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

for any select query that is executed on Hive which does not include group by, joins, aggregate functions, or complex constraints then reducer is not called.

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.

So, if multiple clients try to access the metastore at the same time, they will get an error.

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?

this is problem of query latency. so, we can solve this problem of query latency by partitioning the table according to each month.

so we will be scanning only the partitioned data instead of whole data sets.

as we can not directly create partition on existing table we need to create

partition table.

1. create a partioned table .

```
hive> create table transaction_details_partitioned (
cust_id int,
amount float)
partitioned by(month string,country string)
row format delimited fileds terminated by ',';
```

2. to enable dynamic partition need to set few commands in hive shell

```
hive> set hive.exec.dynamic.partition = true;

hive> set hive.exec.dynamic.partition.mode = nonstrict;
```

3. now load data in fact transfer data from non partitioned table to newly created partition table

```
hive> Insert overwrite table transaction_details_partitioned  partition(month,country) select cust_id,amount,month,country from transaction_details;
```

4. now later we can drop even old table and new table can be renamed to old table name

```
hive> alter table transaction_details_partitioned  rename to transaction_details;
```

5. now we can perform query using each partition and query processing time will be reduce and performance is improve

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

```
alter table transaction_details_partitioned add partition(month = 'Dec') Location ‘user/hive/warehouse/transaction_details_partitioned/’
```

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 = true;

set hive.exec.dynamic.partition.mode = nonstrict;
```

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 external table sample_csv
(
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';
```

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 a temporary table:

```
create table temp_tbl(
ind int,
name string,
e_mail string,
country string)
row format delimited
fields terminated by ','
```

```
stroed as textfile;
```

load data into these file using below command :

```
load data inpath '/input_directory/...' into table temp_tbl;
```

create a table that will store data in sequenceFile format:

```
create table seq_tbl(
ind int,
name string,
e_mail string,
country string)
row format delimited
fields terminated by ','
stroed as sequencefile;
```

transfer data from temporary table into this table:

```
Insert overwrite table seq_tbl select * from temp_tbl;
```

LOAD DATA LOCAL INPATH ‘Home/country/state/’ OVERWRITE INTO TABLE address;
The following statement failed to execute. What can be the cause?

File is missing in local inpath

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:

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

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

Step 3: Add ssh public_rsa id key to the authorized keys file

Step 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:

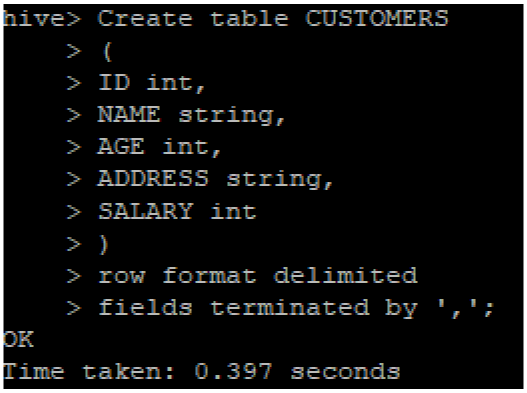
```
ssh -X hadoop@192.168.1.103
```

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

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

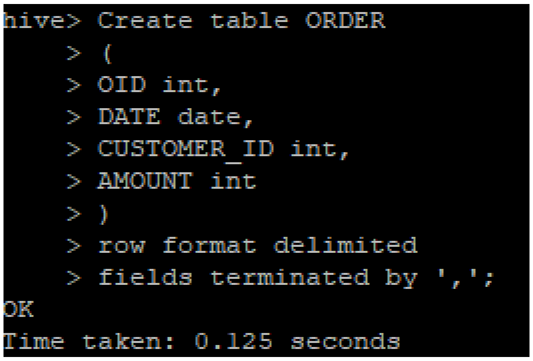
Step 8: Check the output of the jps command on the new node

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



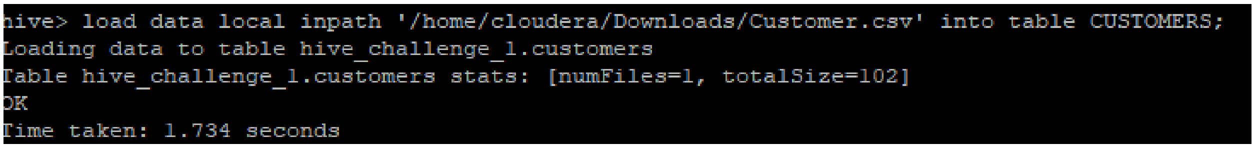
```
hive> Create table CUSTOMERS
    > (
    > ID int,
    > NAME string,
    > AGE int,
    > ADDRESS string,
    > SALARY int
    > )
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.397 seconds
```

Create a Second table ORDER(OID | DATE | CUSTOMER_ID | AMOUNT)



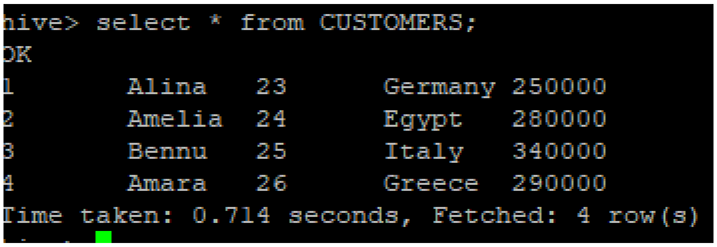
```
hive> Create table ORDER
    > (
    > OID int,
    > DATE date,
    > CUSTOMER_ID int,
    > AMOUNT int
    > )
    > row format delimited
    > fields terminated by ',';
OK
Time taken: 0.125 seconds
```

Loading data into Customers table:



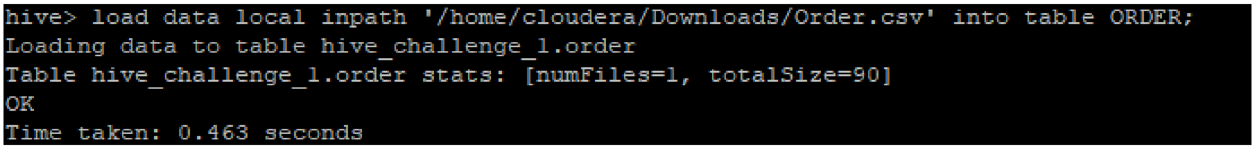
```
hive> load data local inpath '/home/cloudera/Downloads/Customer.csv' into table CUSTOMERS;
Loading data to table hive_challenge_1.customers
Table hive_challenge_1.customers stats: [numFiles=1, totalSize=102]
OK
Time taken: 1.734 seconds
```

Verifying data:



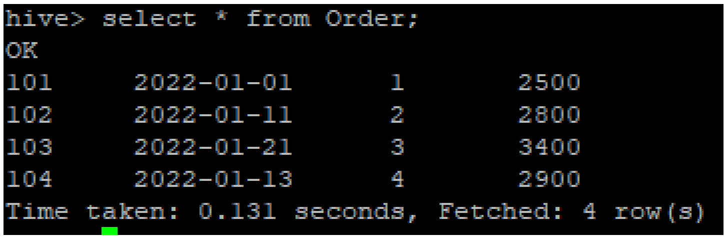
```
hive> select * from CUSTOMERS;
OK
1       Alina   23      Germany 250000
2       Amelia  24      Egypt   280000
3       Bennu   25      Italy   340000
4       Amara   26      Greece  290000
Time taken: 0.714 seconds, Fetched: 4 row(s)
```

Loading data into Order table:



```
hive> load data local inpath '/home/cloudera/Downloads/Order.csv' into table ORDER;
Loading data to table hive_challenge_1.order
Table hive_challenge_1.order stats: [numFiles=1, totalSize=90]
OK
Time taken: 0.463 seconds
```

Verifying data:



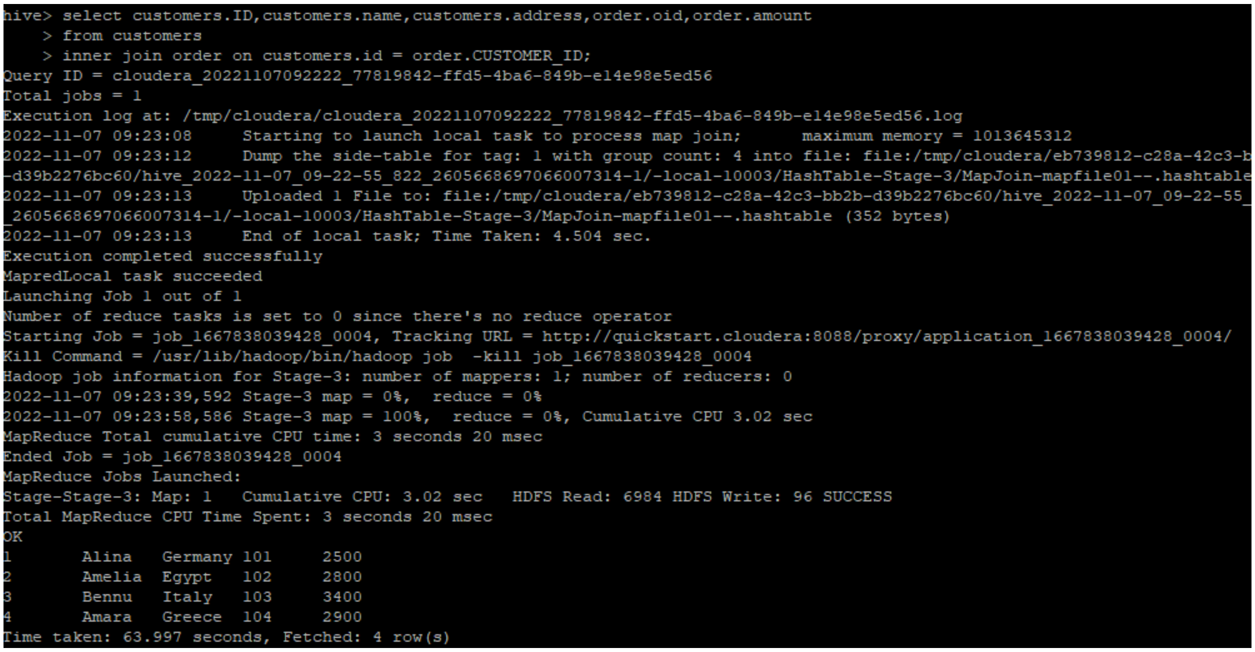
```
hive> select * from Order;
OK
101     2022-01-01      1       2500
102     2022-01-11      2       2800
103     2022-01-21      3       3400
104     2022-01-13      4       2900
Time taken: 0.131 seconds, Fetched: 4 row(s)
```

Inner join:

select customers.ID,customers.name,customers.address,order.oid,order.amount

from customers

inner join order on customers.id = order.CUSTOMER_ID;



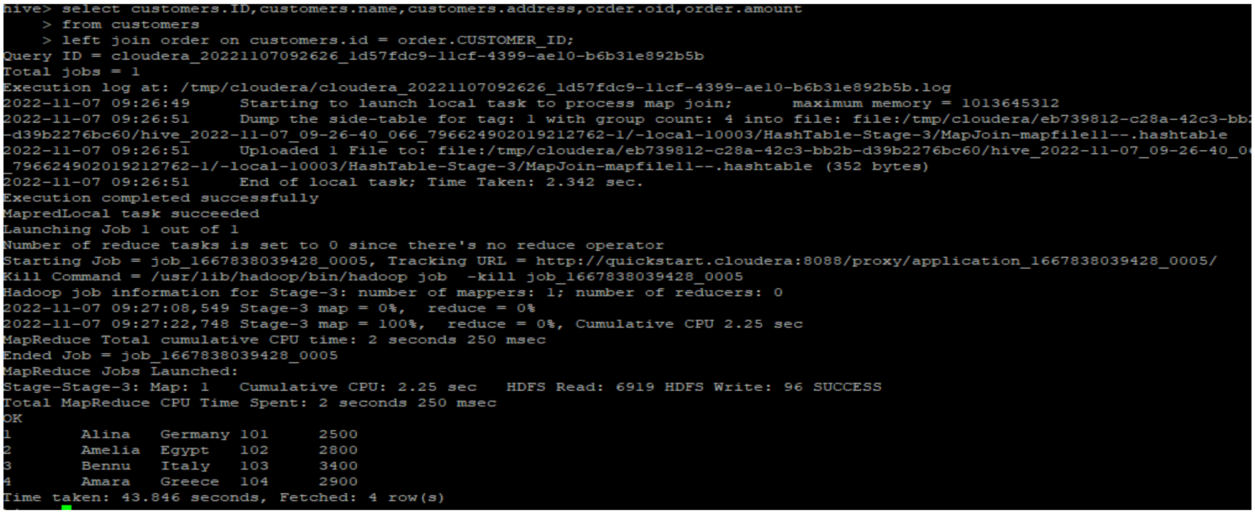
```
hive> select customers.ID,customers.name,customers.address,order.oid,order.amount
    > from customers
    > inner join order on customers.id = order.CUSTOMER_ID;
Query ID = cloudera_20221107092222_77819842-ffd5-4ba6-849b-e14e98e5ed56
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20221107092222_77819842-ffd5-4ba6-849b-e14e98e5ed56.log
2022-11-07 09:23:08     Starting to launch local task to process map join;      maximum memory = 1013645312
2022-11-07 09:23:12     Dump the side-table for tag: 1 with group count: 4 into file: file:/tmp/cloudera/eb739812-c28a-42c3-b
-d39b2276bc60/hive_2022-11-07_09-22-55_822_2605668697066007314-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2022-11-07 09:23:13     Uploaded 1 File to: file:/tmp/cloudera/eb739812-c28a-42c3-bb2b-d39b2276bc60/hive_2022-11-07_09-22-55_
_2605668697066007314-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile01--.hashtable (352 bytes)
2022-11-07 09:23:13     End of local task; Time Taken: 4.504 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1667838039428_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1667838039428_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1667838039428_0004
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2022-11-07 09:23:39,592 Stage-3 map = 0%,  reduce = 0%
2022-11-07 09:23:58,586 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 3.02 sec
MapReduce Total cumulative CPU time: 3 seconds 20 msec
Ended Job = job_1667838039428_0004
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 3.02 sec   HDFS Read: 6984 HDFS Write: 96 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 20 msec
OK
1       Alina   Germany 101     2500
2       Amelia  Egypt   102     2800
3       Bennu   Italy   103     3400
4       Amara   Greece  104     2900
Time taken: 63.997 seconds, Fetched: 4 row(s)
```

Left join:

select customers.ID,customers.name,customers.address,order.oid,order.amount

from customers

left join order on customers.id = order.CUSTOMER_ID;



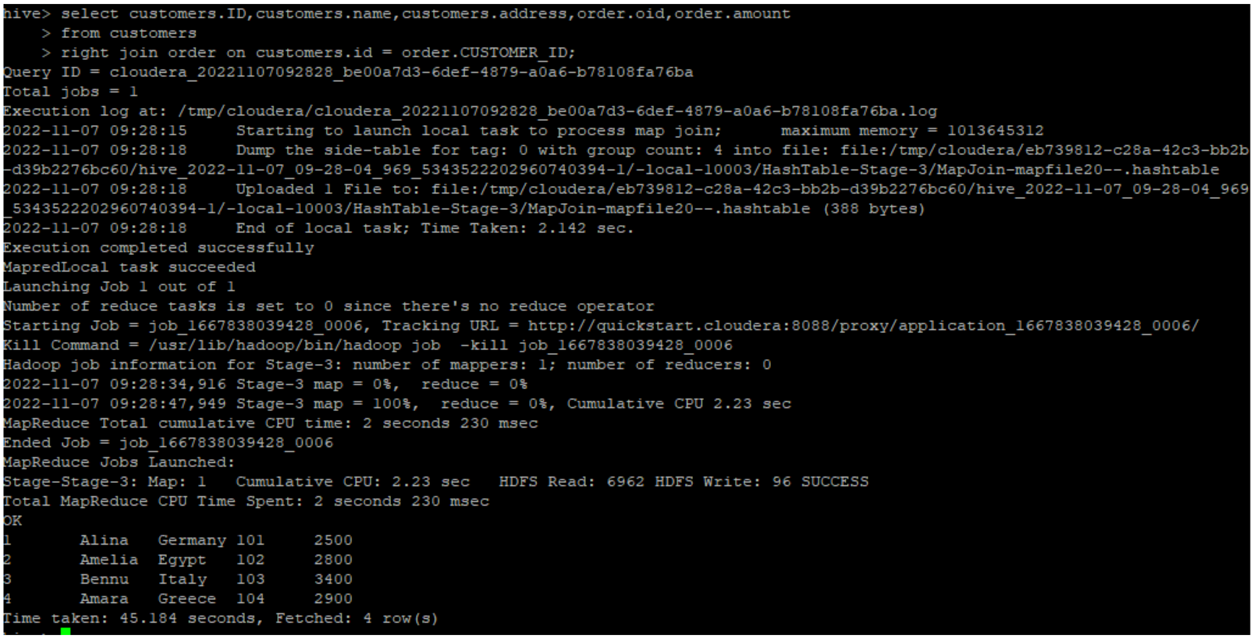
```
hive> select customers.ID,customers.name,customers.address,order.oid,order.amount
    > from customers
    > left join order on customers.id = order.CUSTOMER_ID;
Query ID = cloudera_20221107092626_1d57fdc9-11cf-4399-ae10-b6b31e892b5b
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20221107092626_1d57fdc9-11cf-4399-ae10-b6b31e892b5b.log
2022-11-07 09:26:49     Starting to launch local task to process map join;      maximum memory = 1013645312
2022-11-07 09:26:51     Dump the side-table for tag: 1 with group count: 4 into file: file:/tmp/cloudera/eb739812-c28a-42c3-bb2
-d39b2276bc60/hive_2022-11-07_09-26-40_066_796624902019212762-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile11--.hashtable
2022-11-07 09:26:51     Uploaded 1 File to: file:/tmp/cloudera/eb739812-c28a-42c3-bb2b-d39b2276bc60/hive_2022-11-07_09-26-40_0
_796624902019212762-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile11--.hashtable (352 bytes)
2022-11-07 09:26:51     End of local task; Time Taken: 2.342 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1667838039428_0005, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1667838039428_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1667838039428_0005
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2022-11-07 09:27:08,549 Stage-3 map = 0%,  reduce = 0%
2022-11-07 09:27:22,748 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 2.25 sec
MapReduce Total cumulative CPU time: 2 seconds 250 msec
Ended Job = job_1667838039428_0005
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 2.25 sec   HDFS Read: 6919 HDFS Write: 96 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 250 msec
OK
1       Alina   Germany 101     2500
2       Amelia  Egypt   102     2800
3       Bennu   Italy   103     3400
4       Amara   Greece  104     2900
Time taken: 43.846 seconds, Fetched: 4 row(s)
```

Right join:

select customers.ID,customers.name,customers.address,order.oid,order.amount

from customers

right join order on customers.id = order.CUSTOMER_ID;



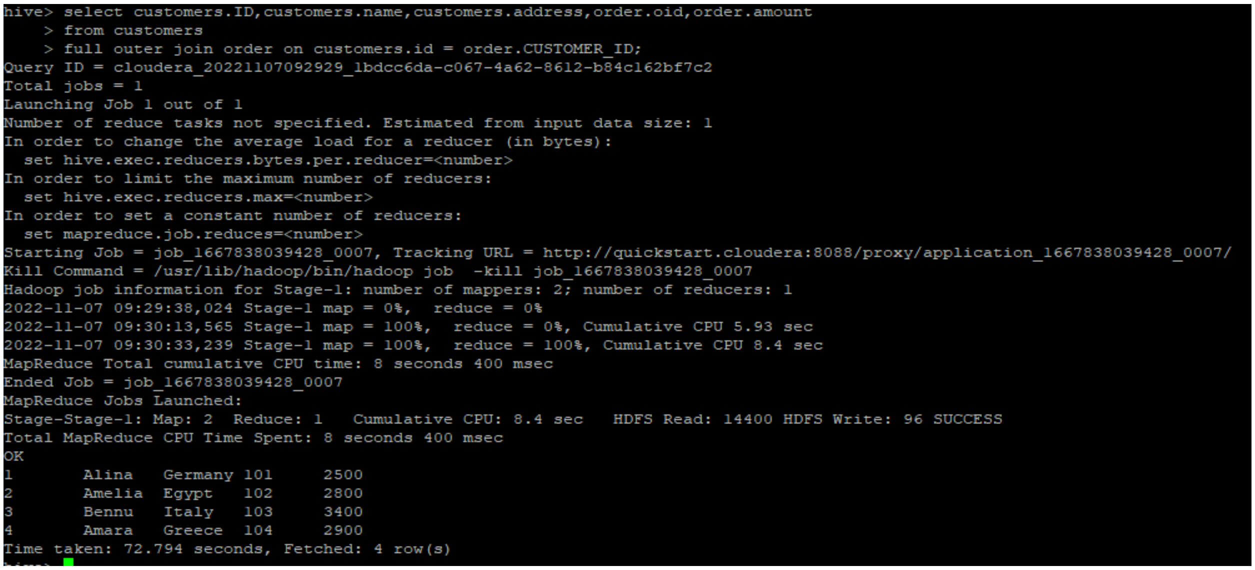
```
hive> select customers.ID,customers.name,customers.address,order.oid,order.amount
    > from customers
    > right join order on customers.id = order.CUSTOMER_ID;
Query ID = cloudera_20221107092828_be00a7d3-6def-4879-a0a6-b78108fa76ba
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20221107092828_be00a7d3-6def-4879-a0a6-b78108fa76ba.log
2022-11-07 09:28:15     Starting to launch local task to process map join;      maximum memory = 1013645312
2022-11-07 09:28:18     Dump the side-table for tag: 0 with group count: 4 into file: file:/tmp/cloudera/eb739812-c28a-42c3-bb2b
-d39b2276bc60/hive_2022-11-07_09-28-04_969_5343522202960740394-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile20--.hashtable
2022-11-07 09:28:18     Uploaded 1 File to: file:/tmp/cloudera/eb739812-c28a-42c3-bb2b-d39b2276bc60/hive_2022-11-07_09-28-04_969
_5343522202960740394-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile20--.hashtable (388 bytes)
2022-11-07 09:28:18     End of local task; Time Taken: 2.142 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1667838039428_0006, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1667838039428_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1667838039428_0006
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0
2022-11-07 09:28:34,916 Stage-3 map = 0%,  reduce = 0%
2022-11-07 09:28:47,949 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 2.23 sec
MapReduce Total cumulative CPU time: 2 seconds 230 msec
Ended Job = job_1667838039428_0006
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1   Cumulative CPU: 2.23 sec   HDFS Read: 6962 HDFS Write: 96 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 230 msec
OK
1       Alina   Germany 101     2500
2       Amelia  Egypt   102     2800
3       Bennu   Italy   103     3400
4       Amara   Greece  104     2900
Time taken: 45.184 seconds, Fetched: 4 row(s)
```

Full outer join:

select customers.ID,customers.name,customers.address,order.oid,order.amount

from customers

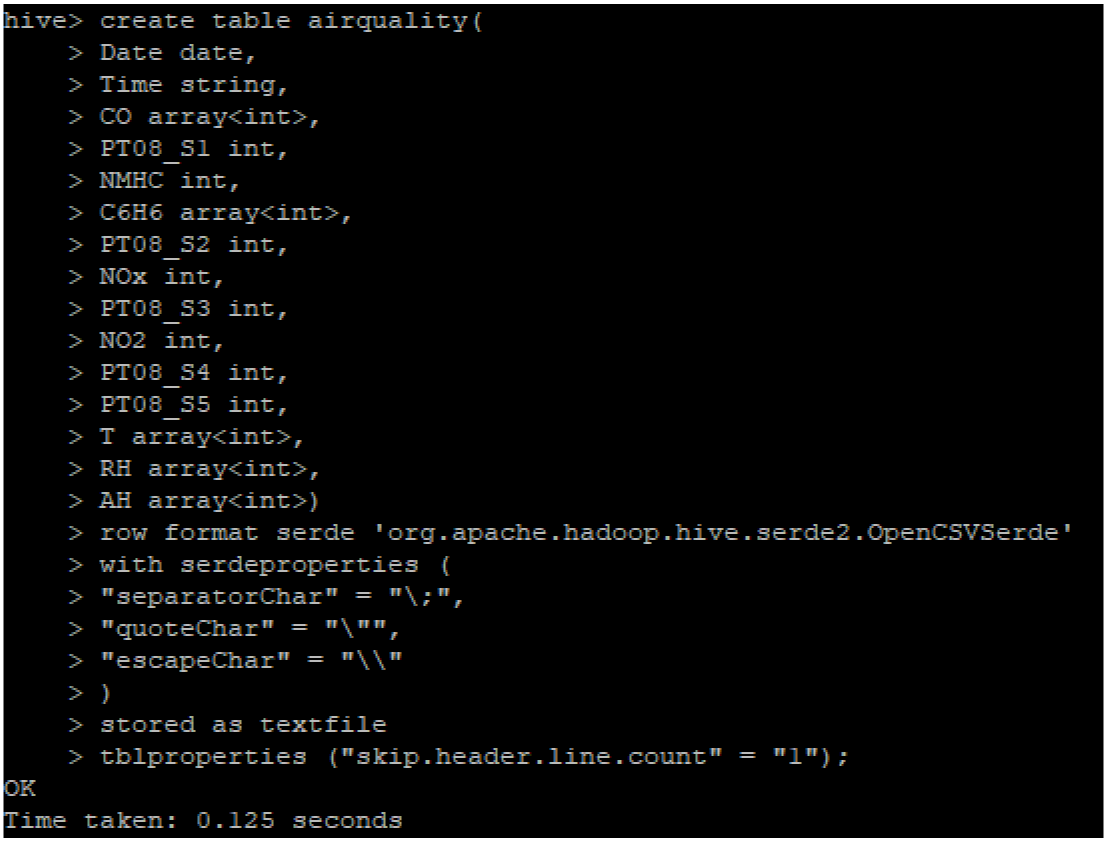
full outer join order on customers.id = order.CUSTOMER_ID;



```
hive> select customers.ID,customers.name,customers.address,order.oid,order.amount
    > from customers
    > full outer join order on customers.id = order.CUSTOMER_ID;
Query ID = cloudera_20221107092929_1bdcc6da-c067-4a62-8612-b84c162bf7c2
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1667838039428_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1667838039428_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1667838039428_0007
Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1
2022-11-07 09:29:38,024 Stage-1 map = 0%,  reduce = 0%
2022-11-07 09:30:13,565 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 5.93 sec
2022-11-07 09:30:33,239 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 8.4 sec
MapReduce Total cumulative CPU time: 8 seconds 400 msec
Ended Job = job_1667838039428_0007
MapReduce Jobs Launched:
Stage-Stage-1: Map: 2  Reduce: 1   Cumulative CPU: 8.4 sec   HDFS Read: 14400 HDFS Write: 96 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 400 msec
OK
1       Alina   Germany 101     2500
2       Amelia  Egypt   102     2800
3       Bennu   Italy   103     3400
4       Amara   Greece  104     2900
Time taken: 72.794 seconds, Fetched: 4 row(s)
```

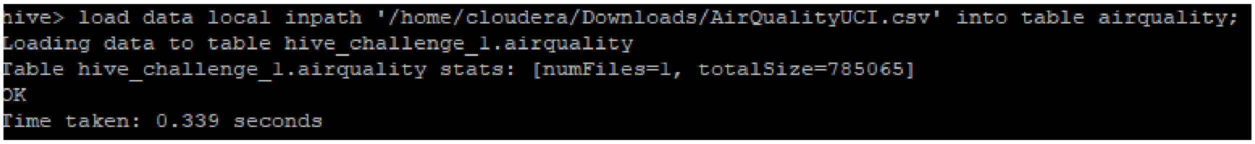
Create a hive table as per given schema in your dataset

```
create table airquality(
Date date,
Time string,
CO array<int>,
PT08_S1 int,
NMHC int,
C6H6 array<int>,
PT08_S2 int,
NOx int,
PT08_S3 int,
NO2 int,
PT08_S4 int,
PT08_S5 int,
T array<int>,
RH array<int>,
AH array<int>)
row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
with serdeproperties (
"separatorChar" = "\;",
"quoteChar" = "\"",
"escapeChar" = "\\"
)
stored as textfile
tblproperties ("skip.header.line.count" = "1");
```



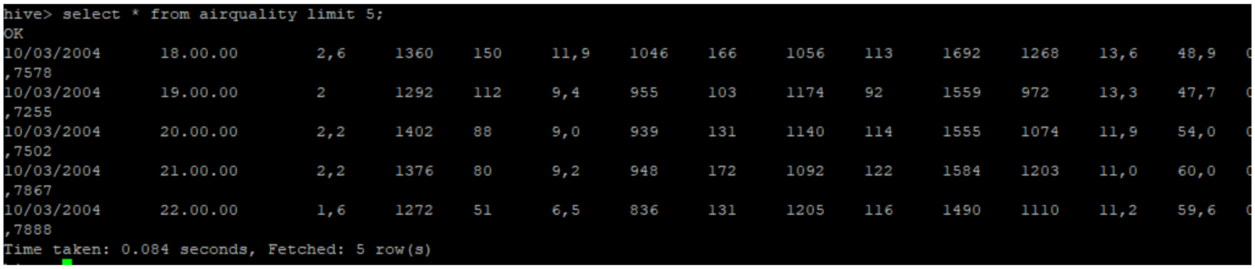
```
hive> create table airquality(
    > Date date,
    > Time string,
    > CO array<int>,
    > PT08_S1 int,
    > NMHC int,
    > C6H6 array<int>,
    > PT08_S2 int,
    > NOx int,
    > PT08_S3 int,
    > NO2 int,
    > PT08_S4 int,
    > PT08_S5 int,
    > T array<int>,
    > RH array<int>,
    > AH array<int>)
    > row format serde 'org.apache.hadoop.hive.serde2.OpenCSVSerde'
    > with serdeproperties (
    > "separatorChar" = "\;",
    > "quoteChar" = "\"",
    > "escapeChar" = "\\"
    > )
    > stored as textfile
    > tblproperties ("skip.header.line.count" = "1");
OK
Time taken: 0.125 seconds
```

try to place a data into table location:



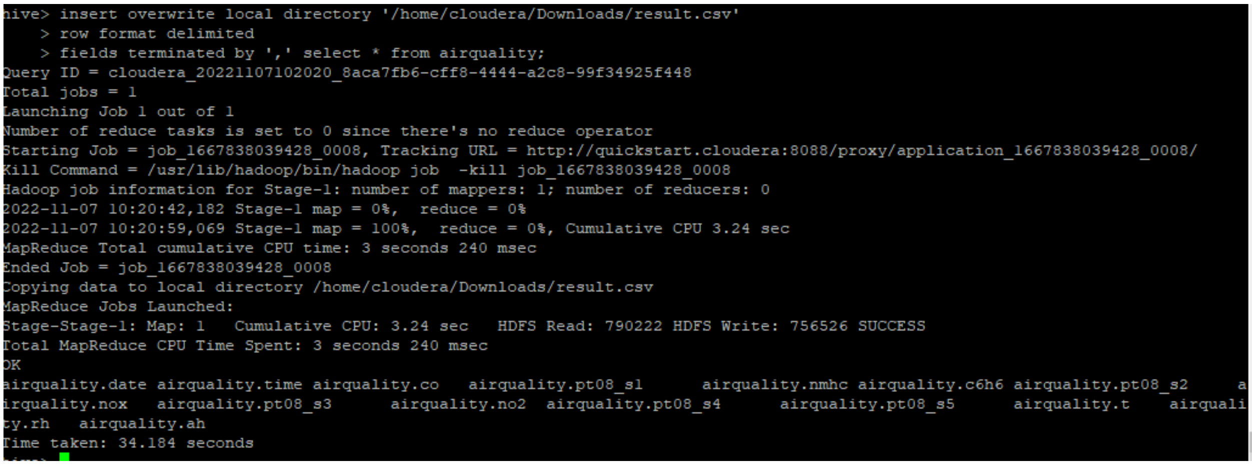
```
hive> load data local inpath '/home/cloudera/Downloads/AirQualityUCI.csv' into table airquality;
Loading data to table hive_challenge_1.airquality
Table hive_challenge_1.airquality stats: [numFiles=1, totalSize=785065]
OK
Time taken: 0.339 seconds
```

Perform a select operation:



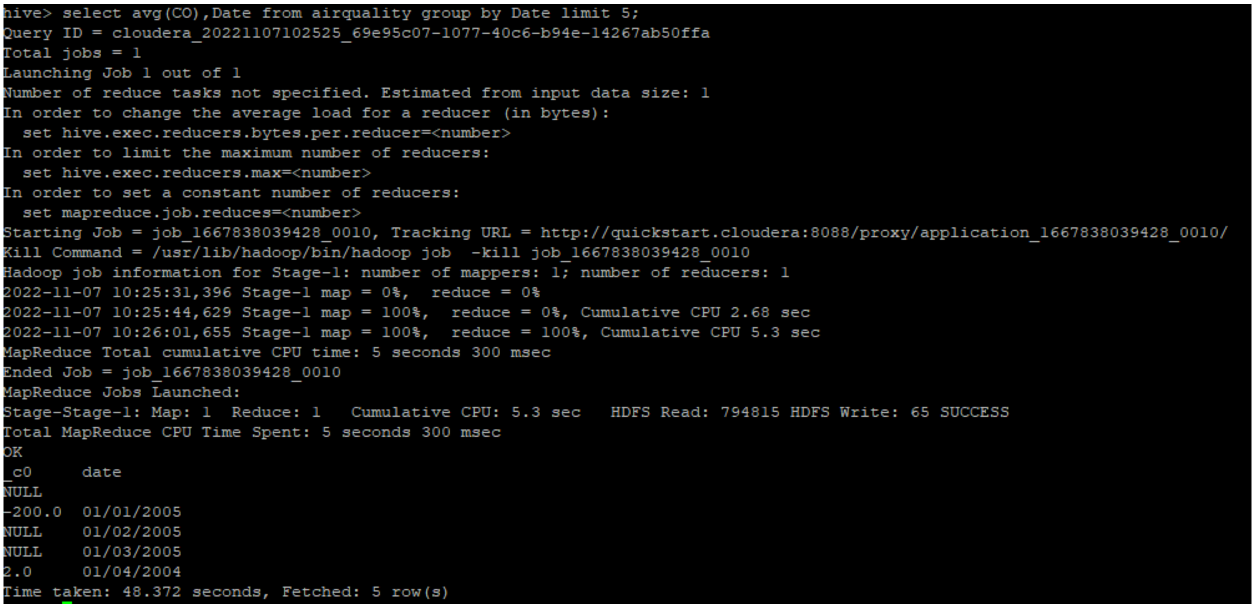
```
hive> select * from airquality limit 5;
OK
10/03/2004      18.00.00        2,6     1360    150     11,9    1046    166     1056    113     1692    1268    13,6    48,9    0
,7578
10/03/2004      19.00.00        2       1292    112     9,4     955     103     1174    92      1559    972     13,3    47,7    0
,7255
10/03/2004      20.00.00        2,2     1402    88      9,0     939     131     1140    114     1555    1074    11,9    54,0    0
,7502
10/03/2004      21.00.00        2,2     1376    80      9,2     948     172     1092    122     1584    1203    11,0    60,0    0
,7867
10/03/2004      22.00.00        1,6     1272    51      6,5     836     131     1205    116     1490    1110    11,2    59,6    0
,7888
Time taken: 0.084 seconds, Fetched: 5 row(s)
```

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



```
hive> insert overwrite local directory '/home/cloudera/Downloads/result.csv'
    > row format delimited
    > fields terminated by ',' select * from airquality;
Query ID = cloudera_20221107102020_8aca7fb6-cff8-4444-a2c8-99f34925f448
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1667838039428_0008, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1667838039428_0008/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1667838039428_0008
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2022-11-07 10:20:42,182 Stage-1 map = 0%,  reduce = 0%
2022-11-07 10:20:59,069 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 3.24 sec
MapReduce Total cumulative CPU time: 3 seconds 240 msec
Ended Job = job_1667838039428_0008
Copying data to local directory /home/cloudera/Downloads/result.csv
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1   Cumulative CPU: 3.24 sec   HDFS Read: 790222 HDFS Write: 756526 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 240 msec
OK
airquality.date airquality.time airquality.co   airquality.pt08_s1      airquality.nmhc airquality.c6h6 airquality.pt08_s2     a
irquality.nox   airquality.pt08_s3      airquality.no2  airquality.pt08_s4      airquality.pt08_s5      airquality.t    airquali
ty.rh   airquality.ah
Time taken: 34.184 seconds
```

Perform group by operation



```
hive> select avg(CO),Date from airquality group by Date limit 5;
Query ID = cloudera_20221107102525_69e95c07-1077-40c6-b94e-14267ab50ffa
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1667838039428_0010, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1667838039428_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1667838039428_0010
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2022-11-07 10:25:31,396 Stage-1 map = 0%,  reduce = 0%
2022-11-07 10:25:44,629 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.68 sec
2022-11-07 10:26:01,655 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU 5.3 sec
MapReduce Total cumulative CPU time: 5 seconds 300 msec
Ended Job = job_1667838039428_0010
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1  Reduce: 1   Cumulative CPU: 5.3 sec   HDFS Read: 794815 HDFS Write: 65 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 300 msec
OK
_c0     date
NULL
-200.0  01/01/2005
NULL    01/02/2005
NULL    01/03/2005
2.0     01/04/2004
Time taken: 48.372 seconds, Fetched: 5 row(s)
```

Perform filter operation at least 5 kinds of filter examples .

select count(*),date from airquality group by date;

select count(*) total,date from airquality group by date having total< 24;

select * from airquality where date = '31/12/2004';

select * from airquality where AH <0.8393;

select avg(cast(NMHC as int)) from airquality;

show and example of regex operation

alter table operation

rename table:

```
hive> alter table airquality rename to Air_quality2;
```

Add column:

```
alter table airquality add columns(humidity int);
```

change column name:

```
alter table airquality change humidity humid int;
```

change column datatype:

```
alter table airquality change humidity humid string;
```

drop table operation

```
drop table airquality;
```

order by operation

```
select * from airquality order by T;
```

where clause operations you have to perform

```
select CO from airquality where Date=` 11-03-2004`
```

sorting operation you have to perform

```
select *  from airquality where date = '31/12/2004' sort by t;
```

distinct operation you have to perform .

```
select distinct date from airquality;
```

like an operation you have to perform

```
select T from airquality  where time like '18%;
```

union operation you have to perform

```
select * from airquality where date = '30/03/2005'
```

```
union all

select * from airquality where date= '31/03/2005';
```

table view operation you have to perform

```
create view air_qual_viw as select * from airquality where date = '31/03/2005';
```