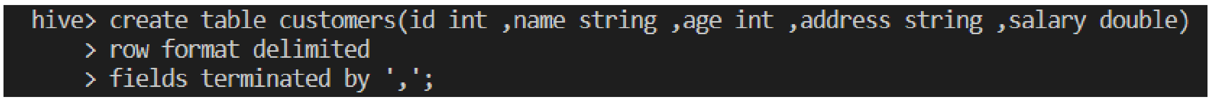
# Hive Assignment-2

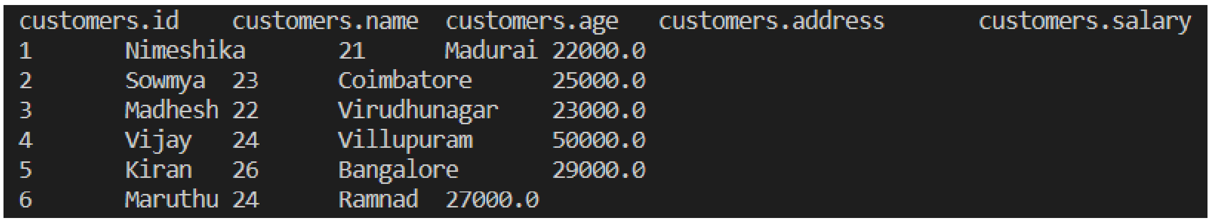
## Hive Practical questions:

## Hive Join operations

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

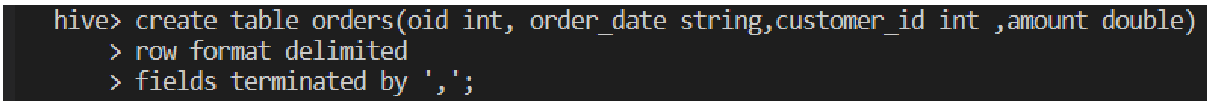


```
hive> create table customers(id int ,name string ,age int ,address string ,salary double)
    > row format delimited
    > fields terminated by ',';
```

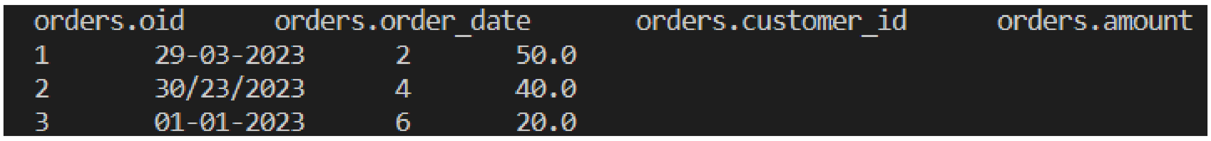


```
customers.id    customers.name  customers.age   customers.address       customers.salary
1       Nimeshika       21      Madurai 22000.0
2       Sowmya  23      Coimbatore      25000.0
3       Madhesh 22      Virudhunagar    23000.0
4       Vijay   24      Villupuram      50000.0
5       Kiran   26      Bangalore       29000.0
6       Maruthu 24      Ramnad  27000.0
```

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



```
hive> create table orders(oid int, order_date string,customer_id int ,amount double)
    > row format delimited
    > fields terminated by ',';
```

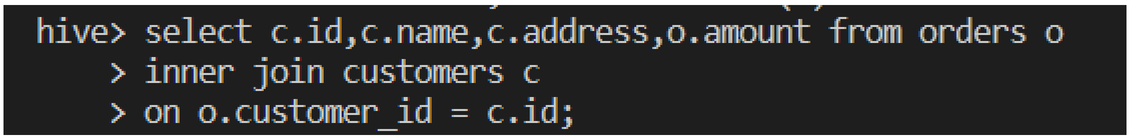


```
orders.oid      orders.order_date       orders.customer_id      orders.amount
1       29-03-2023      2       50.0
2       30/23/2023      4       40.0
3       01-01-2023      6       20.0
```

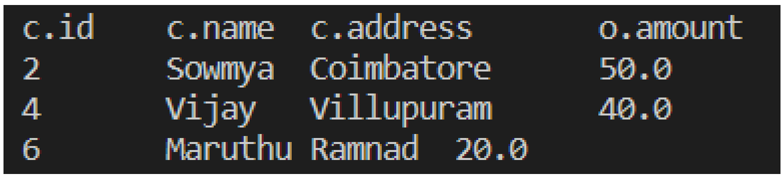
Now perform different joins operations on top of these tables

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

### Inner Join

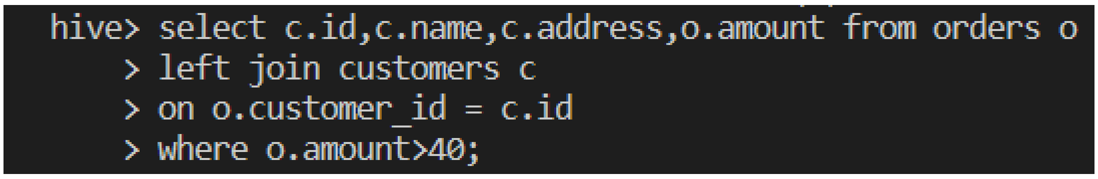


```
hive> select c.id,c.name,c.address,o.amount from orders o
    > inner join customers c
    > on o.customer_id = c.id;
```

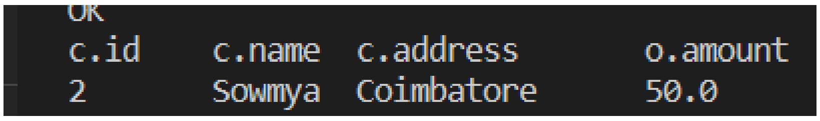


```
c.id    c.name  c.address       o.amount
2       Sowmya  Coimbatore      50.0
4       Vijay   Villupuram      40.0
6       Maruthu Ramnad  20.0
```

## Left Join

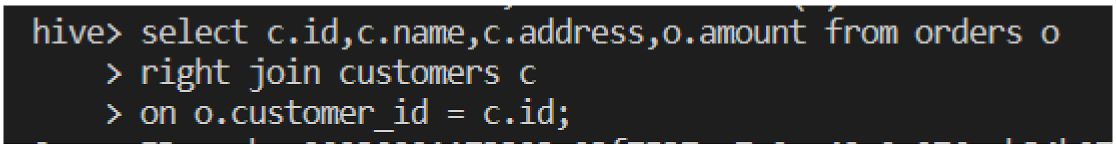


```
hive> select c.id,c.name,c.address,o.amount from orders o
    > left join customers c
    > on o.customer_id = c.id
    > where o.amount>40;
```

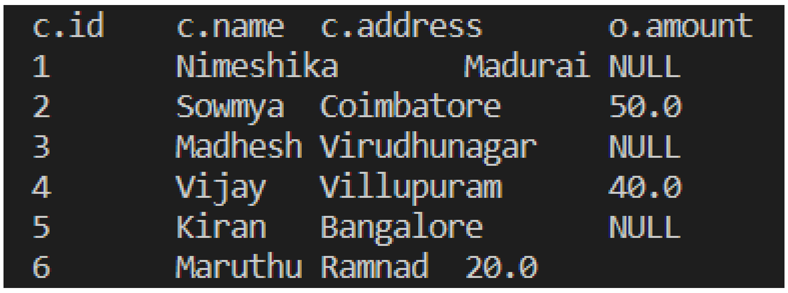


```
OK
c.id    c.name  c.address       o.amount
2       Sowmya  Coimbatore      50.0
```

## Right Join

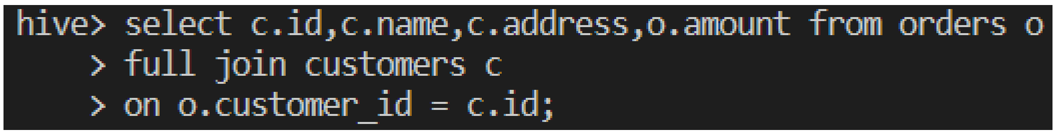


```
hive> select c.id,c.name,c.address,o.amount from orders o
    > right join customers c
    > on o.customer_id = c.id;
```

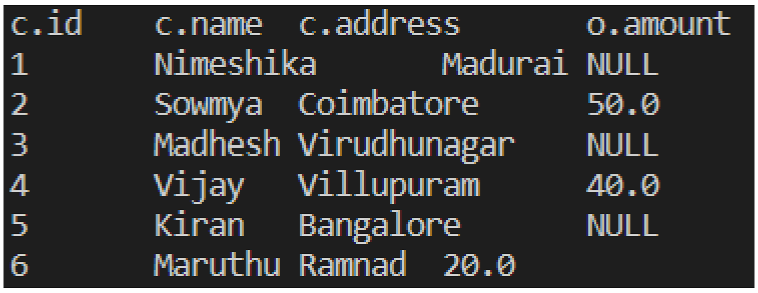


```
c.id    c.name  c.address       o.amount
1       Nimeshika       Madurai NULL
2       Sowmya  Coimbatore      50.0
3       Madhesh Virudhunagar    NULL
4       Vijay   Villupuram      40.0
5       Kiran   Bangalore       NULL
6       Maruthu Ramnad  20.0
```

## Full Join



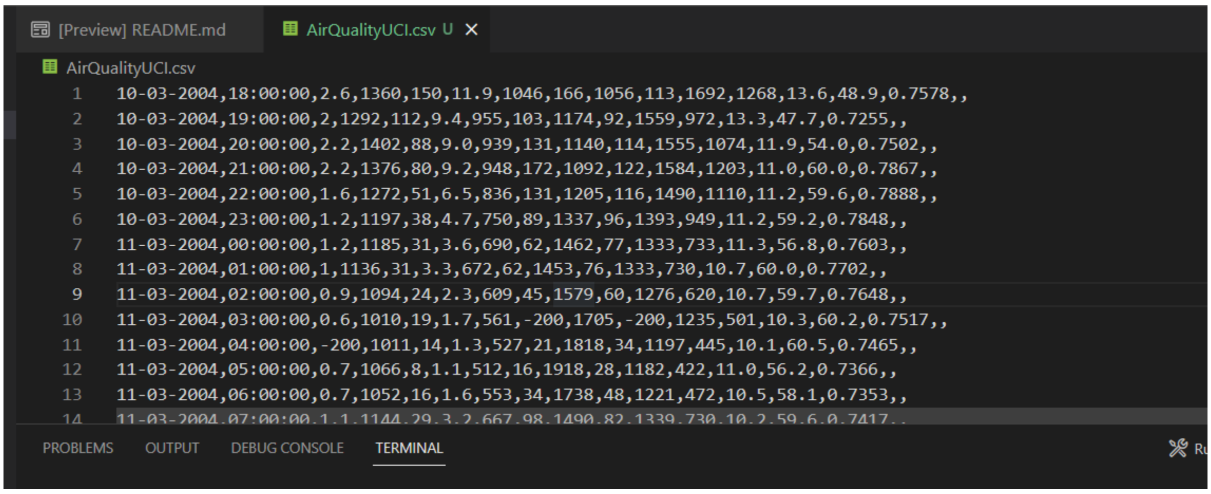
```
hive> select c.id,c.name,c.address,o.amount from orders o
    > full join customers c
    > on o.customer_id = c.id;
```



```
c.id    c.name  c.address       o.amount
1       Nimeshika       Madurai NULL
2       Sowmya  Coimbatore      50.0
3       Madhesh Virudhunagar    NULL
4       Vijay   Villupuram      40.0
5       Kiran   Bangalore       NULL
6       Maruthu Ramnad  20.0
```

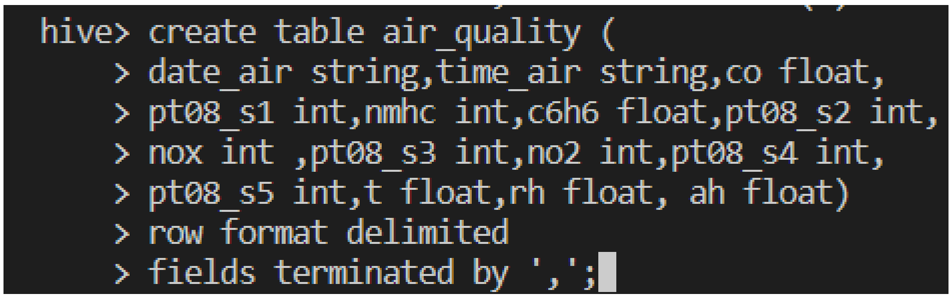
Download a data from the given location -

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



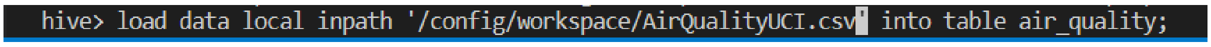
```
10-03-2004,18:00:00,2.6,1360,150,11.9,1046,166,1056,113,1692,1268,13.6,48.9,0.7578,,
10-03-2004,19:00:00,2,1292,112,9.4,955,103,1174,92,1559,972,13.3,47.7,0.7255,,
10-03-2004,20:00:00,2.2,1402,88,9.0,939,131,1140,114,1555,1074,11.9,54.0,0.7502,,
10-03-2004,21:00:00,2.2,1376,80,9.2,948,172,1092,122,1584,1203,11.0,60.0,0.7867,,
10-03-2004,22:00:00,1.6,1272,51,6.5,836,131,1205,116,1490,1110,11.2,59.6,0.7888,,
10-03-2004,23:00:00,1.2,1197,38,4.7,750,89,1337,96,1393,949,11.2,59.2,0.7848,,
11-03-2004,00:00:00,1.2,1185,31,3.6,690,62,1462,77,1333,733,11.3,56.8,0.7603,,
11-03-2004,01:00:00,1,1136,31,3.3,672,62,1453,76,1333,730,10.7,60.0,0.7702,,
11-03-2004,02:00:00,0.9,1094,24,2.3,609,45,1579,60,1276,620,10.7,59.7,0.7648,,
11-03-2004,03:00:00,0.6,1010,19,1.7,561,-200,1705,-200,1235,501,10.3,60.2,0.7517,,
11-03-2004,04:00:00,-200,1011,14,1.3,527,21,1818,34,1197,445,10.1,60.5,0.7465,,
11-03-2004,05:00:00,0.7,1066,8,1.1,512,16,1918,28,1182,422,11.0,56.2,0.7366,,
11-03-2004,06:00:00,0.7,1052,16,1.6,553,34,1738,48,1221,472,10.5,58.1,0.7353,,
11-03-2004,07:00:00,1.1,1144,29,3.2,667,98,1490,82,1339,730,10.2,59.6,0.7417,,
```

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



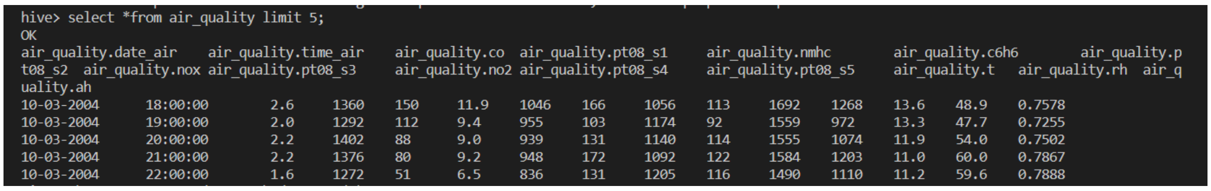
```
hive> create table air_quality (
    > date_air string,time_air string,co float,
    > pt08_s1 int,nmhc int,c6h6 float,pt08_s2 int,
    > nox int ,pt08_s3 int,no2 int,pt08_s4 int,
    > pt08_s5 int,t float,rh float, ah float)
    > row format delimited
    > fields terminated by ',';
```

2. try to place a data into table location

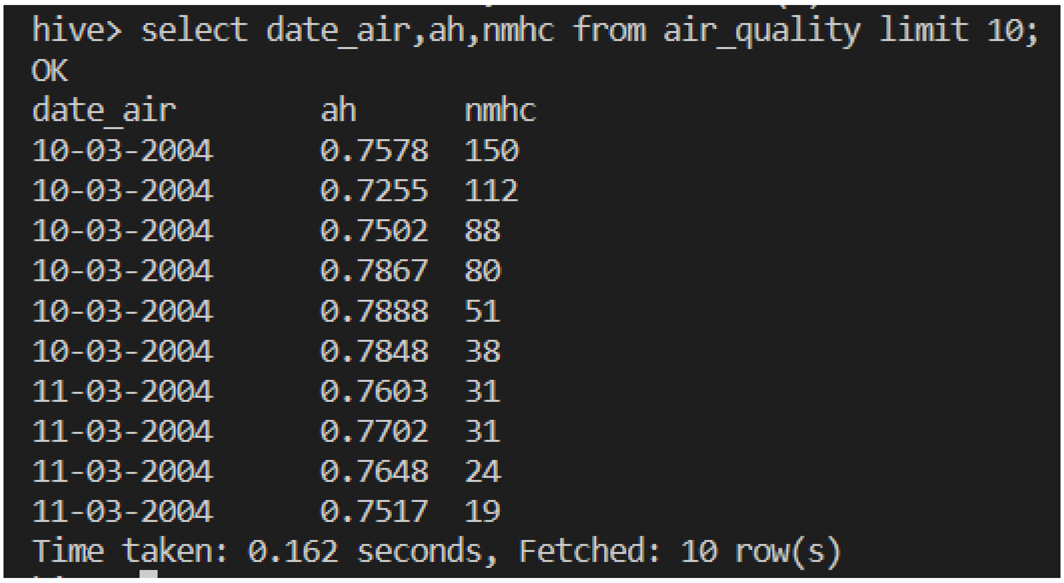


```
hive> load data local inpath '/config/workspace/AirQualityUCI.csv' into table air_quality;
```

3. Perform a select operation .

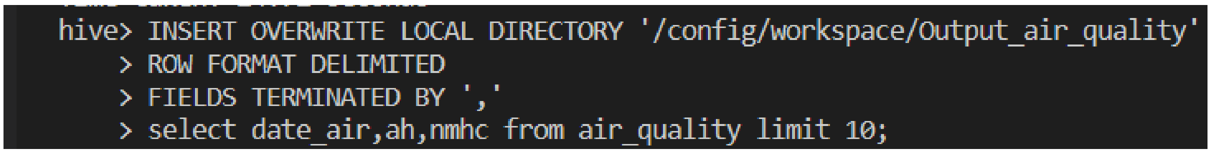


```
hive> select *from air_quality limit 5;
OK
air_quality.date_air    air_quality.time_air    air_quality.co  air_quality.pt08_s1     air_quality.nmhc        air_quality.c6h6        air_quality.p
t08_s2  air_quality.nox air_quality.pt08_s3     air_quality.no2 air_quality.pt08_s4     air_quality.pt08_s5     air_quality.t   air_quality.rh  air_q
uality.ah
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        2.0     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
```

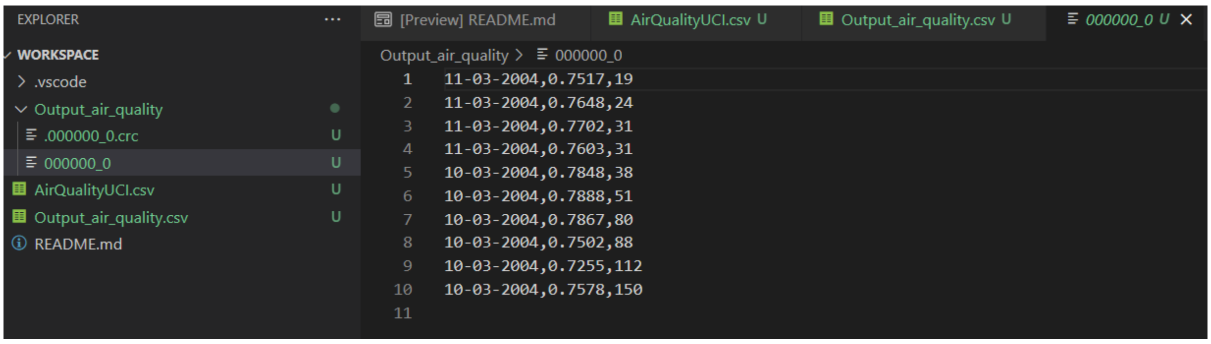


```
hive> select date_air,ah,nmhc from air_quality limit 10;
OK
date_air        ah      nmhc
10-03-2004      0.7578  150
10-03-2004      0.7255  112
10-03-2004      0.7502  88
10-03-2004      0.7867  80
10-03-2004      0.7888  51
10-03-2004      0.7848  38
11-03-2004      0.7603  31
11-03-2004      0.7702  31
11-03-2004      0.7648  24
11-03-2004      0.7517  19
Time taken: 0.162 seconds, Fetched: 10 row(s)
```

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

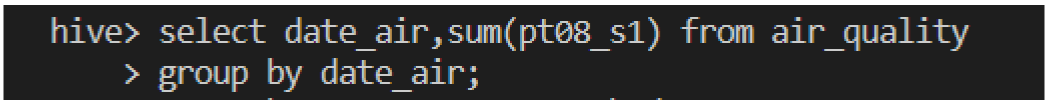


```
hive> INSERT OVERWRITE LOCAL DIRECTORY '/config/workspace/Output_air_quality'
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > select date_air,ah,nmhc from air_quality limit 10;
```

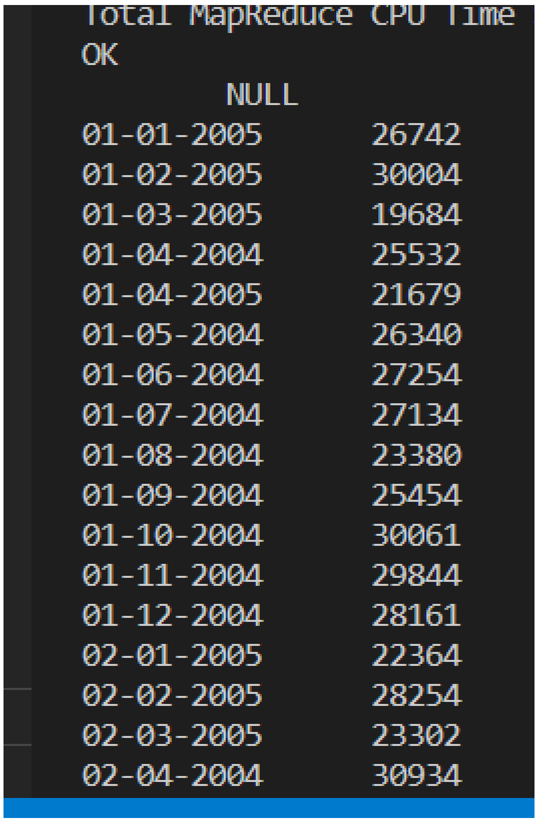


```
11-03-2004,0.7517,19
11-03-2004,0.7648,24
11-03-2004,0.7702,31
11-03-2004,0.7603,31
10-03-2004,0.7848,38
10-03-2004,0.7888,51
10-03-2004,0.7867,80
10-03-2004,0.7502,88
10-03-2004,0.7255,112
10-03-2004,0.7578,150
```

5. Perform group by operation.

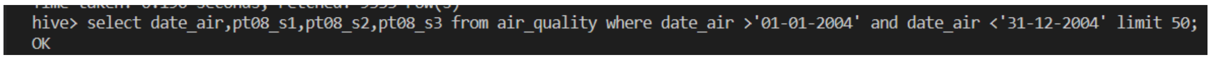


```
hive> select date_air,sum(pt08_s1) from air_quality
    > group by date_air;
```

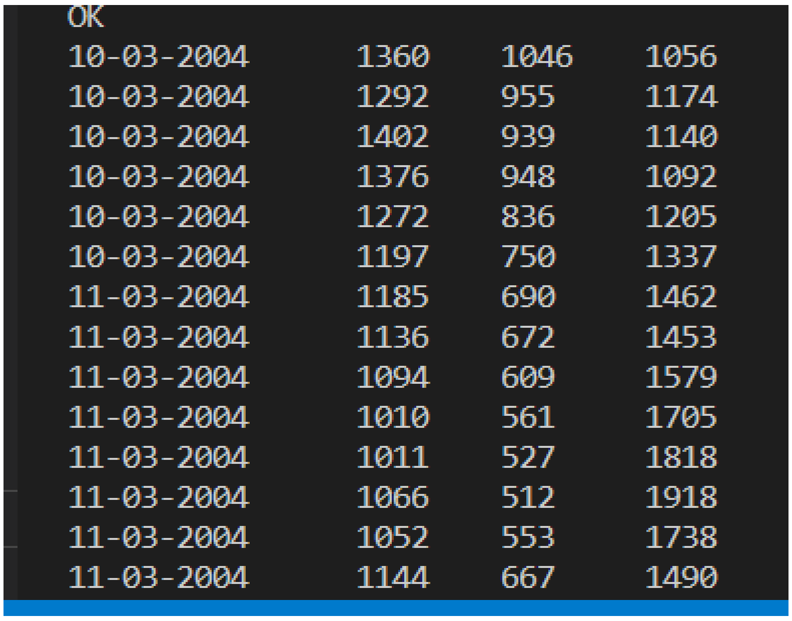


```
Total MapReduce CPU Time
OK
          NULL
01-01-2005      26742
01-02-2005      30004
01-03-2005      19684
01-04-2004      25532
01-04-2005      21679
01-05-2004      26340
01-06-2004      27254
01-07-2004      27134
01-08-2004      23380
01-09-2004      25454
01-10-2004      30061
01-11-2004      29844
01-12-2004      28161
02-01-2005      22364
02-02-2005      28254
02-03-2005      23302
02-04-2004      30934
```

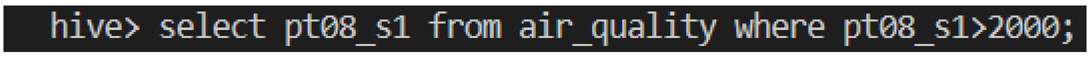
7. Perform filter operation at least 5 kinds of filter examples .



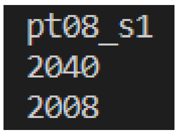
```
hive> select date_air,pt08_s1,pt08_s2,pt08_s3 from air_quality where date_air >'01-01-2004' and date_air <'31-12-2004' limit 50;
OK
```



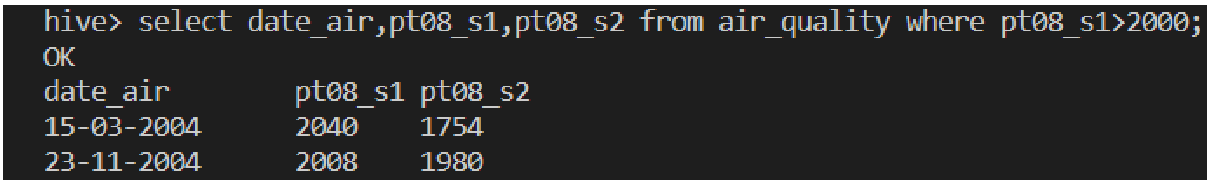
```
OK
10-03-2004      1360    1046    1056
10-03-2004      1292    955     1174
10-03-2004      1402    939     1140
10-03-2004      1376    948     1092
10-03-2004      1272    836     1205
10-03-2004      1197    750     1337
11-03-2004      1185    690     1462
11-03-2004      1136    672     1453
11-03-2004      1094    609     1579
11-03-2004      1010    561     1705
11-03-2004      1011    527     1818
11-03-2004      1066    512     1918
11-03-2004      1052    553     1738
11-03-2004      1144    667     1490
```



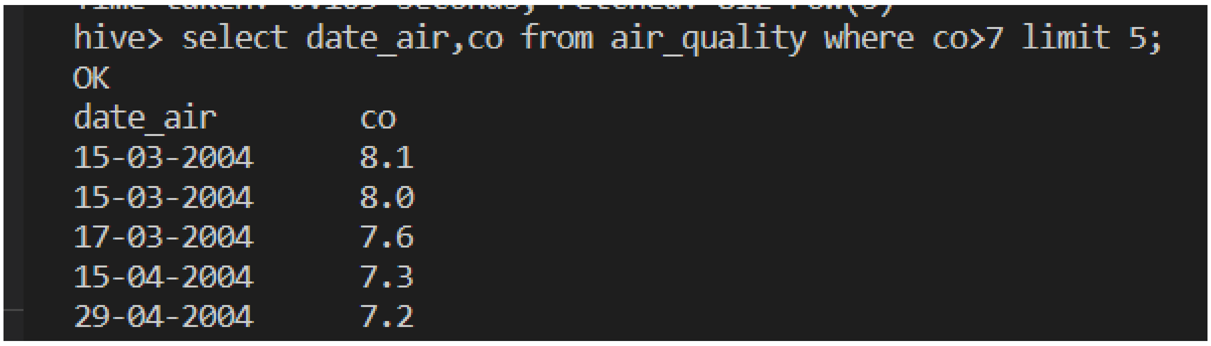
```
hive> select pt08_s1 from air_quality where pt08_s1>2000;
```



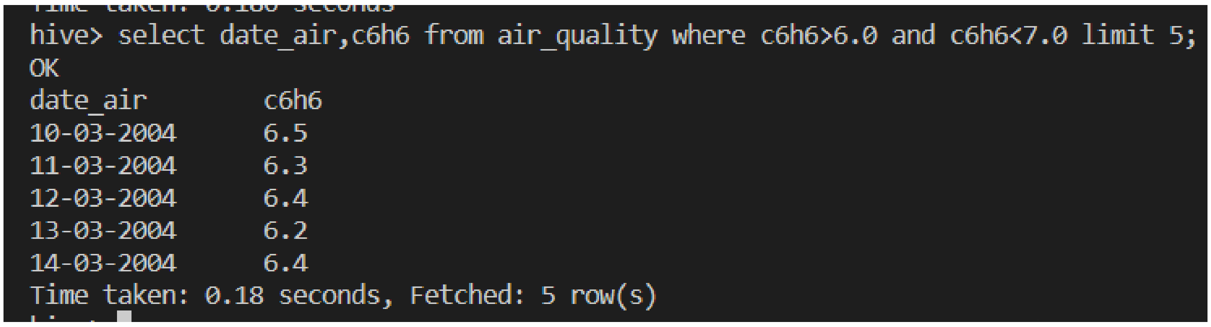
```
pt08_s1
2040
2008
```



```
hive> select date_air,pt08_s1,pt08_s2 from air_quality where pt08_s1>2000;
OK
date_air        pt08_s1 pt08_s2
15-03-2004      2040    1754
23-11-2004      2008    1980
```

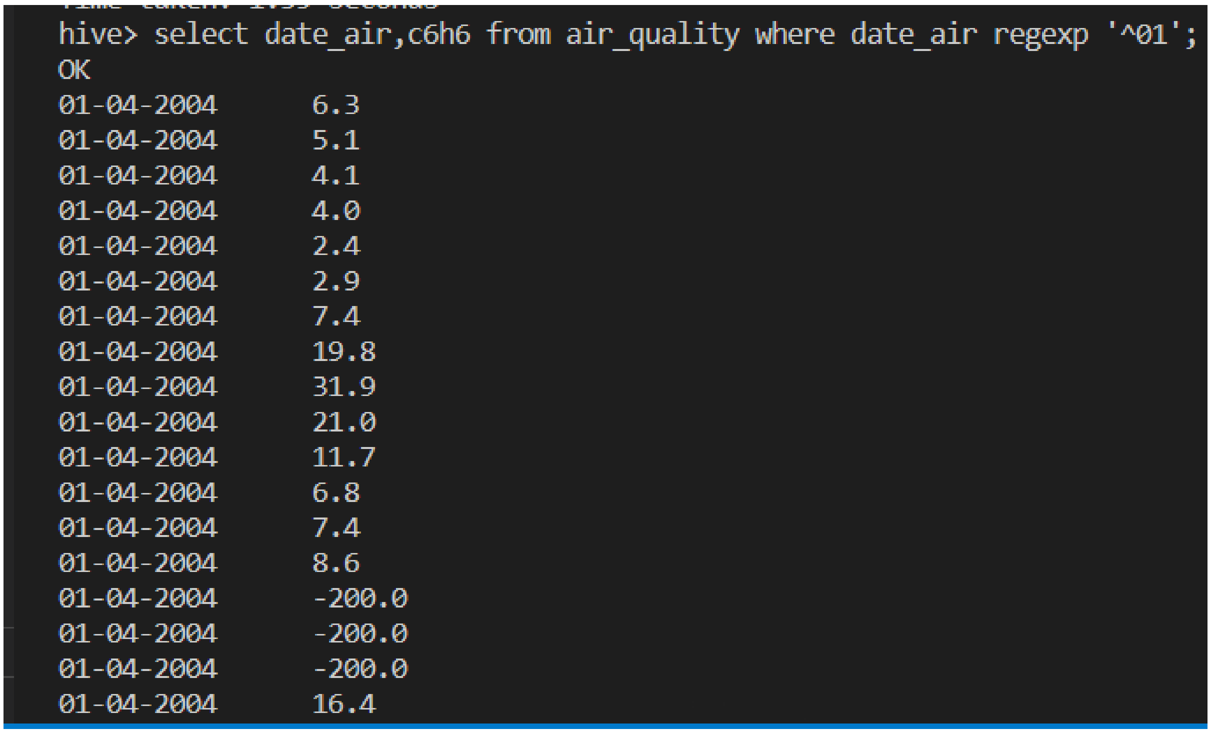


```
hive> select date_air,co from air_quality where co>7 limit 5;
OK
date_air        co
15-03-2004      8.1
15-03-2004      8.0
17-03-2004      7.6
15-04-2004      7.3
29-04-2004      7.2
```



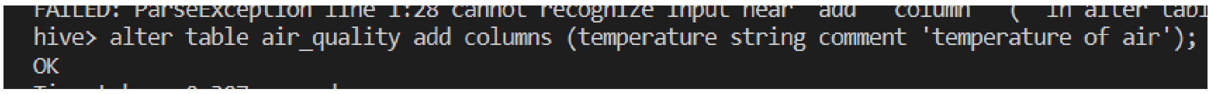
```
hive> select date_air,c6h6 from air_quality where c6h6>6.0 and c6h6<7.0 limit 5;
OK
date_air        c6h6
10-03-2004      6.5
11-03-2004      6.3
12-03-2004      6.4
13-03-2004      6.2
14-03-2004      6.4
Time taken: 0.18 seconds, Fetched: 5 row(s)
```

8. show and example of regex operation

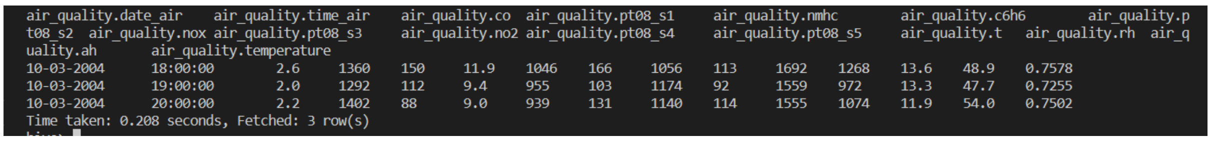


```
hive> select date_air,c6h6 from air_quality where date_air regexp '^01';
OK
01-04-2004      6.3
01-04-2004      5.1
01-04-2004      4.1
01-04-2004      4.0
01-04-2004      2.4
01-04-2004      2.9
01-04-2004      7.4
01-04-2004      19.8
01-04-2004      31.9
01-04-2004      21.0
01-04-2004      11.7
01-04-2004      6.8
01-04-2004      7.4
01-04-2004      8.6
01-04-2004      -200.0
01-04-2004      -200.0
01-04-2004      -200.0
01-04-2004      16.4
```

9. alter table operation

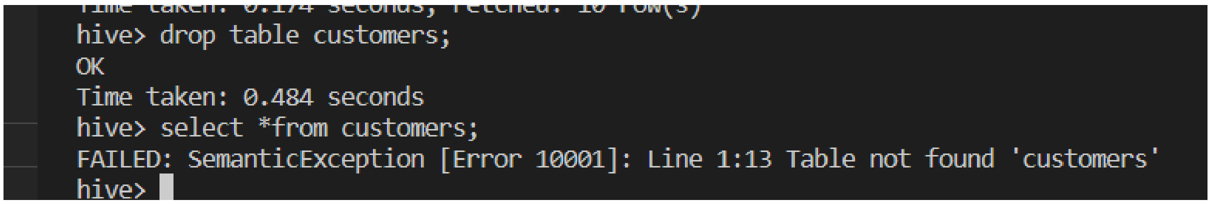


```
FAILED: ParseException line 1:28 cannot recognize input near 'add' 'column' '(' in alter tabl
hive> alter table air_quality add columns (temperature string comment 'temperature of air');
OK
```



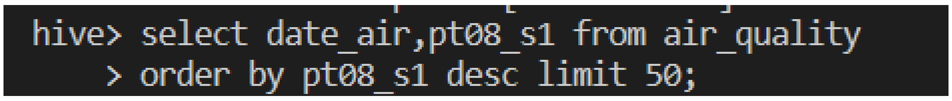
```
air_quality.date_air    air_quality.time_air    air_quality.co  air_quality.pt08_s1     air_quality.nmhc        air_quality.c6h6        air_quality.p
t08_s2  air_quality.nox air_quality.pt08_s3     air_quality.no2 air_quality.pt08_s4     air_quality.pt08_s5     air_quality.t   air_quality.rh  air_q
uality.ah       air_quality.temperature
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        2.0     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
Time taken: 0.208 seconds, Fetched: 3 row(s)
```

10 . drop table operation

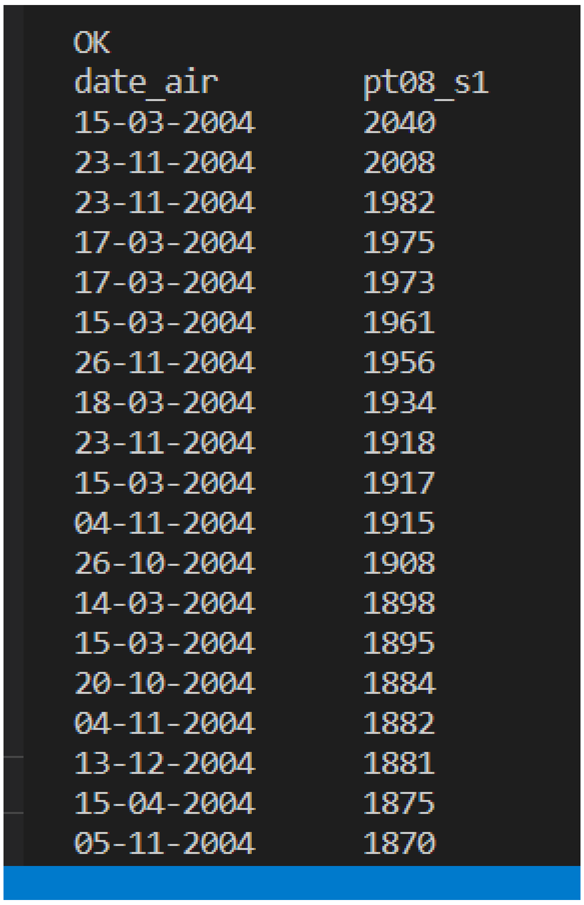


```
hive> drop table customers;
OK
Time taken: 0.484 seconds
hive> select *from customers;
FAILED: SemanticException [Error 10001]: Line 1:13 Table not found 'customers'
hive>
```

12 . order by operation .

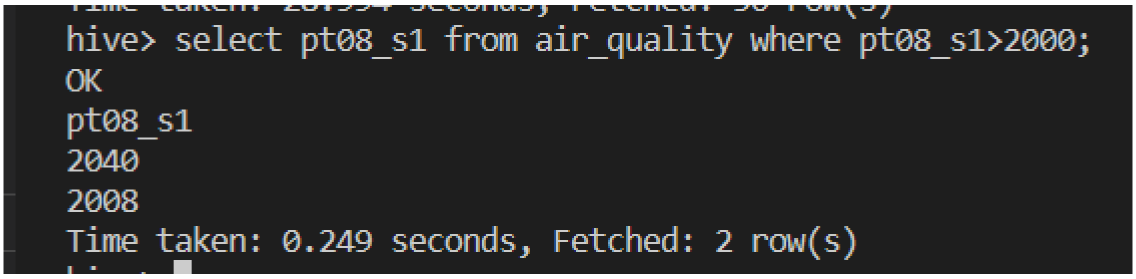


```
hive> select date_air,pt08_s1 from air_quality
    > order by pt08_s1 desc limit 50;
```



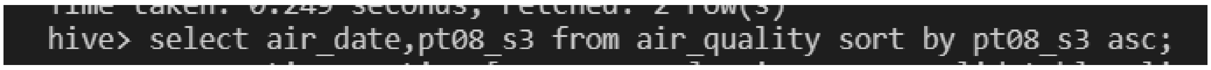
```
OK
date_air        pt08_s1
15-03-2004      2040
23-11-2004      2008
23-11-2004      1982
17-03-2004      1975
17-03-2004      1973
15-03-2004      1961
26-11-2004      1956
18-03-2004      1934
23-11-2004      1918
15-03-2004      1917
04-11-2004      1915
26-10-2004      1908
14-03-2004      1898
15-03-2004      1895
20-10-2004      1884
04-11-2004      1882
13-12-2004      1881
15-04-2004      1875
05-11-2004      1870
```

13 . where clause operations you have to perform .

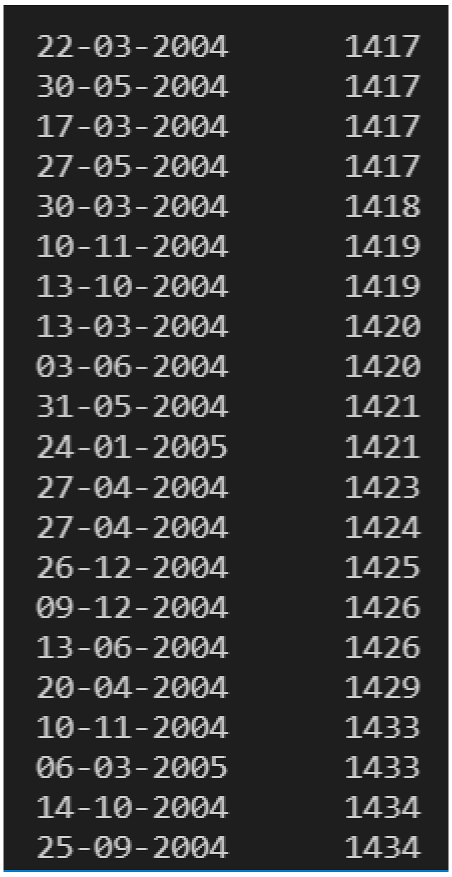


```
hive> select pt08_s1 from air_quality where pt08_s1>2000;
OK
pt08_s1
2040
2008
Time taken: 0.249 seconds, Fetched: 2 row(s)
```

14 . sorting operation you have to perform .

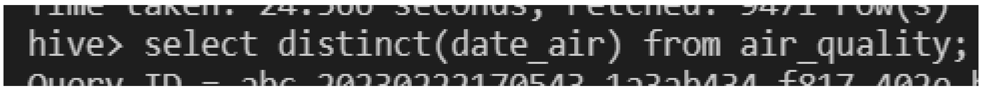


```
hive> select air_date,pt08_s3 from air_quality sort by pt08_s3 asc;
```

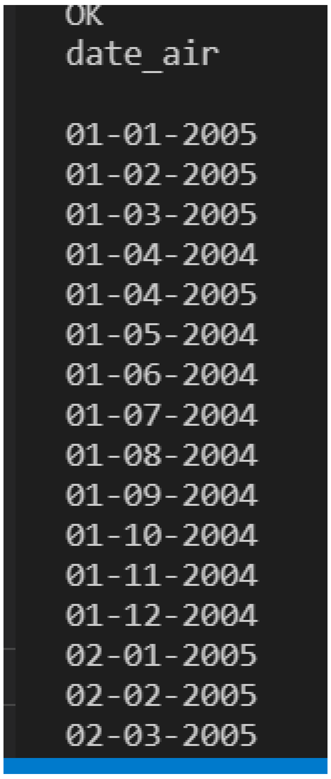


```
22-03-2004      1417
30-05-2004      1417
17-03-2004      1417
27-05-2004      1417
30-03-2004      1418
10-11-2004      1419
13-10-2004      1419
13-03-2004      1420
03-06-2004      1420
31-05-2004      1421
24-01-2005      1421
27-04-2004      1423
27-04-2004      1424
26-12-2004      1425
09-12-2004      1426
13-06-2004      1426
20-04-2004      1429
10-11-2004      1433
06-03-2005      1433
14-10-2004      1434
25-09-2004      1434
```

15 . distinct operation you have to perform .



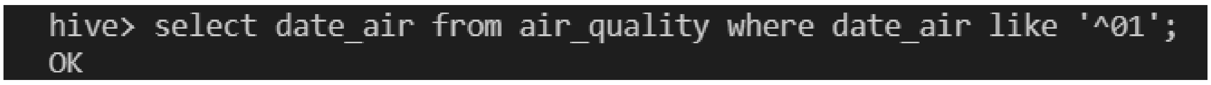
```
hive> select distinct(date_air) from air_quality;
```



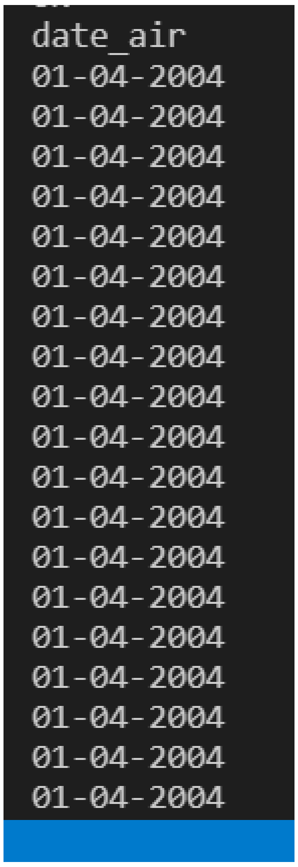
```
OK
date_air

01-01-2005
01-02-2005
01-03-2005
01-04-2004
01-04-2005
01-05-2004
01-06-2004
01-07-2004
01-08-2004
01-09-2004
01-10-2004
01-11-2004
01-12-2004
02-01-2005
02-02-2005
02-03-2005
```

16 . like an operation you have to perform .

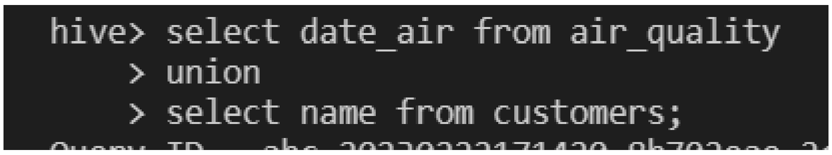


```
hive> select date_air from air_quality where date_air like '^01';
OK
```

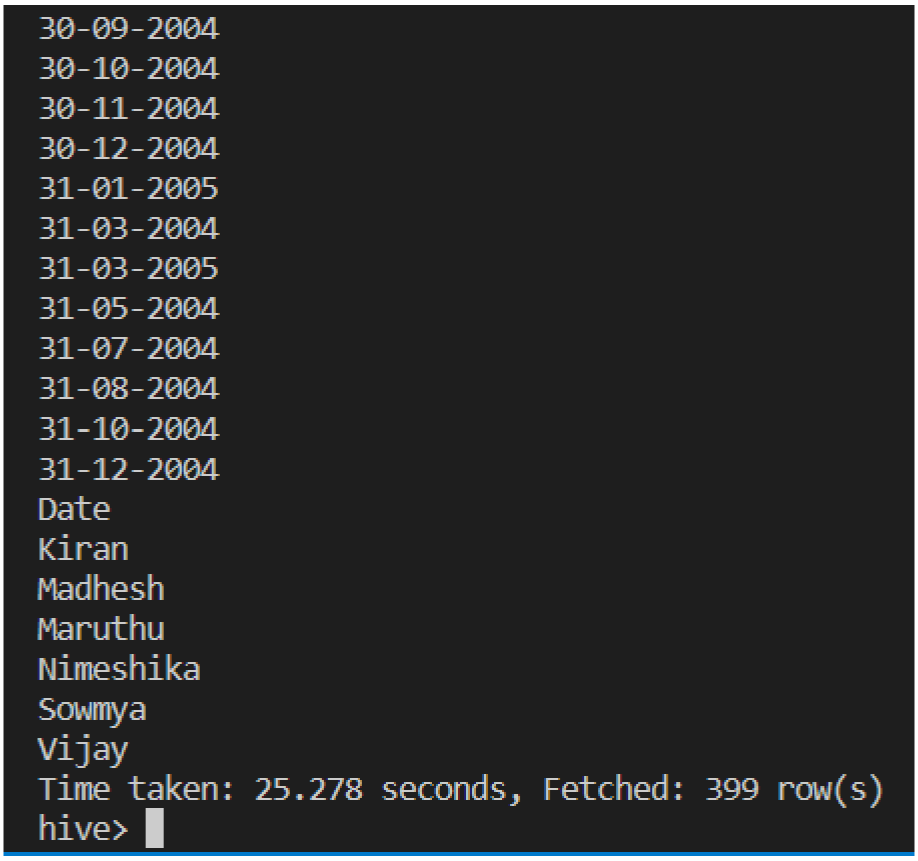


```
date_air
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
01-04-2004
```

17 . union operation you have to perform .

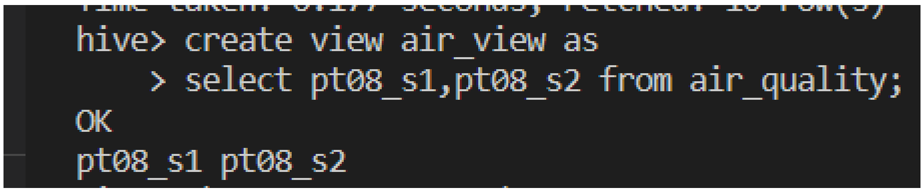


```
hive> select date_air from air_quality
    > union
    > select name from customers;
```

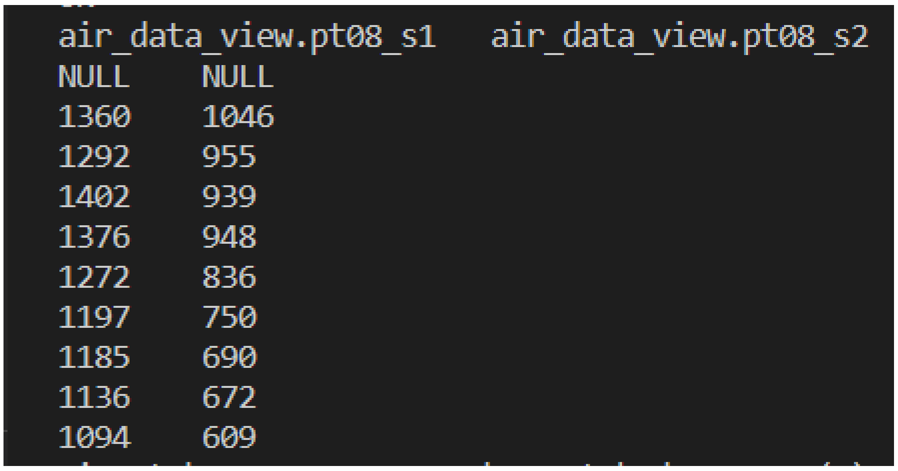


```
30-09-2004
30-10-2004
30-11-2004
30-12-2004
31-01-2005
31-03-2004
31-03-2005
31-05-2004
31-07-2004
31-08-2004
31-10-2004
31-12-2004
Date
Kiran
Madhesh
Maruthu
Nimeshika
Sowmya
Vijay
Time taken: 25.278 seconds, Fetched: 399 row(s)
hive>
```

18 . table view operation you have to perform.



```
hive> create view air_view as
    > select pt08_s1,pt08_s2 from air_quality;
OK
pt08_s1 pt08_s2
```



```
air_data_view.pt08_s1   air_data_view.pt08_s2
NULL    NULL
1360    1046
1292    955
1402    939
1376    948
1272    836
1197    750
1185    690
1136    672
1094    609
```

**Scenario Based questions:**

**1.Will the reducer work or not if you use "Limit 1" in any HiveQL query?**

Reducer will not work if we use limit in select clause,

When used in the other aggregation query even though if we use limit in the query the reducer will work.

**2.Suppose I have installed Apache Hive on top of my Hadoop cluster using default meta store configuration. Then, what will happen if we have multiple clients trying to access Hive at the same time?**

The default meta store configuration allows only one hive session to opened at a time for accessing the metastore.

If multiple clients try to access the metastore then they will get a error.

**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 issue by Partitioning the table according to reach month.

Hence we will be scanning the data for particular month and not the whole dataset.

Steps are:

a)create a partition table

```
CREATE TABLE transaction_details_parition
 (cust_id INT, amount FLOAT, month STRING, country
STRING)
 PARTITIONED BY (month STRING)
 ROW FORMAT DELIMITED
 FIELDS TERMINATED BY ‘,’ ;
```

b)enable dynamic partitioning

```
SET hive.exec.dynamic.partition = true;
```

c)transfer the data from non partitioned table to partitioned table.

```
INSERT OVERWRITE TABLE transacation_details_partition
(month) SELECT cust_id, amount, country, month FROM
transaction_details;
```

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

With the help of alter table command

```
ALTER TABLE transcation_details_partition ADD PARTITION
(month=’December’) location ‘’
```

**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?**

To remove this error we need to execute the commands as,

```
SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partiton.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?**

SERDE allows us to convert the unstructured bytes into records then we can process it using hive.

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 '';

**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?**

It can be resolved using SequenceFile(it will help to group the small files into a single file)

Create a temporary table:

CREATE TABLE temp_table (id INT, name STRING, e-mail STRING, country STRING)

ROW FORMAT FIELDS DELIMITED TERMINATED BY ‘,’ STORED AS TEXTFILE;

Load the data into temp_table:

LOAD DATA INPATH ‘/input’ INTO TABLE temp_table;

Create a table that will store data in SequenceFile format:

CREATE TABLE sample_seqfile (id INT, name STRING, e-mail STRING, country STRING)

ROW FORMAT FIELDS DELIMITED TERMINATED BY ‘,’ STORED AS SEQUENCEFILE;

Transfer the data from the temporary table into the sample_seqfile table:

INSERT OVERWRITE TABLE sample SELECT * FROM temp_table;

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

**OVERWRITE INTO TABLE address;**

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

Several errors may occur as,

a)Syntax error

b)Permission issue

c)Missing file

d)Table not found

e)Overwrite issue

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

Yes, it is possible to add 100 nodes to an existing Hive cluster with 100 nodes. The process of adding more nodes to an existing Hive cluster is known as scaling out.

Here are the general steps to add more nodes to a Hive cluster:

a) Install the necessary software on the new nodes, such as Hadoop and Hive.

b) Configure the new nodes to join the existing cluster. This involves updating the configuration files of the new nodes with the information about the existing nodes and their roles in the cluster.

c) Start the Hadoop and Hive services on the new nodes.

d) Once the new nodes have joined the cluster and are up and running, you can rebalance the data across the nodes to ensure that the workload is distributed evenly.