**Hive Assignment-2**

**Scenario Based questions:**

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

It depends on what your query is.

e.g.

-If the 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 like that, then reducers will be 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. 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.

**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?**

- Create a partition table as below,

hive> create table part\_table (cust\_id INT, amount FLOAT, month STRING, country STRING)

> row format delimited

> fields terminated by ‘,’ ;

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

- Load the required data.

hive> insert overwrite table part\_table(month) partition(month) select cust\_id,sum(amount) as total\_revenue,month,country from transaction\_details;

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

hive> alter table part\_table add partition(month="December") location '\location of 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?**

We have to follow below steps.

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 table employee (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?**

We can use the SequenceFile format which will group these small files together to form a single sequence file.

We cannot append the data in hive in text file.

* Create a temporary table as shown below,

>create table temp\_table (id int, name string, e-mail string, country string)

>row format delimited

>field terminated by ‘,’

>lines terminated by ‘\n’

>stored as textfile;

- Load the data into temp\_table,as shown below,

>load data inpath ‘/input’ into table temp\_table;

- Create a table that will store data in SequenceFile format as shown below,

>create table sample\_seqfile (id int, name string, e-mail string, country string)

>row format delimited

>field terminated by ‘,’

>lines terminated by ‘\n’

>stored as sequencefile;

- Transfer the data from the temporary table into the sample\_seqfile table:

>insert overwrite table sample\_seqfile select \* from temp\_table;

So now a single SequenceFile is generated which contains the data present in all of the input files and therefore, the problem of having lots of small files is eliminated.

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

**OVERWRITE INTO TABLE address;**

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

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

If you want to load the data from hadoop file system then we do not have to mention "LOCAL".

**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)**

hive> create table customer(id int, name string, age int, address string, salary int)

> row format delimited

> fields terminated by '|'

> lines terminated by '\n'

> stored as textfile;

OK

Time taken: 0.158 seconds

hive> load data local inpath '/home/gopalkrishna/KRP/Customers\_Assignment.txt' into table customer;

Loading data to table default.customer

Table default.customer stats: [numFiles=1, totalSize=184]

OK

Time taken: 1.083 seconds

hive> create table orders(oid int, o\_date string, customer\_id int, amount int)

> row format delimited

> fields terminated by '|'

> lines terminated by '\n'

> stored as textfile;

OK

Time taken: 0.159 seconds

hive> load data local inpath '/home/gopalkrishna/KRP/Orders\_Assignment.txt' into table orders;

Loading data to table default.orders

Table default.orders stats: [numFiles=1, totalSize=184]

OK

Time taken: 0.578 seconds

hive> select \* from customer;

OK

101 Maria 23 USA 60000

102 Phyllis 22 Canada 85000

103 Russell 27 USA 72790

105 Riya 30 India 101000

108 Sarah 22 Norway NULL

109 Jose 26 Canada 77000

110 Christine 23 Sweden 87430

Time taken: 0.883 seconds, Fetched: 7 row(s)

hive> select \* from orders;

OK

10001 22/11/2018 101 570

10023 13/01/2019 102 3500

1009 23/06/2013 103 300

10033 27/11/2019 105 500

10078 11/12/2021 108 560

10065 13/06/2021 106 348

10011 17/09/2017 110 5000

Time taken: 0.186 seconds, Fetched: 7 row(s)

Note: hive > set hive.cli.print.header=true; ===>>> If we use this command ,it will show header.

Here in the tables, I’ve not given header.

Inner Join:-

hive> select \* from customer c inner join orders o on c.id=o.customer\_id;

Query ID = gopalkrishna\_20230517064643\_b674e681-14c0-4915-9609-e42851099492

Total jobs = 1

Execution log at: /tmp/gopalkrishna/gopalkrishna\_20230517064643\_b674e681-14c0-4915-9609-e42851099492.log

2023-05-17 06:46:53 Starting to launch local task to process map join; maximum memory = 518979584

2023-05-17 06:46:56 Dump the side-table for tag: 0 with group count: 7 into file: file:/tmp/gopalkrishna/3f909f85-af91-4c4a-b1f6-fc3865400a32/hive\_2023-05-17\_06-46-43\_130\_5378669139224779684-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile20--.hashtable

2023-05-17 06:46:56 Uploaded 1 File to: file:/tmp/gopalkrishna/3f909f85-af91-4c4a-b1f6-fc3865400a32/hive\_2023-05-17\_06-46-43\_130\_5378669139224779684-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile20--.hashtable (516 bytes)

2023-05-17 06:46:56 End of local task; Time Taken: 2.911 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\_1683964517605\_0072, Tracking URL = http://ubuntu:8088/proxy/application\_1683964517605\_0072/

Kill Command = /home/gopalkrishna/INSTALL/hadoop-2.6.0/bin/hadoop job -kill job\_1683964517605\_0072

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0

2023-05-17 06:47:11,380 Stage-3 map = 0%, reduce = 0%

2023-05-17 06:47:28,361 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 3.04 sec

MapReduce Total cumulative CPU time: 3 seconds 40 msec

Ended Job = job\_1683964517605\_0072

MapReduce Jobs Launched:

Stage-Stage-3: Map: 1 Cumulative CPU: 3.04 sec HDFS Read: 7657 HDFS Write: 306 SUCCESS

Total MapReduce CPU Time Spent: 3 seconds 40 msec

OK

101 Maria 23 USA 60000 10001 22/11/2018 101 570

102 Phyllis 22 Canada85000 10023 13/01/2019 102 3500

103 Russell27 USA 72790 1009 23/06/2013 103 300

105 Riya 30 India 10100010033 27/11/2019 105 500

108 Sarah 22 NorwayNULL 10078 11/12/2021 108 560

110 Christine 23 Sweden87430 10011 17/09/2017 110 5000

Time taken: 47.458 seconds, Fetched: 6 row(s)

Left Outer Join:-

hive> select \* from customer c left outer join orders o on c.id = o.customer\_id;

Query ID = gopalkrishna\_20230517065634\_4c23f88c-3d30-458c-b9ba-32ad2d6a1791

Total jobs = 1

Execution log at: /tmp/gopalkrishna/gopalkrishna\_20230517065634\_4c23f88c-3d30-458c-b9ba-32ad2d6a1791.log

2023-05-17 06:56:43 Starting to launch local task to process map join; maximum memory = 518979584

2023-05-17 06:56:45 Dump the side-table for tag: 1 with group count: 7 into file: file:/tmp/gopalkrishna/3f909f85-af91-4c4a-b1f6-fc3865400a32/hive\_2023-05-17\_06-56-34\_480\_8450034859433743855-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile41--.hashtable

2023-05-17 06:56:45 Uploaded 1 File to: file:/tmp/gopalkrishna/3f909f85-af91-4c4a-b1f6-fc3865400a32/hive\_2023-05-17\_06-56-34\_480\_8450034859433743855-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile41--.hashtable (516 bytes)

2023-05-17 06:56:45 End of local task; Time Taken: 2.103 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\_1683964517605\_0074, Tracking URL = http://ubuntu:8088/proxy/application\_1683964517605\_0074/

Kill Command = /home/gopalkrishna/INSTALL/hadoop-2.6.0/bin/hadoop job -kill job\_1683964517605\_0074

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0

2023-05-17 06:56:59,795 Stage-3 map = 0%, reduce = 0%

2023-05-17 06:57:12,942 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.38 sec

MapReduce Total cumulative CPU time: 2 seconds 380 msec

Ended Job = job\_1683964517605\_0074

MapReduce Jobs Launched:

Stage-Stage-3: Map: 1 Cumulative CPU: 2.38 sec HDFS Read: 7471 HDFS Write: 343 SUCCESS

Total MapReduce CPU Time Spent: 2 seconds 380 msec

OK

101 Maria 23 USA 60000 10001 22/11/2018 101 570

102 Phyllis 22 Canada85000 10023 13/01/2019 102 3500

103 Russell27 USA 72790 1009 23/06/2013 103 300

105 Riya 30 India 10100010033 27/11/2019 105 500

108 Sarah 22 NorwayNULL 10078 11/12/2021 108 560

109 Jose 26 Canada77000 NULL NULL NULL NULL

110 Christine 23 Sweden87430 10011 17/09/2017 110 5000

Time taken: 39.637 seconds, Fetched: 7 row(s)

Right Outer Join:-

hive> select \* from customer c right outer join orders o on c.id = o.customer\_id;

Query ID = gopalkrishna\_20230517070127\_6860d9db-311f-40b7-b464-bcec5f7a1b62

Total jobs = 1

Execution log at: /tmp/gopalkrishna/gopalkrishna\_20230517070127\_6860d9db-311f-40b7-b464-bcec5f7a1b62.log

2023-05-17 07:01:36 Starting to launch local task to process map join; maximum memory = 518979584

2023-05-17 07:01:37 Dump the side-table for tag: 0 with group count: 7 into file: file:/tmp/gopalkrishna/3f909f85-af91-4c4a-b1f6-fc3865400a32/hive\_2023-05-17\_07-01-27\_177\_2889235994196029026-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile50--.hashtable

2023-05-17 07:01:38 Uploaded 1 File to: file:/tmp/gopalkrishna/3f909f85-af91-4c4a-b1f6-fc3865400a32/hive\_2023-05-17\_07-01-27\_177\_2889235994196029026-1/-local-10003/HashTable-Stage-3/MapJoin-mapfile50--.hashtable (516 bytes)

2023-05-17 07:01:38 End of local task; Time Taken: 2.092 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\_1683964517605\_0075, Tracking URL = http://ubuntu:8088/proxy/application\_1683964517605\_0075/

Kill Command = /home/gopalkrishna/INSTALL/hadoop-2.6.0/bin/hadoop job -kill job\_1683964517605\_0075

Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 0

2023-05-17 07:01:53,309 Stage-3 map = 0%, reduce = 0%

2023-05-17 07:02:05,384 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 2.31 sec

MapReduce Total cumulative CPU time: 2 seconds 310 msec

Ended Job = job\_1683964517605\_0075

MapReduce Jobs Launched:

Stage-Stage-3: Map: 1 Cumulative CPU: 2.31 sec HDFS Read: 7452 HDFS Write: 346 SUCCESS

Total MapReduce CPU Time Spent: 2 seconds 310 msec

OK

101 Maria 23 USA 60000 10001 22/11/2018 101 570

102 Phyllis 22 Canada 85000 10023 13/01/2019 102 3500

103 Russell 27 USA 72790 1009 23/06/2013 103 300

105 Riya 30 India 101000 10033 27/11/2019 105 500

108 Sarah 22 Norway NULL 10078 11/12/2021 108 560

NULL NULL NULL NULL NULL 10065 13/06/2021 106 348

110 Christine 23 Sweden 87430 10011 17/09/2017 110 5000

Time taken: 40.467 seconds, Fetched: 7 row(s)

Full Outer Join:-

hive> select \* from customer c full outer join orders o on c.id = o.customer\_id;

Query ID = gopalkrishna\_20230517070331\_01b71700-0527-4487-8d7c-26df8ab4bd73

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\_1683964517605\_0076, Tracking URL = http://ubuntu:8088/proxy/application\_1683964517605\_0076/

Kill Command = /home/gopalkrishna/INSTALL/hadoop-2.6.0/bin/hadoop job -kill job\_1683964517605\_0076

Hadoop job information for Stage-1: number of mappers: 2; number of reducers: 1

2023-05-17 07:03:45,101 Stage-1 map = 0%, reduce = 0%

2023-05-17 07:04:12,461 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 7.03 sec

2023-05-17 07:04:25,539 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 9.34 sec

MapReduce Total cumulative CPU time: 9 seconds 340 msec

Ended Job = job\_1683964517605\_0076

MapReduce Jobs Launched:

Stage-Stage-1: Map: 2 Reduce: 1 Cumulative CPU: 9.34 sec HDFS Read: 14892 HDFS Write: 383 SUCCESS

Total MapReduce CPU Time Spent: 9 seconds 340 msec

OK

101 Maria 23 USA 60000 10001 22/11/2018 101 570

102 Phyllis 22 Canada85000 10023 13/01/2019 102 3500

103 Russell27 USA 72790 1009 23/06/2013 103 300

105 Riya 30 India 10100010033 27/11/2019 105 500

NULL NULL NULL NULL NULL 10065 13/06/2021 106 348

108 Sarah 22 NorwayNULL 10078 11/12/2021 108 560

109 Jose 26 Canada77000 NULL NULL NULL NULL

110 Christine 23 Sweden87430 10011 17/09/2017 110 5000

Time taken: 56.694 seconds, Fetched: 8 row(s)

**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**

hive> create table air\_quality\_UCI(Dates string,Times string,CO\_GT float,PT08\_S1\_CO int,NMHC\_GT int,C6H6\_GT float,PT08\_S2\_NMHC int,NOx\_GT int,PT08\_S3\_NOx int,NO2\_GT int,PT08\_S4\_NO2 int,PT08\_S5\_O3 int,T float,RH float,AH float)

> row format delimited

> fields terminated by ','

> lines terminated by '\n'

> tblproperties("skip.header.line.count"="1");

OK

Time taken: 3.678 seconds

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

hive> load data local inpath '/home/gopalkrishna/KRP/AirQualityUCI\_CSV.csv' into table air\_quality\_UCI;

Loading data to table default.air\_quality\_uci

Table default.air\_quality\_uci stats: [numFiles=1, totalSize=785065]

OK

Time taken: 1.87 seconds

1. **Perform a select operation .**

hive> set hive.cli.print.header=true;

hive> select \* from air\_quality\_UCI limit 5;

OK

air\_quality\_uci.dates air\_quality\_uci.times air\_quality\_uci.co\_gt air\_quality\_uci.pt08\_s1\_co air\_quality\_uci.nmhc\_gt air\_quality\_uci.c6h6\_gt air\_quality\_uci.pt08\_s2\_nmhc air\_quality\_uci.nox\_gt air\_quality\_uci.pt08\_s3\_nox air\_quality\_uci.no2\_gt air\_quality\_uci.pt08\_s4\_no2 air\_quality\_uci.pt08\_s5\_o3 air\_quality\_uci.t air\_quality\_uci.rh air\_quality\_uci.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

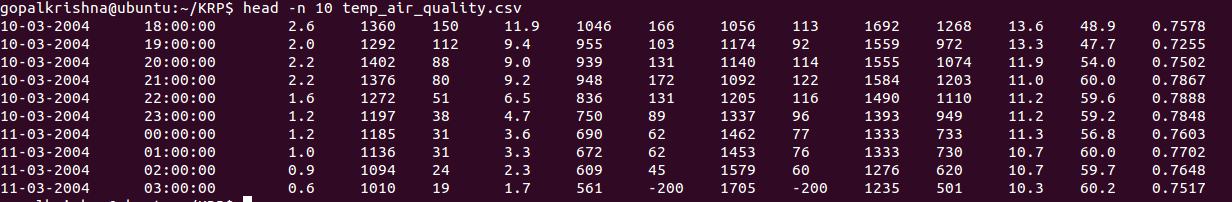
10-03-2004 22:00:00 1.6 1272 51 6.5 836 131 1205 116 1490 1110 11.2 59.6 0.7888

Time taken: 0.733 seconds, Fetched: 5 row(s)

****

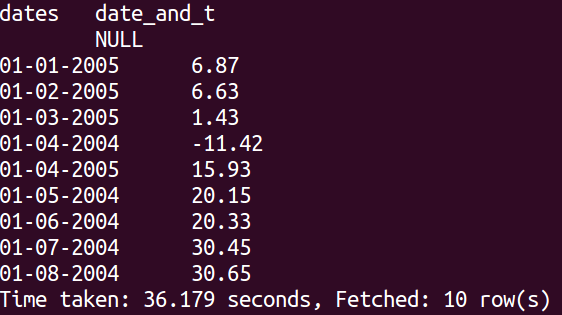
1. **Fetch the result of the select operation in your local as a csv file .**

gopalkrishna@ubuntu:~$ hive -e 'select \* from air\_quality\_uci' > /home/gopalkrishna/KRP/temp\_air\_quality.csv



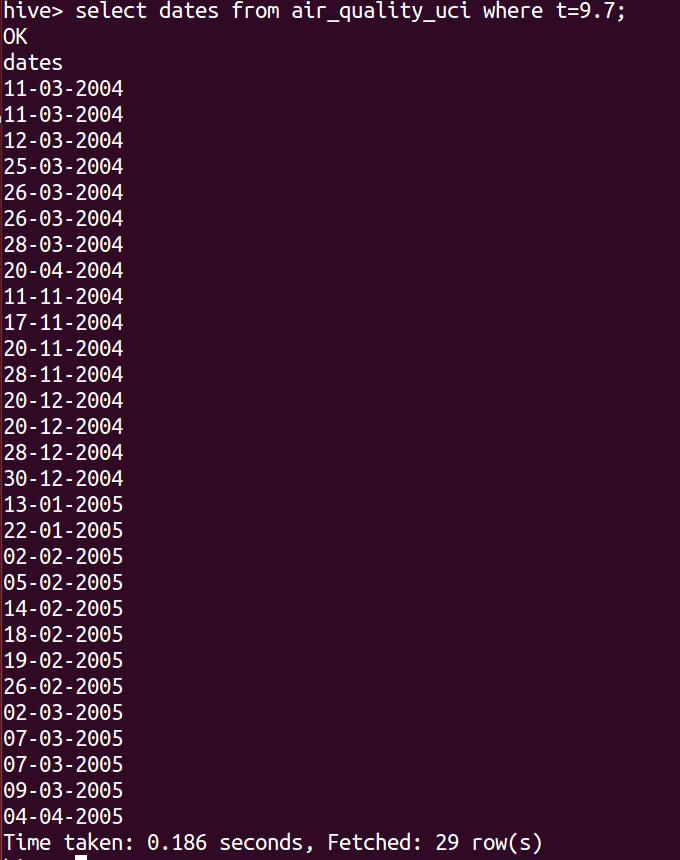
1. **Perform group by operation .**

hive> select dates,round(avg(T),2) as Date\_and\_T from air\_quality\_uci group by (dates) limit 10;

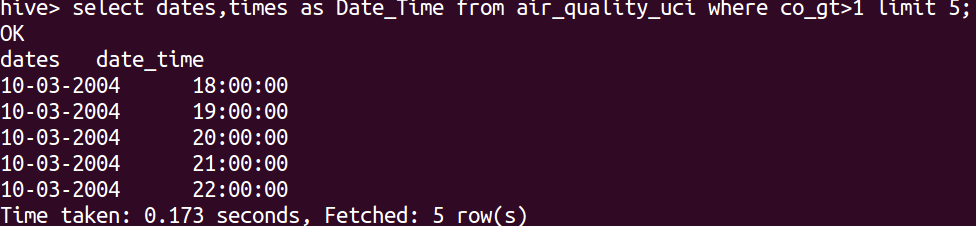


1. **Perform filter operation at least 5 kinds of filter examples .**

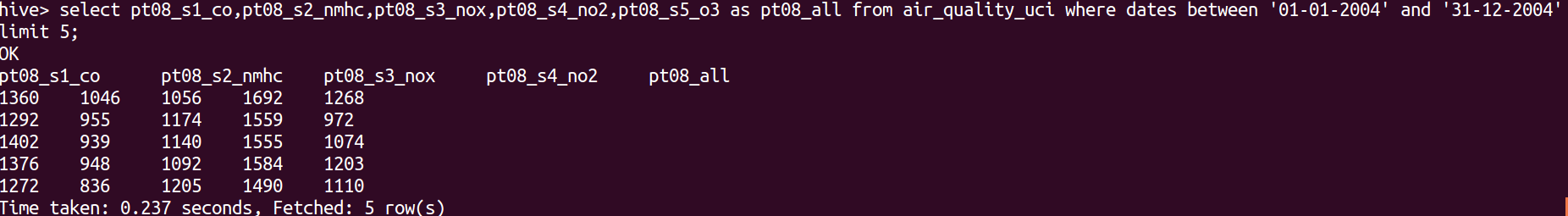
hive> select dates from air\_quality\_uci where t=9.7;



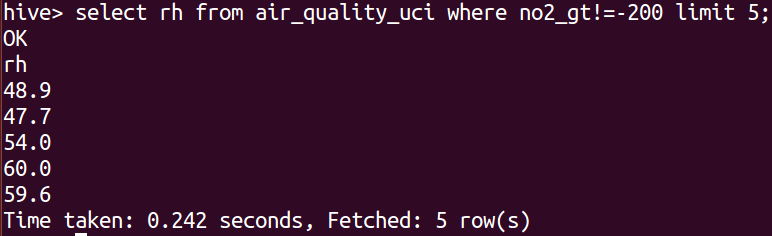
hive> select dates,times as Date\_Time from air\_quality\_uci where co\_gt>1 limit 5;



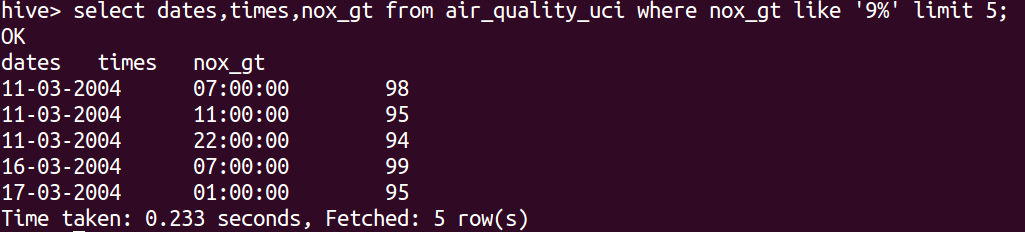
hive> select pt08\_s1\_co,pt08\_s2\_nmhc,pt08\_s3\_nox,pt08\_s4\_no2,pt08\_s5\_o3 as pt08\_all from air\_quality\_uci where dates between '01-01-2004' and '31-12-2004'limit 5;



hive> select rh from air\_quality\_uci where no2\_gt!=-200 limit 5;

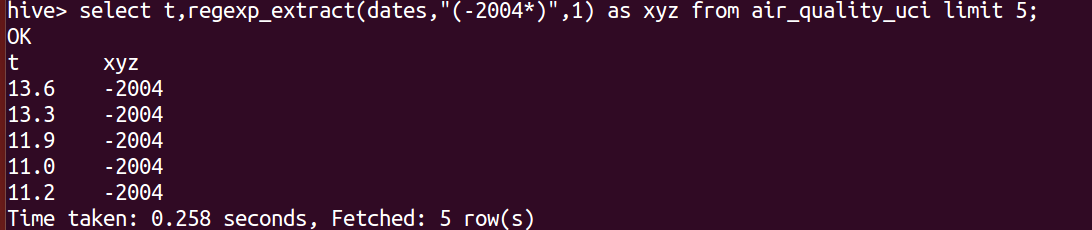


hive> select dates,times,nox\_gt from air\_quality\_uci where nox\_gt like '9%' limit 5;

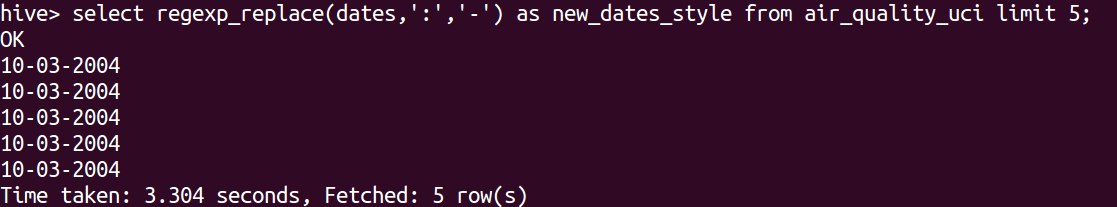


1. **show and example of regex operation**

hive> select t,regexp\_extract(dates,"(-2004\*)",1) as xyz from air\_quality\_uci limit 5;

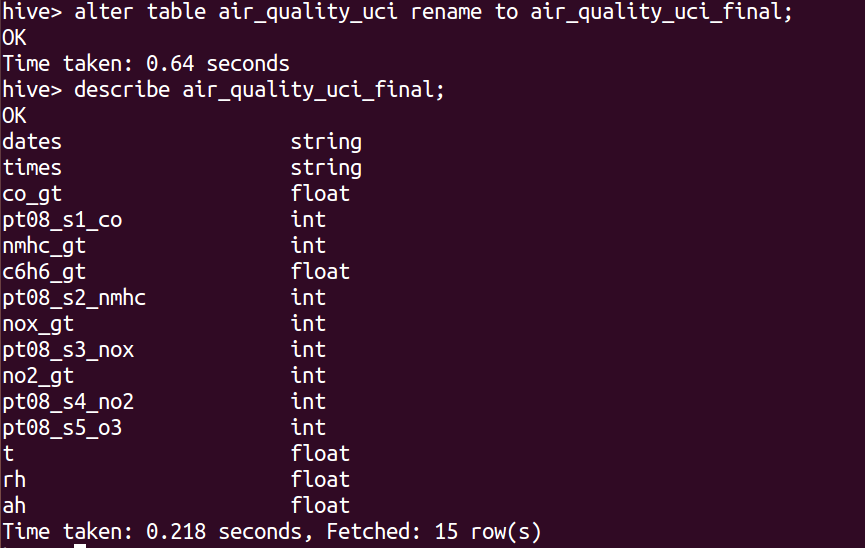
****

hive> select regexp\_replace(dates,':','-') as new\_dates\_style from air\_quality\_uci limit 5;



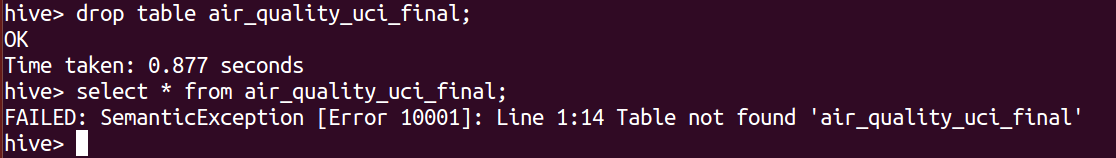
1. **alter table operation**

hive> alter table air\_quality\_uci rename to air\_quality\_uci\_final;

****

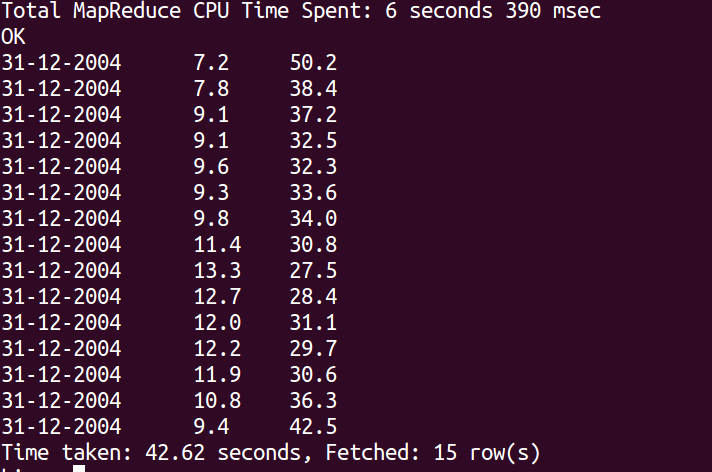
**9 . drop table operation.**

hive> drop table air\_quality\_uci\_final;



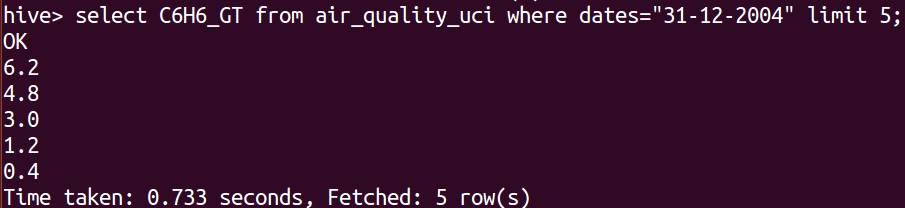
**10. order by operation .**

hive> select dates,t,rh as dates\_t\_rh from air\_quality\_uci order by dates desc limit 15;

****

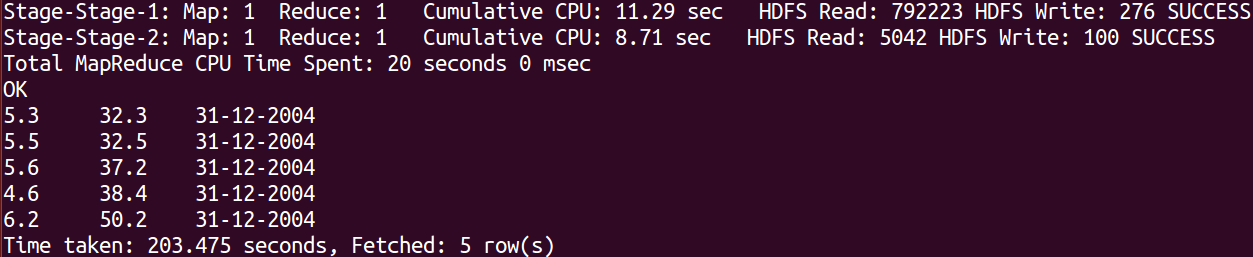
**11 . where clause operations you have to perform.**

hive> select C6H6\_GT from air\_quality\_uci where dates="31-12-2004" limit 5;

****

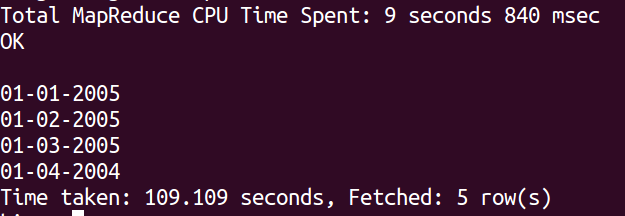
**12 . sorting operation you have to perform .**

hive> select C6H6\_GT,rh,dates from air\_quality\_uci sort by dates desc limit 5;



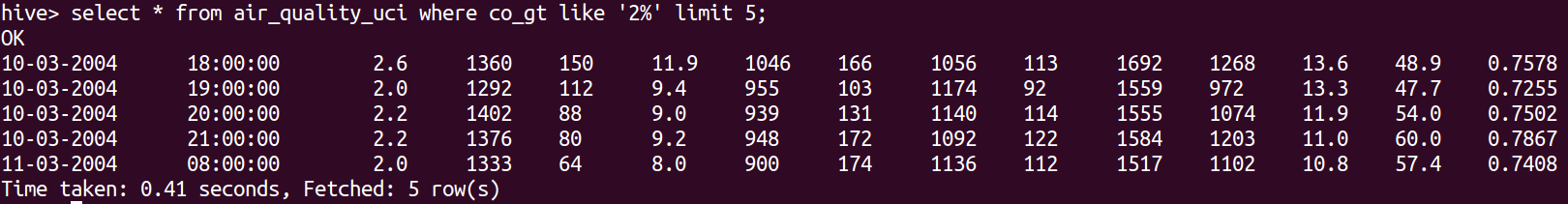
**13 . distinct operation you have to perform .**

hive> select distinct(dates) from air\_quality\_uci limit 5;

****

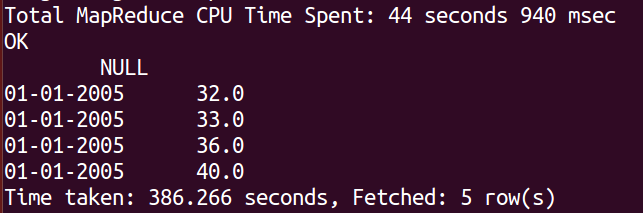
**14 . like an operation you have to perform .**

hive> select \* from air\_quality\_uci where co\_gt like '2%' limit 5;



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

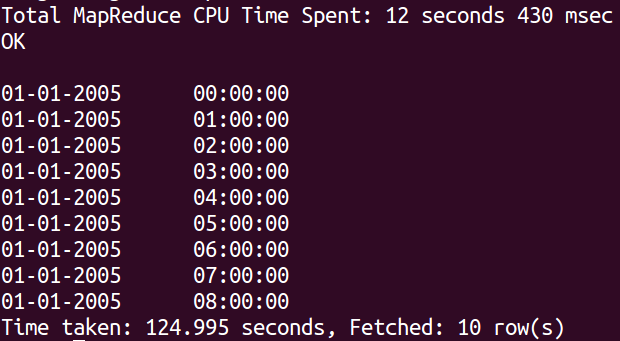
hive> select distinct(dates),round(t) as table\_1 from air\_quality\_uci union all select distinct(dates),round(rh) as table\_2 from air\_quality\_uci limit 5;

****

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

hive> create view table\_view as select distinct(dates),times from air\_quality\_uci;

hive> select \* from table\_view limit 10;



----------------------------------------------------------------------------------------------

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