_sales asc limit 1;

* **Calculate quartelry sales for each city**

select count(sales) as quarterly\_sales, city,qtr\_id from sales\_orc group by city,qtr\_id order by quarterly\_sales desc;

* **Find a month for each year in which maximum number of quantities were sold**

select month\_id, year\_id, count(quantityordered) as qty from sales\_orc where year\_id = 2003

group by year\_id order by qty desc limit 1;

select month\_id, year\_id, count(quantityordered) as qty from sales\_orc where year\_id = 2004

group by year\_id order by qty desc limit 1;

select month\_id, year\_id, count(quantityordered) as qty from sales\_orc where year\_id = 2005

group by year\_id order by qty desc limit 1;

**Scenario Based questions**

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

It actually depends on the query. I believe that if we use simple query like “select \* from table” hive reducer wont work. If you use any aggregation commands like avg,sum,max,min, I think reducer comes into the picture. However, we can still check the hive query plan (plan of execution) buy adding the command “explain” before the query. This shows the plan of execution of hive query and also show if any reducers will be used.

**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?**

There are two types of answers for this question.

* if the Hadoop was installed in single node cluster (I mean like in one laptop, like we use our systems for practice) we can access the data through hive and there won’t be any problem,
* if the hive was install in Hadoop cluster (nodes are distributed in architecture ) in this case there won’t be any data retrival, since hive uses debry to store meta and hence meta data will be configured only in one node (i.e the client querying the data won’t get the appropriate results or no data at all)

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 partition the data on month column and this allow us to faster executiones.

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

Alter table partitioned\_transaction\_details add partition (month=’december’) location (‘/locationinhdfs’)

**5) 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?**

Set hive.exec.dynamic.partition.mode=nonstrict

**6) 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?**

Create table table\_with\_serde (

Id int,

first\_name string,

last\_name string,

Email string,

gender string,

ip\_address string)

Row format serde "org.apache.hadoop.hive.serde2.opencsvserde"

WITH SERDEPROPERTIES

("separatorChar" = ",",

"quoteChar" = "\"")

Location("/tmp/data ")

Tblproperties("skip.header.line.count"=);

**7) 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?**

Create table normal\_table( id int, name string, e-mail string, country string)

Row format delimited

Fields terminated by ',';

Load data local inpath ("/localpath") into table normal\_table

Create table sequence( id int, name string, e-mail string, country string)

Row format delimited

Fields terminated by ','

Stored as squencefile;

Insert overwrite table sequence select \* from normal\_table

**8)LOAD DATA LOCAL INPATH ‘Home/country/state/’**

**OVERWRITE INTO TABLE address;**

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

it is a permission issue

**Hive Practical questions:**

Download a data from the given location -

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

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

create table customers (id int, name string,age int,address string , salary int ) row format delimited fields terminated by '|' stored as textfile;

**Create a table named orders (OID | DATE | CUSTOMER\_ID | AMOUNT)**

create table orders(order\_id int, date string,cx\_id int, amount int ) row format delimited fields terminated by '|' stored as textfile;

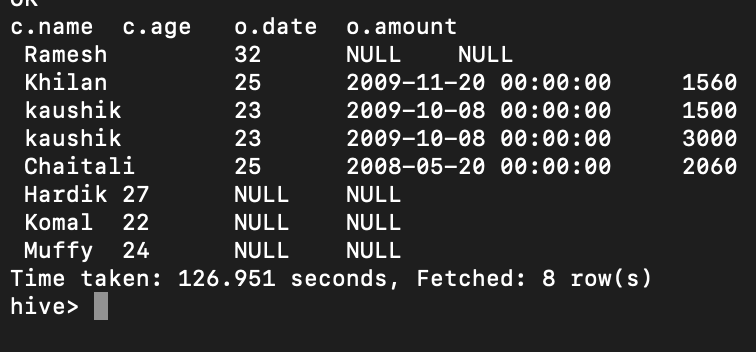
;

Now perform different joins operations on top of these tables

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

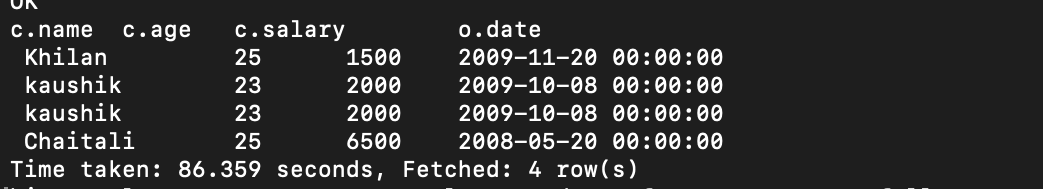
**Full outer join**

select c.name, c.age, o.date,o.amount from customers c full outer join orders o on (o.cx\_id=c.id);



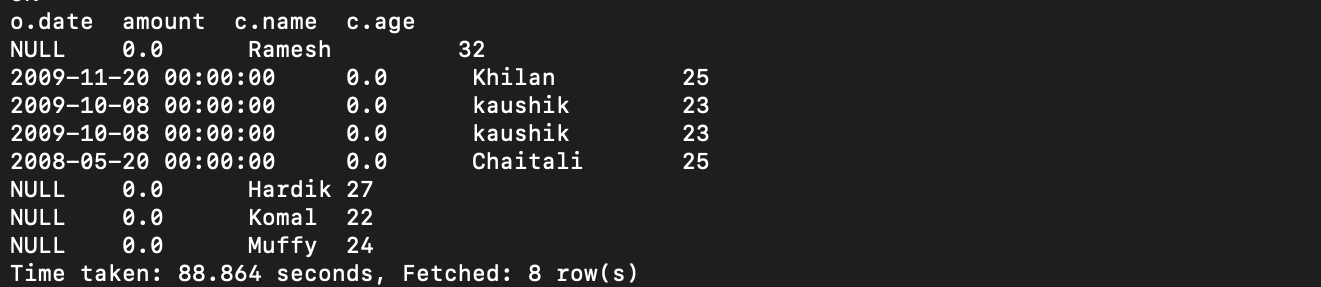
**Inner join**

select c.name, c.age,c.salary, o.date from customers c inner join orders o on( c.id = o.cx\_id);



**Right outer**

select o.date,0.amount, c.name,c.age from orders o right outer join customers c on (o.cx\_id=c.id) ;



**Left outer join**

select c.name, c.age,c.salary, o.date from customers c left outer join orders o on( c.id = o.cx\_id);

**BUILD A DATA PIPELINE WITH HIVE**

**Download a data from the given location**

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

**Create a hive table as per given schema in your dataset**

create table airquality (Date string, Time string, `CO(GT)` string,`PT08\_S1(CO)` int, `NMHC(GT)` float, `C6H6(GT)` string,`PT08\_S2(NMHC)` float,`NOx(GT)` float, `PT08\_S3(NOx)` float, `NO2(GT)` float,`PT08\_S4(NO2)` float, `PT08\_S5(O3)` float, T string, RH string, AH string )

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

WITH SERDEPROPERTIES (

"separatorChar" = "\u0059", # “uoo59” is semicolumn delimeter

"quoteChar" = "\"")

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

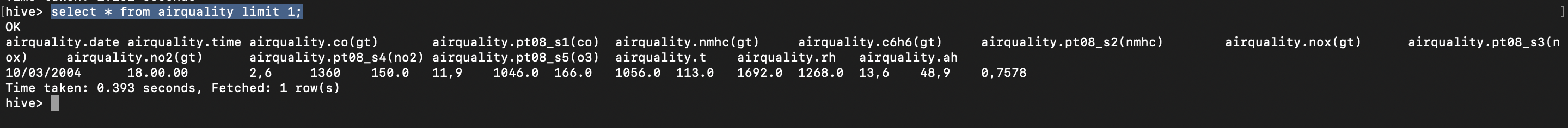
**note**: hive doesn’t allow users to query from the column like “PT08.S1(CO) ”.

in documentation of hive (hive version 0.13) it was written that it supports Unicode characters, but we are using hive (Hive 1.1.0-cdh5.7.0) and when we query column containing the dot in the column name (i.e “PT08.S1(CO) ”.) it throws error, due to version dependency. Therefore to overcome this problem we have replaced the column (.) with (\_) (i.e., ` PT08.S1(CO)` => ` PT08\_S1(CO)`)

**2. try to place a data into table location**

hadoop fs -copyFromLocal 'file:///AirQualityUCI.csv' /user/hive/warehouse/airquality/

**3.Perform a select operation .**

select \* from airquality limit 1;

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

insert overwrite local directory '/tmp/results.csv' row format delimited fields terminated by ',' select \* from airquality limit 2;

**5. Perform group by operation .**

select sum(`pt08\_s1(co)`), `co(gt)` from airquality group by `co(gt)`;

**7. Perform filter operation at least 5 kinds of filter examples**

select \* from air\_quality where CO<1;

select distinct time from air\_quality where C6H6 < 2

and dt='11-03-2004';

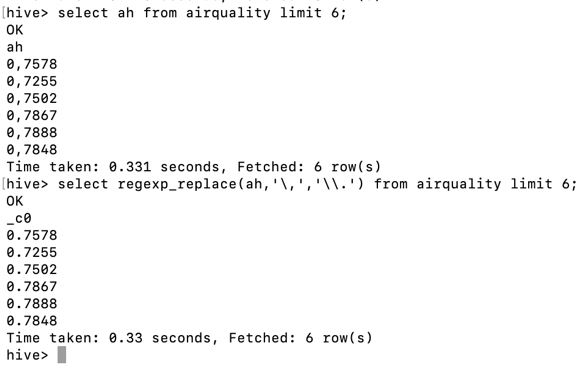
select \* from air\_quality where nox>10;

select \* from air\_quality where no2>20;

select \* from air\_quality where PT08\_S1>15;

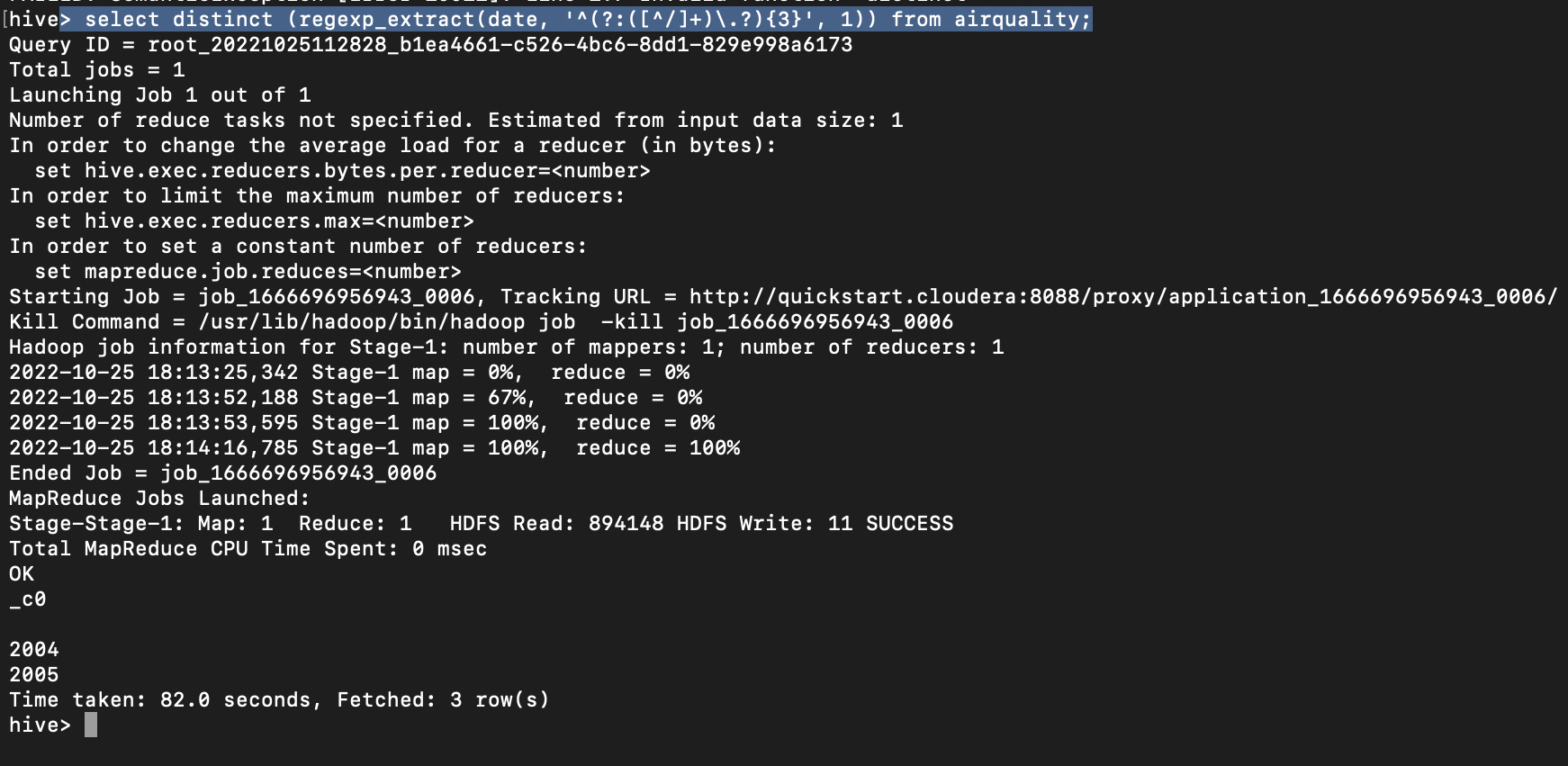
**8. show and example of regex operation**

select regexp\_replace(ah,'\,','\\.') from airquality limit 6;

****

**15 . distinct operation you have to perform .**

select distinct (regexp\_extract(date, '^(?:([^/]+)\.?){3}', 1)) from airquality;



The above command shows the distinct years of date column which was extracted by regex\_extract.

**9. alter table operation**

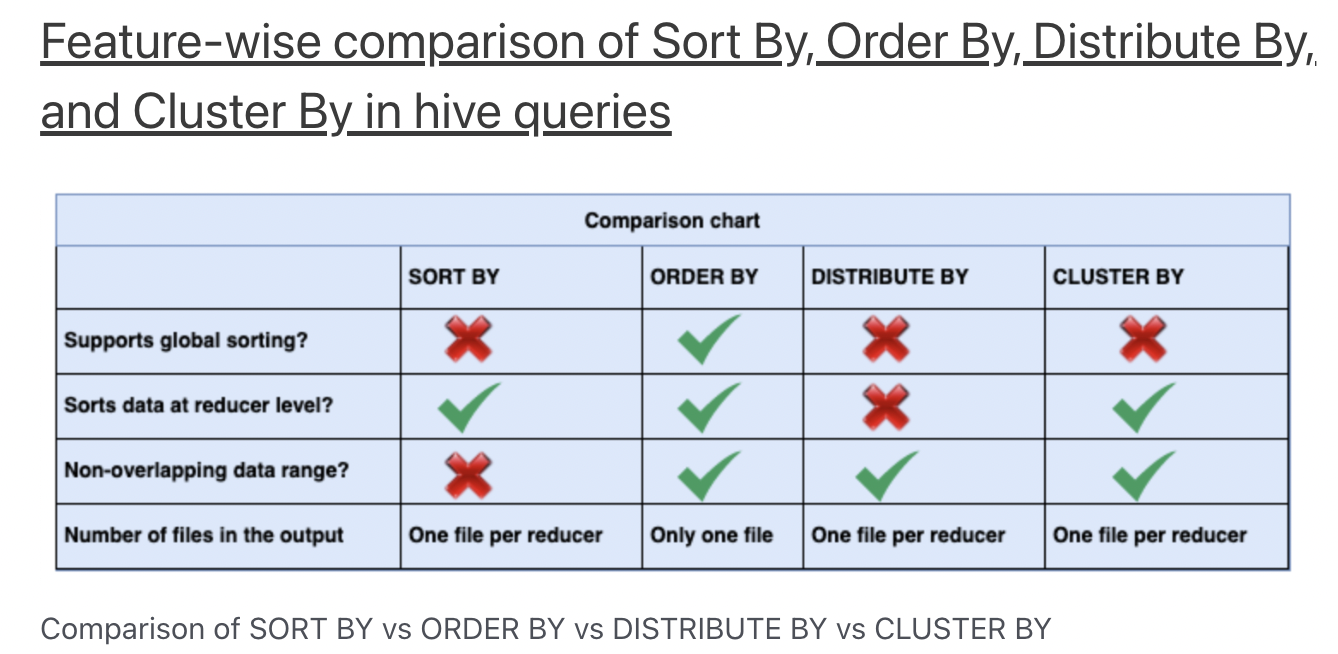
alter table airquality change time timestamp string;

**10 . drop table operation**

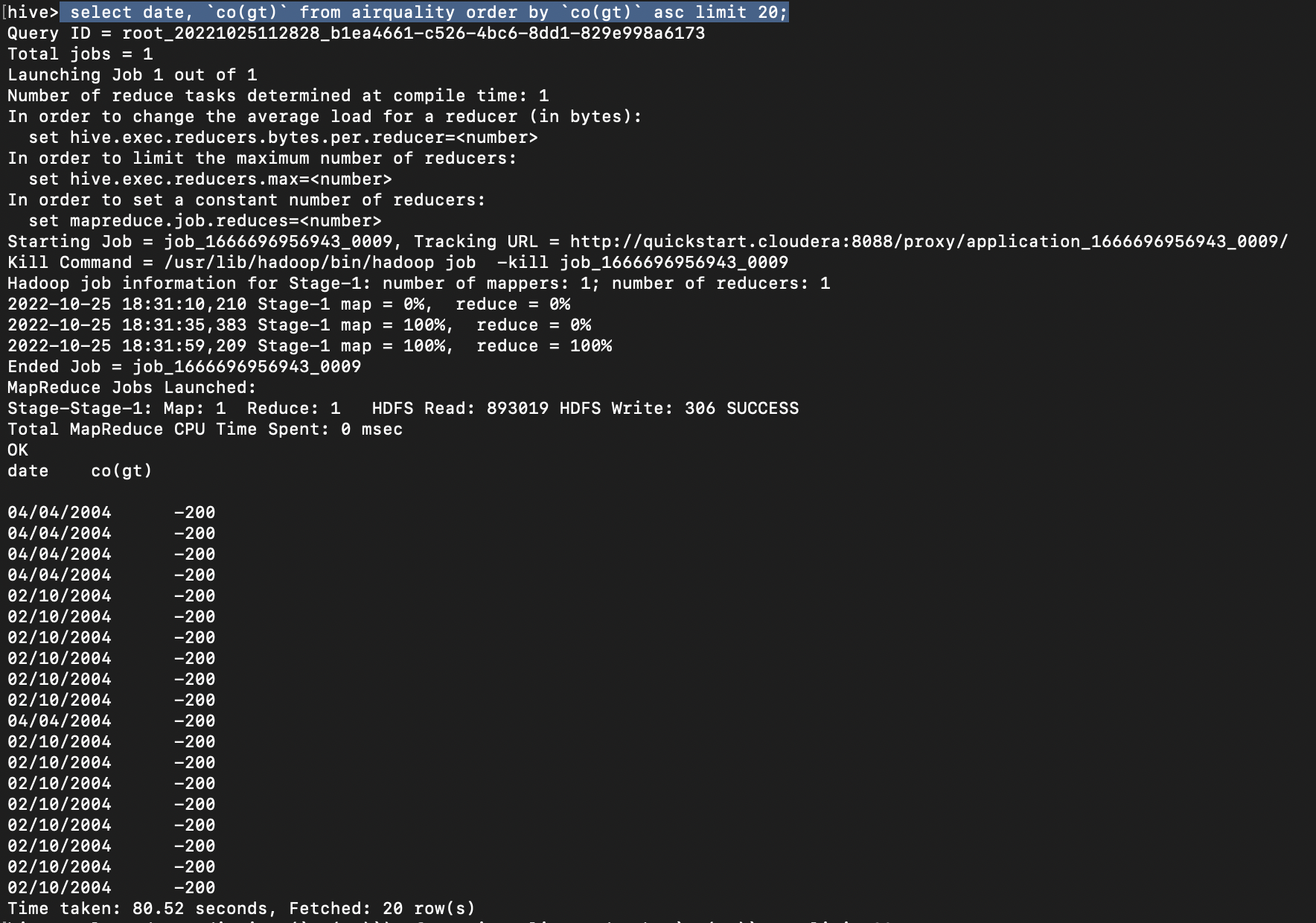
Drop table airquality;

**11.order by operation .**

**12.sorting operation you have to perform**



Note: since we are local cluster I’m doing order by, but if we are on Hadoop cluster we need to select the operation based on data

select date,`co(gt)` from airquality order by `co(gt)` asc limit 20;

**17 . union operation you have to perform .**

select date from airquality union all select date from airquality;

**18. table view operation you have to perform .**

create view if not exists air as select \* from airquality limit 1;

**19 . like an operation you have to perform .**

select `co(gt)` from airquality where `co(gt)` like '-20%';