Problem Statement 1: Jimmy, from the healthcare department, has requested a report that shows how the number of treatments each age category of patients has gone through in the year 2022.

The age category is as follows, Children (00-14 years), Youth (15-24 years), Adults (25-64 years), and Seniors (65 years and over).

Assist Jimmy in generating the report.

Query:

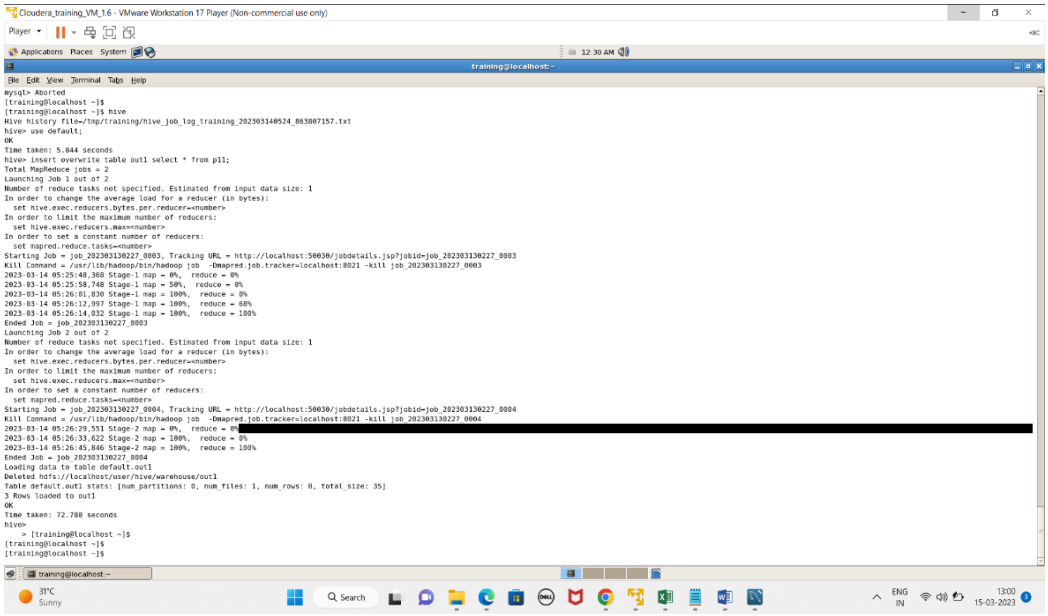
```
> create view p11 as select e.category, count(*) as count from (select (case when DATEDIFF("2022-12-01",p.dob) / 365.25 <=14 then "children"
 when DATEDIFF("2022-12-01",p.dob) / 365.25  <=24 then "youth"
 when datediff("2022-12-01",p.dob) <= 64  then "Adults"
 else "Seniors"
  end) as category from treatment t join patient p on t.patientID=p.patientID where year(t.`date`)=2022) e group by e.category;
```

Output:

```
hive>
   > select * from p11;
Seniors 2105
children            780
youth    82
```

External Table:

```
hive> create external table out1(category string, count int);
OK
Time taken: 1.499 seconds
hive> insert overwrite table out1 select * from p11;
```



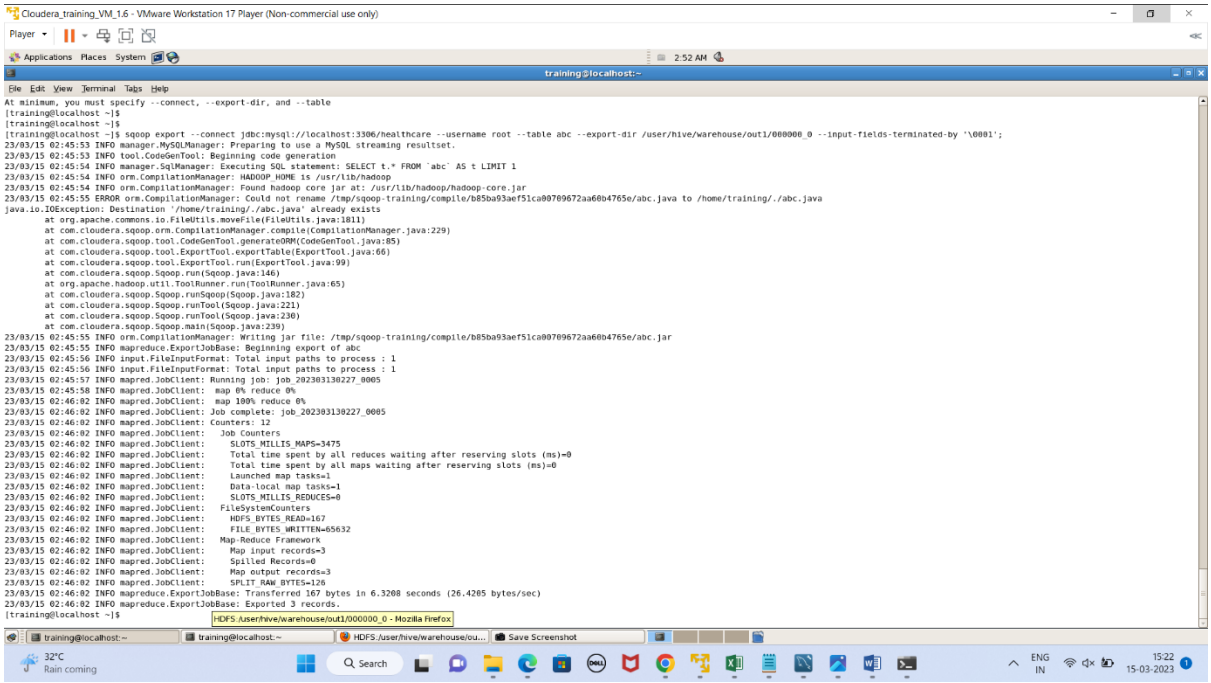
Creating empty table in Mysql:

mysql> create table abc(category varchar(20), count int);

Query OK, 0 rows affected (0.01 sec)

Sqoop Export:

[training@localhost ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root --table abc --export-dir /user/hive/warehouse/out1/000000_0 --input-fields-terminated-by '\0001';



mysql> select * from abc;

+----------+-------+

| category | count |

+----------+-------+

| Seniors | 2105 |

| children | 780 |

| youth | 82 |

+----------+-------+

3 rows in set (0.00 sec)

Problem Statement 2: Jimmy, from the healthcare department, wants to know which disease is infecting people of which gender more often.

Assist Jimmy with this purpose by generating a report that shows for each disease the male-to-female ratio. Sort the data in a way that is helpful for Jimmy.

Query:

hive> create view p12 as select d.diseasename, p.gender, count(*) as cnt from disease d join treatment t on d.diseaseid = t.diseaseid join person p on p.personid = t.patientid group by d.diseasename, p.gender order by cnt;

OK

Time taken: 2.03 seconds

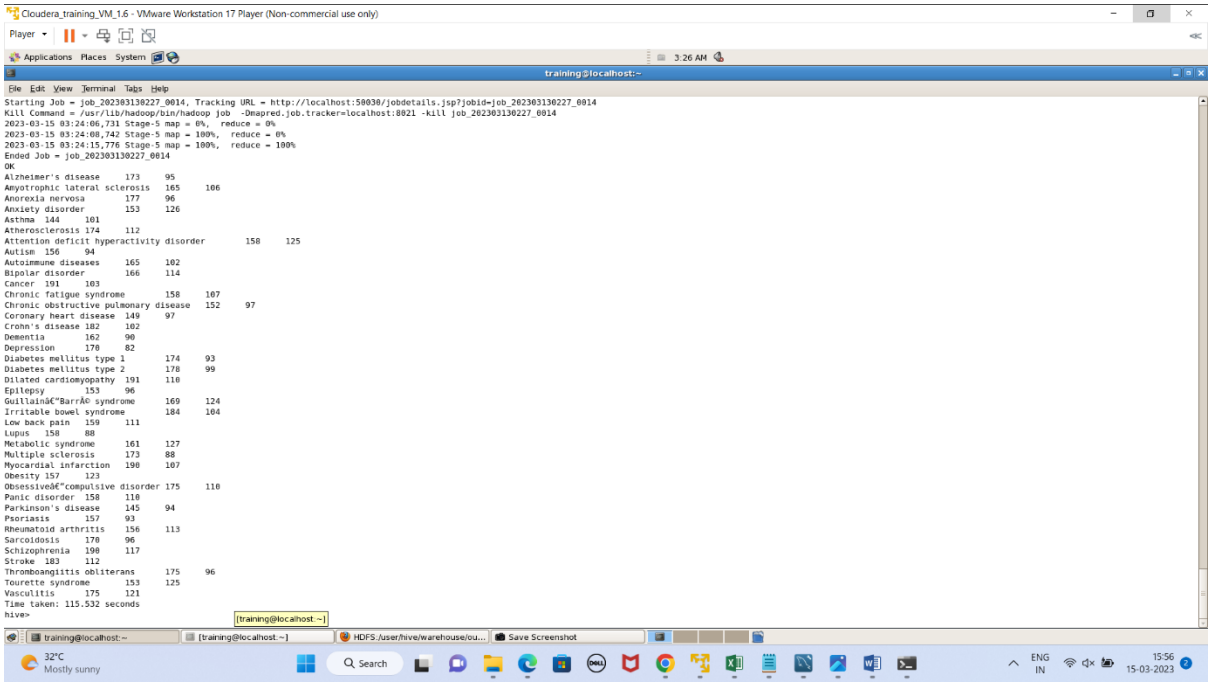
hive> create view p121 as select a.d diseasename, b.cnt male, a.cnt female from (select diseasename d , cnt from p12 where gender = 'female')a join (select diseasename d, cnt from p12 where gender='male')b on a.d=b.d;

OK

Time taken: 0.189 seconds

hive>

> select * from p121;



External Table:

hive>

> create external table out2(category string,female int, male int);

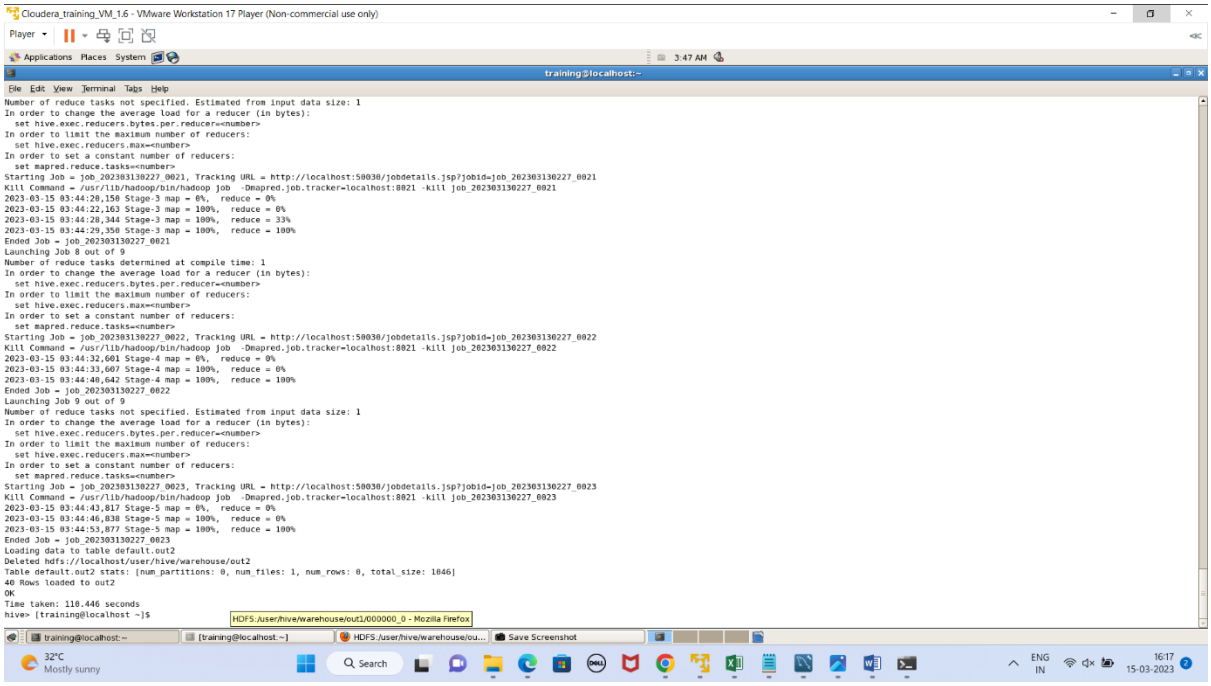
OK

Time taken: 0.048 seconds

hive> insert overwrite table out2 select * from p121;

Total MapReduce jobs = 9

Launching Job 1 out of 9



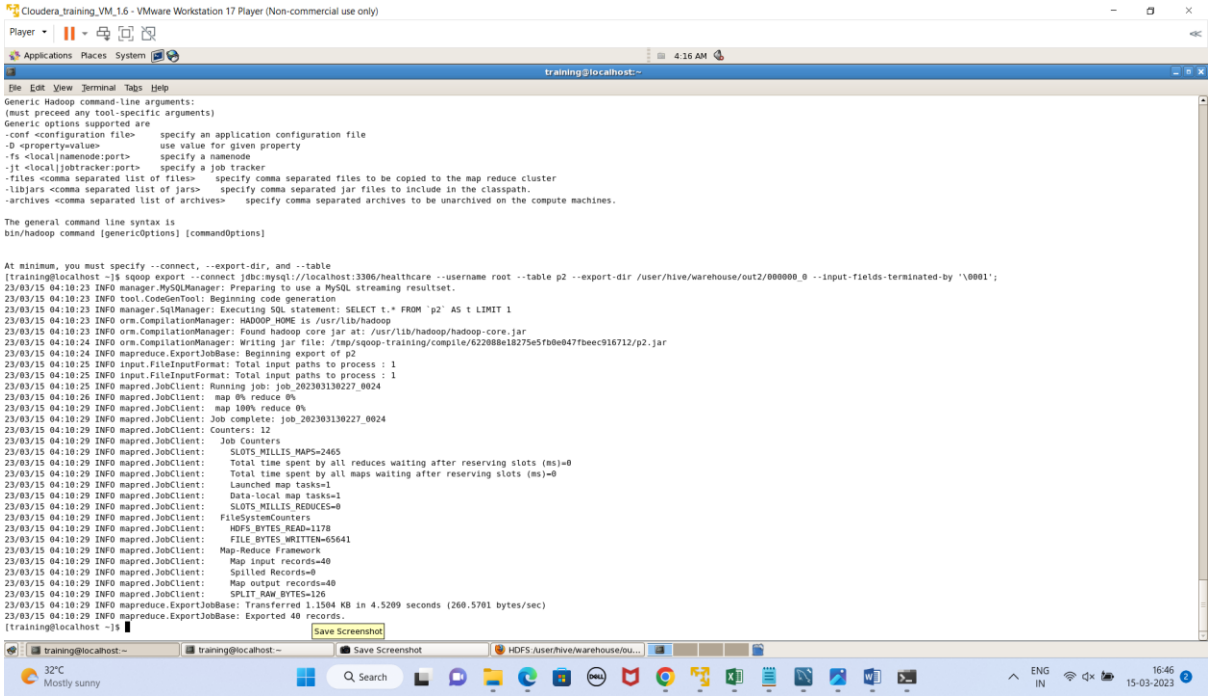
Creating empty table in Mysql:

mysql> create table p2(category varchar(20), female int, male int);

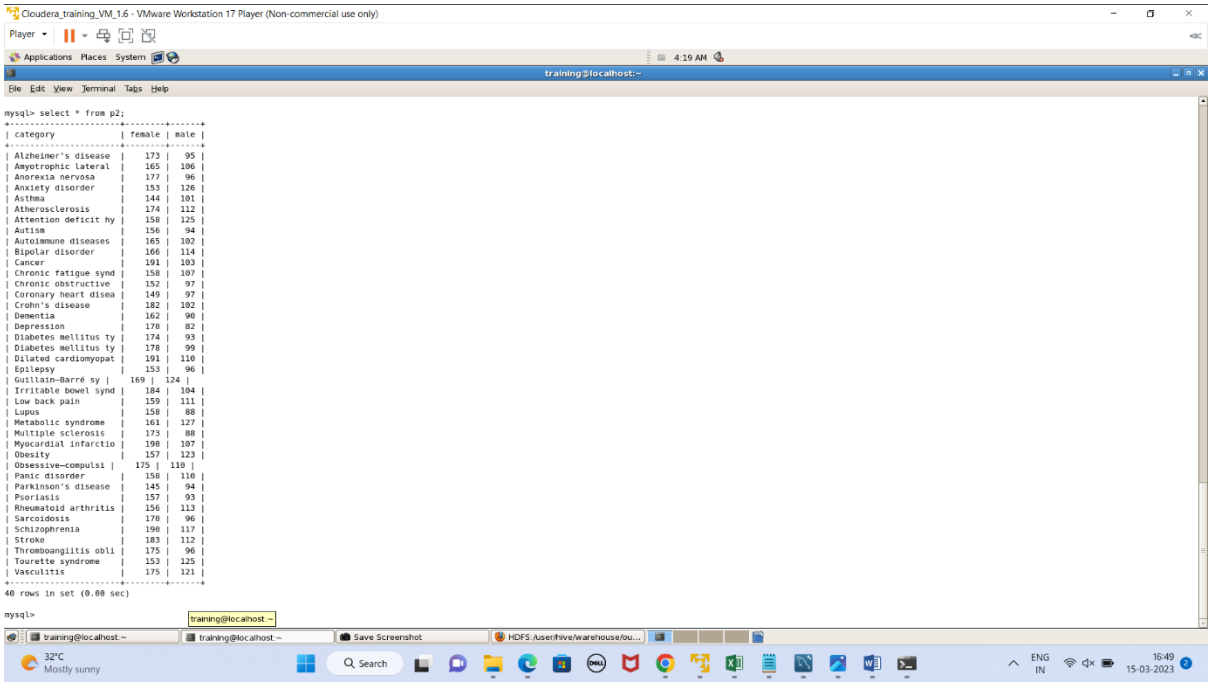
Query OK, 0 rows affected (0.01 sec)

Sqoop Export:

[training@localhost ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root --table p2 --export-dir /user/hive/warehouse/out2/000000_0 --input-fields-terminated-by '\0001';



mysql> select * from p2;

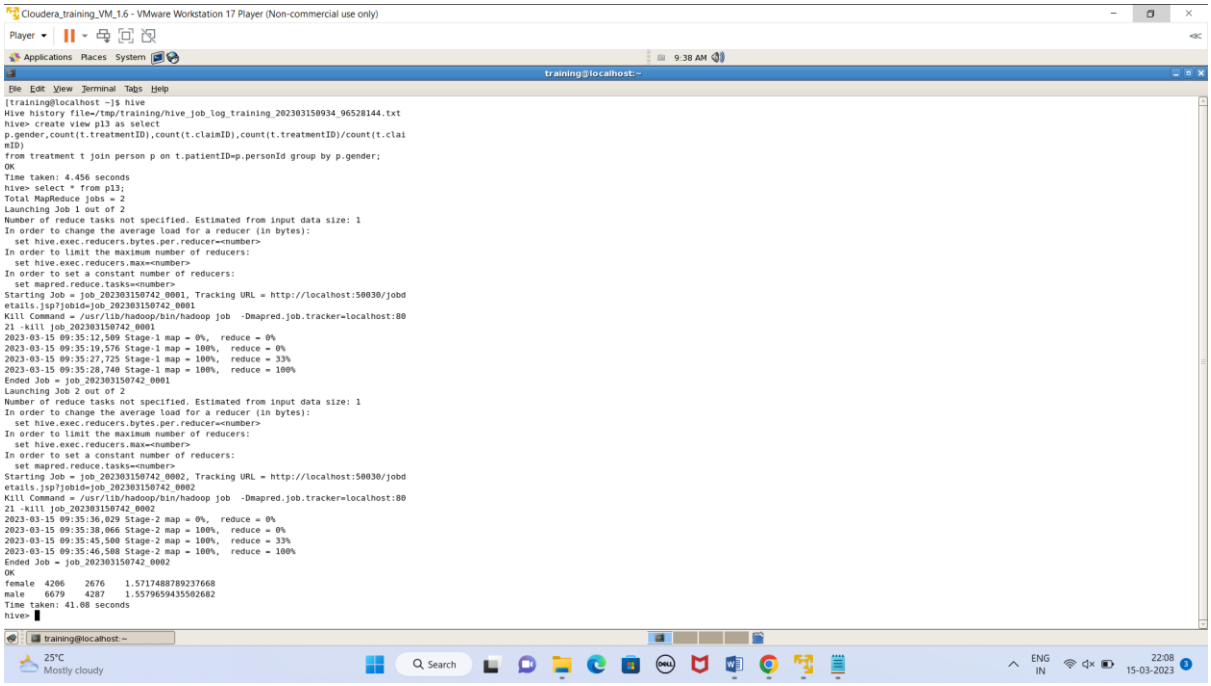


Problem Statement 3: Jacob, from insurance management, has noticed that insurance claims are not made for all the treatments. He also wants to figure out if the gender of the patient has any impact on the insurance claim. Assist Jacob in this situation by generating a report that finds for each gender the number of treatments, number of claims, and treatment-to-claim ratio. And notice if there is a significant difference between the treatment-to-claim ratio of male and female patients.

Query:

hive> select p.gender,count(t.treatmentID),count(t.claimID),count(t.treatmentID)/count(t.claimID) from

treatment t join person p on t.patientID=p.personID group by p.gender;

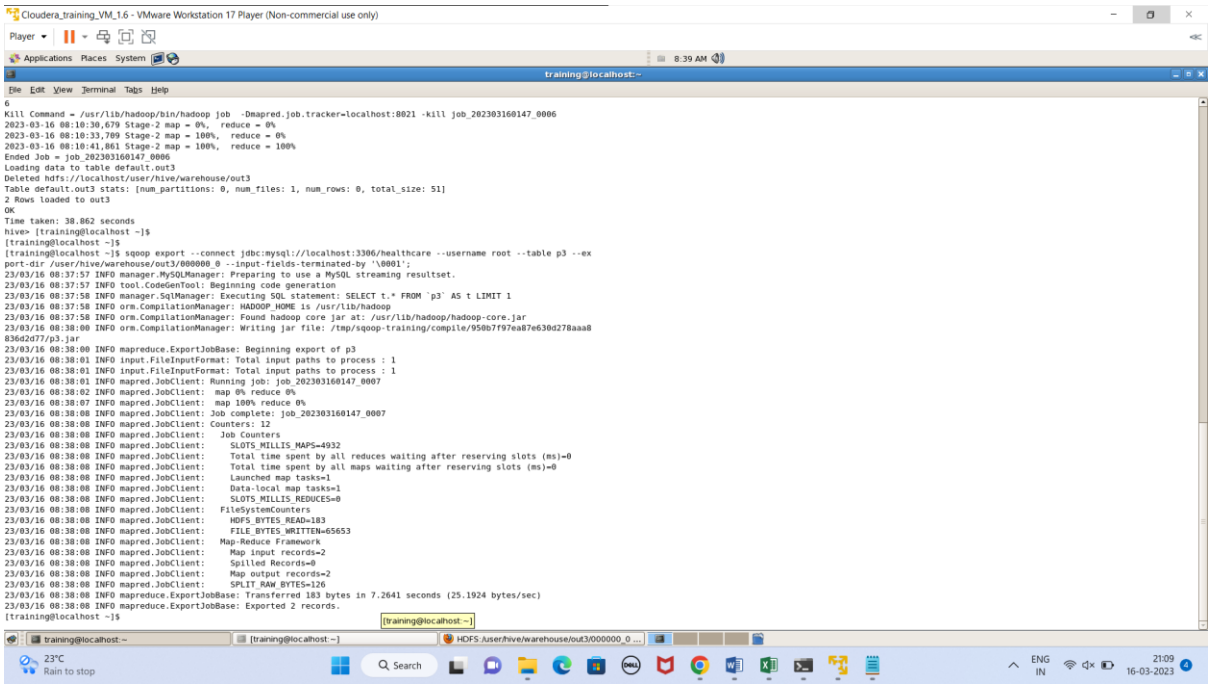


mysql> create table p3(chatagory varchar(20),t_count int,c_count int,t_to_c_ratio decimal);

Query OK, 0 rows affected (0.01 sec)

Sqoop Export:

[training@localhost ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root --table p3 --export-dir /user/hive/warehouse/out3/000000_0 --input-fields-terminated-by '\0001';



Problem Statement 4: The Healthcare department wants a report about the inventory of pharmacies. Generate a report on their behalf that shows how many units of medicine each pharmacy has in their inventory, the total maximum retail price of those medicines, and the total price of all the medicines after discount.

Note: discount field in keep signifies the percentage of discount on the maximum price.

Query:

hive>

> select a.pid as PharmacyID,sum(a.total),sum(a.after_discount) from (select k.pharmacyid as pid,(k.quantity*m.maxprice) as total,((k.quantity*m.maxprice)-((k.quantity*m.maxprice)*k.discount/100)) as after_discount from pharmacy p join keep k on k.pharmacyid=p.pharmacyid join medicine m on m.medicineid=k.medicineid)a group by a.pid;

External Table:

hive> create external table out4(pharmacyid int,total_amount double,total_amount_after_discount double);

OK

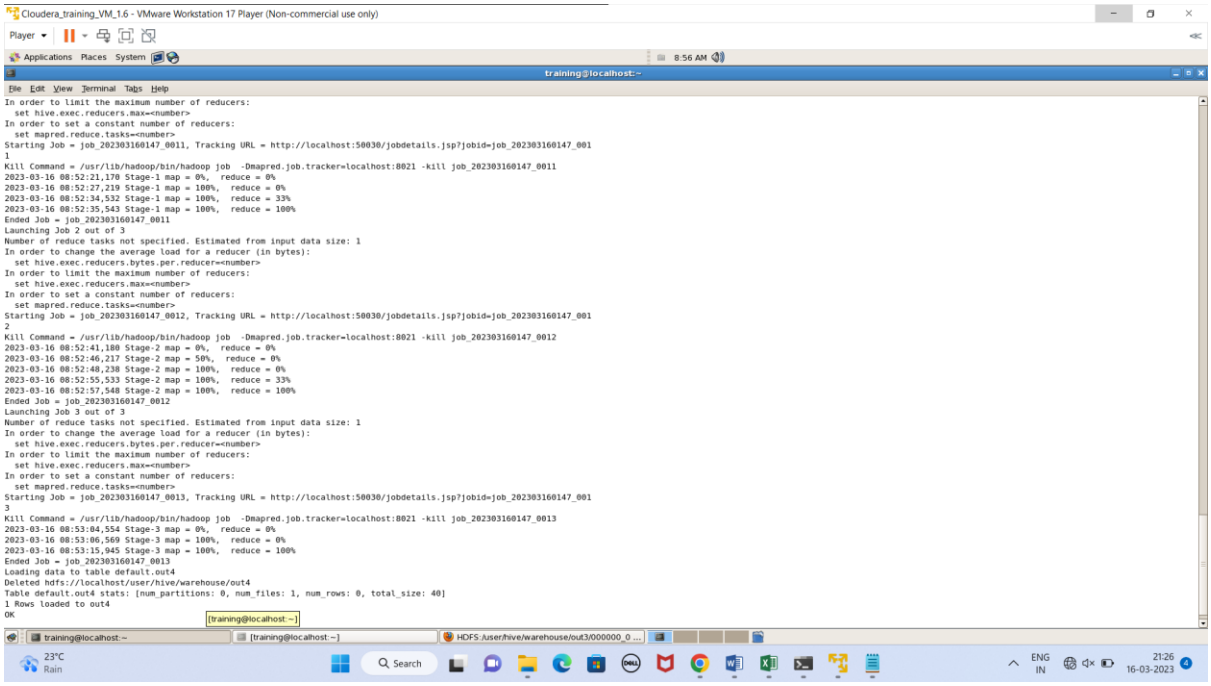
Time taken: 0.344 seconds

hive> insert overwrite table out4 select a.pid as PharmacyID,sum(a.total),sum(a.after_discount)

from (select k.pharmacyid as pid,(k.quantity*m.maxprice) as total,((k.quantity*m.maxprice)-

((k.quantity*m.maxprice)*k.discount/100)) as after_discount from pharmacy p join keep k on

k.pharmacyid=p.pharmacyid join medicine m on m.medicineid=k.medicineid)a group by a.pid;



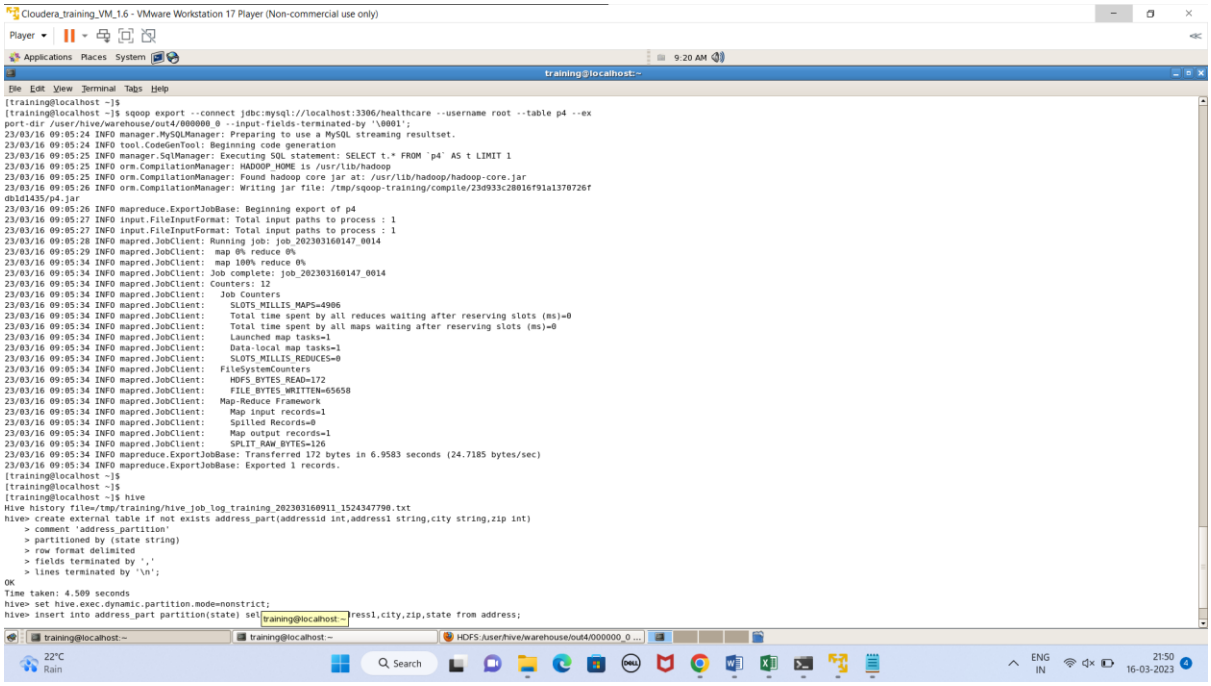
mysql> create table p4(pharmacyid int,total_amount_after_discount double);

Query OK, 0 rows affected (0.00 sec)

Sqoop Export:

[training@localhost ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root --table p4 --export-dir /user/hive/warehouse/out4/000000_0 --input-fields-terminated-by '\0001';

Creating partition table on address:



Problem Statement 5: The healthcare department wants a state-wise health report to assess which state requires more attention in the healthcare sector. Generate a report for them that shows the state name, number of registered people in the state, number of registered patients in the state, and the people-to-patient ratio. sort the data by people-to-patient ratio.

Query:

hive>

>

> create view patientt as select a.state as state,count(pa.patientid) as patient_count from address a

join person pe on pe.addressid=a.addressid join patient pa on pa.patientid=pe.personid group by

a.state;

OK

Time taken: 0.374 seconds

hive> create view personn as select a.state as statee,count(pe.personid) as person_count from address a

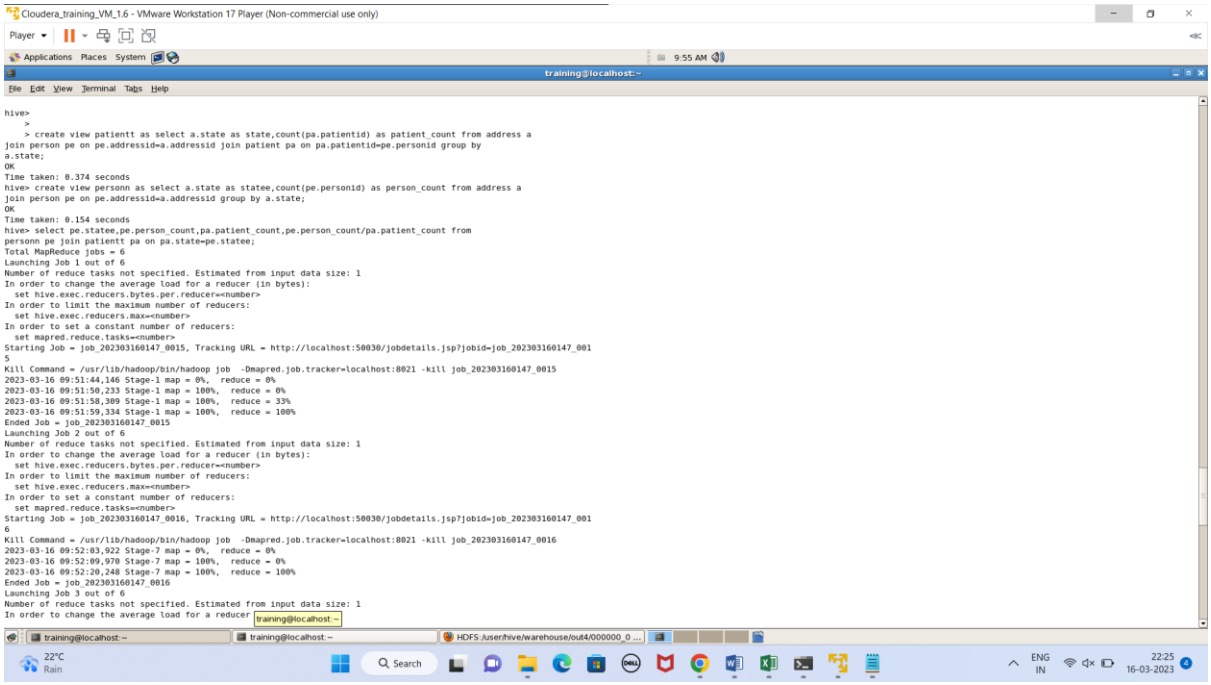
join person pe on pe.addressid=a.addressid group by a.state;

OK

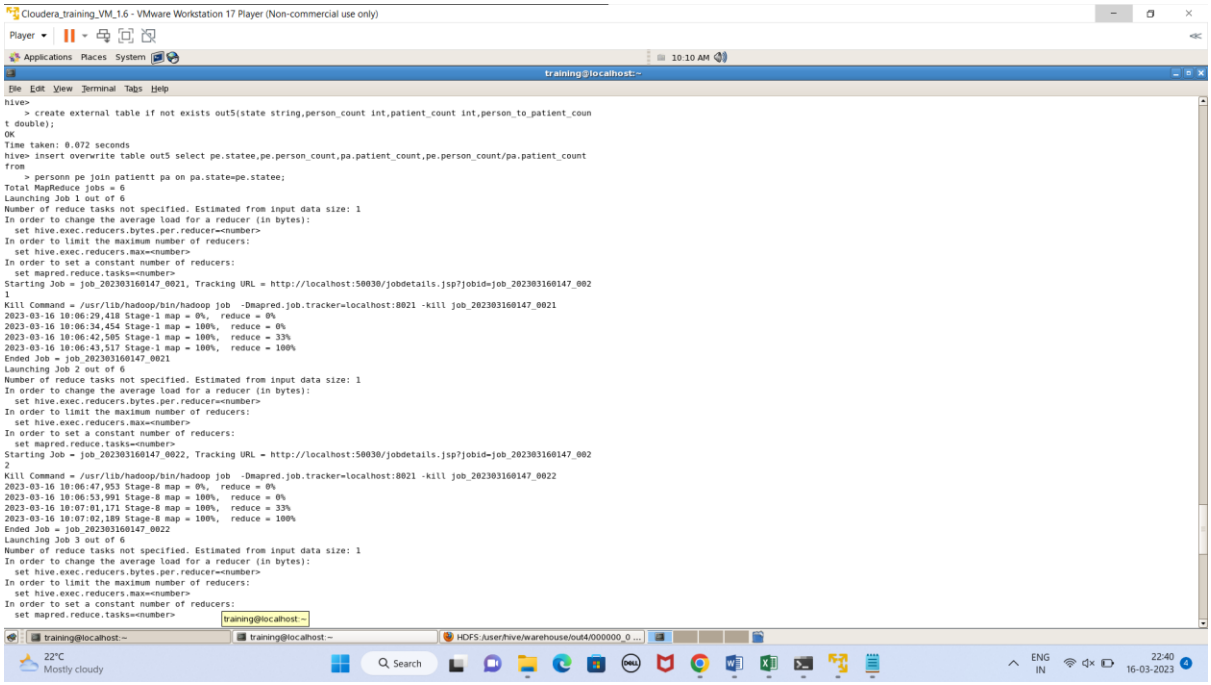
Time taken: 0.154 seconds

hive> select pe.statee,pe.person_count,pa.patient_count,pe.person_count/pa.patient_count from

personn pe join patientt pa on pa.state=pe.statee;



CREATING EXTERNAL TABLE:



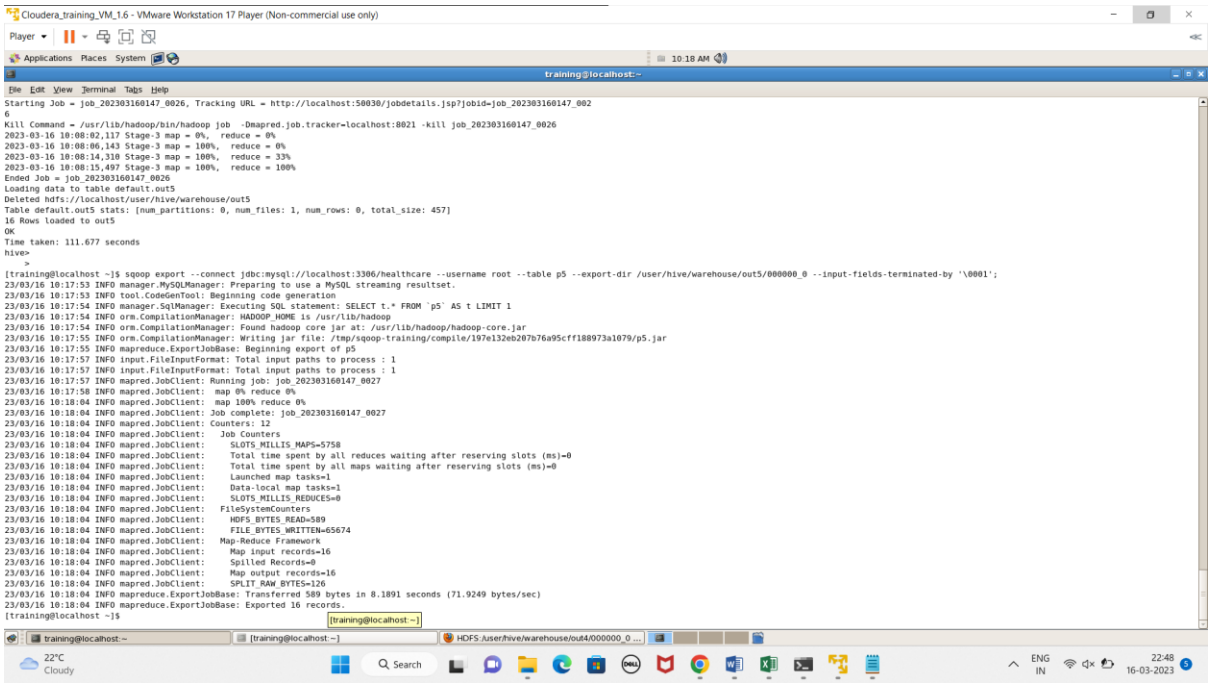
Export the data to SQL Database:

mysql> create table p5(state varchar(20),person_count int,patient_count int,person_to_patient_count double);

Query OK, 0 rows affected (0.00 sec)

Sqoop Export:

[training@localhost ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root --table p5 --export-dir /user/hive/warehouse/out5/000000_0 --input-fields-terminated-by '\0001';



Problem statement -6: Johansson is trying to prepare a report on patients who have gone through treatments more than once. Help Johansson prepare a report that shows the patient's name, the number of treatments they have undergone, and their age, Sort the data in a way that the patients who have undergone more treatments appear on top.

Query:

SELECT P.PERSONNAME as PERSONNAME,X.CNT as TREATMENTCOUNT,cast(datediff('2023-03-14',PA.DOB)/365 as int) as AGE FROM

(select T.PATIENTID as PATIENTID,COUNT(t.TREATMENTID) as CNT FROM TREATMENT T join PATIENT P on P.PATIENTID=T.PATIENTID

GROUP BY T.PATIENTID HAVING COUNT(t.TREATMENTID)>1 ORDER BY 2)X join Patient PA on PA.PATIENTID=X.PATIENTID join Person P on P.PERSONID=PA.PATIENTID ORDER BY 2 DESC;

```
   > SELECT P.PERSONNAME as PERSONNAME,X.CNT as TREATMENTCOUNT,cast(datediff('2023-03-14',PA.DOB)/365 as int) as AGE FROM
   > (select T.PATIENTID as PATIENTID,COUNT(t.TREATMENTID) as CNT FROM TREATMENT T join PATIENT P on P.PATIENTID=T.PATIENTID
   > GROUP BY T.PATIENTID HAVING COUNT(t.TREATMENTID)>1 ORDER BY 2)X join Patient PA on PA.PATIENTID=X.PATIENTID join Person P on P.PERSONID=PA.PATIENTID ORDER BY 2 DESC;
uery ID = cloudera_20230314112121_0e183c77-8971-4dbd-b742-bc09a63715a5
otal jobs = 3
xecution log at: /tmp/cloudera/cloudera_20230314112121_0e183c77-8971-4dbd-b742-bc09a63715a5.log
023-03-14 11:21:34     Starting to launch local task to process map join;      maximum memory = 1013645312
023-03-14 11:21:37     Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/3771d813-6edb-4720-ae8e-aa47a7a22fdc/hive_2023-03-14_11-21-25_659_239433
file31--.hashtable
023-03-14 11:21:37     Uploaded 1 File to: file:/tmp/cloudera/3771d813-6edb-4720-ae8e-aa47a7a22fdc/hive_2023-03-14_11-21-25_659_2394331964551247661-1/-local-10010/HashTable-Stage-
023-03-14 11:21:37     End of local task; Time Taken: 3.207 sec.
xecution completed successfully
lapredLocal task succeeded
aunching Job 1 out of 3
umber of reduce tasks not specified. Estimated from input data size: 1
n order to change the average load for a reducer (in bytes):
 set hive.exec.reducers.bytes.per.reducer=<number>
n order to limit the maximum number of reducers:
 set hive.exec.reducers.max=<number>
n order to set a constant number of reducers:
 set mapreduce.job.reduces=<number>
tarting Job = job_1678815864957_0004, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678815864957_0004/
ill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678815864957_0004
adoop job information for Stage-2: number of mappers: 1; number of reducers: 1
023-03-14 11:21:51,685 Stage-2 map = 0%,  reduce = 0%
023-03-14 11:22:05,983 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 3.33 sec
023-03-14 11:22:25,098 Stage-2 map = 100%,  reduce = 67%, Cumulative CPU 6.08 sec
023-03-14 11:22:40,429 Stage-2 map = 100%,  reduce = 100%, Cumulative CPU 12.92 sec
lapReduce Total cumulative CPU time: 12 seconds 920 msec
nded Job = job_1678815864957_0004
aunching Job 2 out of 3
umber of reduce tasks determined at compile time: 1
n order to change the average load for a reducer (in bytes):
```

CREATING EXTERNAL TABLE TO STORE OUTPUT:

```
hive> create external table out6(Name string,count int,age int);
OK
Time taken: 13.077 seconds
hive> insert overwrite table out6 SELECT P.PERSONNAME as PERSONNAME,X.CNT as TREATMENTCOUNT,cast(datediff('2023-03-14',PA.DOB)/365 as int) as AGE FROM
    > (select T.PATIENTID as PATIENTID,COUNT(t.TREATMENTID) as CNT FROM TREATMENT T join PATIENT P on P.PATIENTID=T.PATIENTID
    > GROUP BY T.PATIENTID HAVING COUNT(t.TREATMENTID)>1 ORDER BY 2)X join Patient PA on PA.PATIENTID=X.PATIENTID join Person P on P.PERSONID=PA.PATIENTID ORDER BY TREATMENTCOUNT DESC;
Query ID = cloudera_20230314115151_6e250901-e9e9-416a-b7d8-4182aa493b8a
Total jobs = 3
Execution log at: /tmp/cloudera/cloudera_20230314115151_6e250901-e9e9-416a-b7d8-4182aa493b8a.log
2023-03-14 11:52:04     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-14 11:52:19     Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/3771d813-6edb-4720-ae8e-aa47a7a22fdc/hive_2023-03-14_11-51-29_146_3073198751
pfile71--.hashtable
2023-03-14 11:52:20     Uploaded 1 File to: file:/tmp/cloudera/3771d813-6edb-4720-ae8e-aa47a7a22fdc/hive_2023-03-14_11-51-29_146_3073198751630529297-1/-local-10008/HashTable-Stage-2/Ma
2023-03-14 11:52:20     End of local task; Time Taken: 15.634 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 3
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_1678815864957_0010, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678815864957_0010/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678815864957_0010
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2023-03-14 11:53:24,287 Stage-2 map = 0%,  reduce = 0%
2023-03-14 11:54:25,132 Stage-2 map = 0%,  reduce = 0%, Cumulative CPU 5.9 sec
2023-03-14 11:54:28,105 Stage-2 map = 100%,  reduce = 0%, Cumulative CPU 10.75 sec
```

Problem Statement 7: Jhonny, from the finance department of Arizona(AZ), has requested areport that lists the total quantity of medicine each pharmacy in his state has prescribed that falls under **Tax criteria I** for treatments that took place in 2021. Assist Jhonny in generating the report.

Query:

SELECT X.STATE AS STATE,X.PHARMACYID AS PHARMACYID ,Y.QNT AS QUANTITY FROM (SELECT A.STATE AS STATE,P.PHARMACYID AS PHARMACYID FROM PHARMACY P JOIN ADDRESS_PART A ON A.ADDRESSID=P.ADDRESSID)X join (SELECT P.PHARMACYID AS PHARMACYID,SUM(C.QUANTITY) AS QNT FROM PRESCRIPTION PR JOIN CONTAIN C ON PR.PRESCRIPTIONID=C.PRESCRIPTIONID JOIN MEDICINE M ON C.MEDICINEID=M.MEDICINEID JOINTREATMENT T ON T.TREATMENTID=PR.TREATMENTID JOIN PHARMACY P ON PR.PHARMACYID=P.PHARMACYID WHERE YEAR(T.`DATE`)=2021 AND M.TAXCRITERIA='I' GROUP BY P.PHARMACYID ORDER BY P.PHARMACYID)Y ON X.PHARMACYID=Y.PHARMACYID WHERE X.STATE='AZ' ORDER BY QUANTITY;

```
hive> SELECT X.STATE AS STATE,X.PHARMACYID AS PHARMACYID ,Y.QNT AS QUANTITY FROM (SELECT A.STATE AS STATE,P.PHARMACYI
D AS PHARMACYID FROM PHARMACY P JOIN ADDRESS_PART A ON A.ADDRESSID=P.ADDRESSID)X join (SELECT P.PHARMACYID AS PHARMAC
YID,SUM(C.QUANTITY) AS QNT FROM PRESCRIPTION PR JOIN CONTAIN C ON PR.PRESCRIPTIONID=C.PRESCRIPTIONID JOIN MEDICINE M
ON C.MEDICINEID=M.MEDICINEID JOIN TREATMENT T ON T.TREATMENTID=PR.TREATMENTID JOIN PHARMACY P ON PR.PHARMACYID=P.PHAR
MACYID WHERE YEAR(T.`DATE`)=2021 AND M.TAXCRITERIA='I' GROUP BY P.PHARMACYID ORDER BY P.PHARMACYID)Y  ON X.PHARMACYID
=Y.PHARMACYID WHERE X.STATE='AZ' ORDER BY QUANTITY;
Query ID = cloudera_20230315023131_9828b6dd-3e29-491a-9c09-1b47c98374bf
Total jobs = 7
Execution log at: /tmp/cloudera/cloudera_20230315023131_9828b6dd-3e29-491a-9c09-1b47c98374bf.log
2023-03-15 02:32:18     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-15 02:32:22     Dump the side-table for tag: 1 with group count: 159 into file: file:/tmp/cloudera/1c951b48-7
b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-31-54_835_8079902312757180943-1/-local-10016/HashTable-Stage-18/MapJoin
-mapfile21--.hashtable
2023-03-15 02:32:22     Uploaded 1 File to: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_0
2-31-54_835_8079902312757180943-1/-local-10016/HashTable-Stage-18/MapJoin-mapfile21--.hashtable (3617 bytes)
2023-03-15 02:32:22     End of local task; Time Taken: 4.031 sec.
Execution completed successfully
MapredLocal task succeeded
Execution log at: /tmp/cloudera/cloudera_20230315023131_9828b6dd-3e29-491a-9c09-1b47c98374bf.log
2023-03-15 02:32:36     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-15 02:32:43     Dump the side-table for tag: 1 with group count: 213 into file: file:/tmp/cloudera/1c951b48-7
b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-31-54_835_8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoin-
mapfile31--.hashtable
2023-03-15 02:32:43     Uploaded 1 File to: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_0
2-31-54_835_8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoin-mapfile31--.hashtable (4540 bytes)
2023-03-15 02:32:43     Dump the side-table for tag: 1 with group count: 2646 into file: file:/tmp/cloudera/1c951b48-
7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-31-54_835_8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoin
-mapfile41--.hashtable
2023-03-15 02:32:43     Uploaded 1 File to: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_0
2-31-54_835_8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoin-mapfile41--.hashtable (56099 bytes)
2023-03-15 02:32:43     Dump the side-table for tag: 1 with group count: 28646 into file: file:/tmp/cloudera/1c951b48
-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-31-54_835_8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoi
n-mapfile51--.hashtable
2023-03-15 02:32:44     Uploaded 1 File to: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_0
2-31-54_835_8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoin-mapfile51--.hashtable (575851 bytes)
2023-03-15 02:32:44     Dump the side-table for tag: 0 with group count: 13428 into file: file:/tmp/cloudera/1c951b48
-7b84-41f5-884c-ecd1e3871525/hive 2023-03-15 02-31-54 835 8079902312757180943-1/-local-10018/HashTable-Stage-9/MapJoi
```

```
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: :
Total MapReduce CPU Time Spent: 27 seconds 270 msec
OK
AZ      8933    123
AZ      8897    179
AZ      3799    211
AZ      2218    290
AZ      9659    329
AZ      3104    348
AZ      5450    358
AZ      9681    364
AZ      1624    369
AZ      1478    411
AZ      8829    412
AZ      4938    448
AZ      5480    460
AZ      1628    524
AZ      2301    535
AZ      3536    567
AZ      8442    567
Cloudera Live : Welcome! - Cloudera
Live Beginner Tutorial - Mozilla Firefox    hed: 17 row(s)
```

External table:

hive> create external table out7(state string,pharmacyid int,count int) row format delimited fieldsterminated by "," lines terminated by "\n";

Time taken: 0.178 seconds

INSERT DATA INTO EXTERNAL TABLE:

```
hive> create external table out7(state string,pharmacyid int,count int) row format delimited fields terminated by "," lines terminated by "\n";
OK
Time taken: 0.178 seconds
hive> insert overwrite table out7 SELECT X.STATE AS STATE,X.PHARMACYID AS PHARMACYID ,Y.QNT AS QUANTITY FROM (SELECT A.STATE AS STATE,P.PHARMACYID AS PHARMACYID FROM PHARMACY P JOIN ADDRESS_PART A ON A.ADDRESSID=P.ADDRESSID)X join (SELEC
T P.PHARMACYID AS PHARMACYID,SUM(C.QUANTITY) AS QNT FROM PRESCRIPTION PR JOIN CONTAIN C ON PR.PRESCRIPTIONID=C.PRESCRIPTIONID JOIN MEDICINE M ON C.MEDICINEID=M.MEDICINEID JOIN TREATMENT T ON T.TREATMENTID=PR.TREATMENTID JOIN PHARMACY P O
N PR.PHARMACYID=P.PHARMACYID WHERE YEAR(T.`DATE`)=2021 AND M.TAXCRITERIA='I' GROUP BY P.PHARMACYID ORDER BY P.PHARMACYID)Y  ON X.PHARMACYID=Y.PHARMACYID WHERE X.STATE='AZ' ORDER BY QUANTITY;
Query ID = cloudera_20230315024646_a93e278e-d729-4c79-95b7-610ba77a1dc2
Total jobs = 7
Execution log at: /tmp/cloudera/cloudera_20230315024646_a93e278e-d729-4c79-95b7-610ba77a1dc2.log
2023-03-15 02:46:50     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-15 02:46:54     Dump the side-table for tag: 1 with group count: 159 into file: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-46-35_681_4308551305438668496-1/-local-10014/HashTable-Stage-19/MapJoin-ma
pfile91--.hashtable
2023-03-15 02:46:54     Uploaded 1 File to: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-46-35_681_4308551305438668496-1/-local-10014/HashTable-Stage-19/MapJoin-mapfile91--.hashtable (3617 bytes)
2023-03-15 02:46:54     End of local task; Time Taken: 3.753 sec.
Execution completed successfully
MapredLocal task succeeded
Execution log at: /tmp/cloudera/cloudera_20230315024646_a93e278e-d729-4c79-95b7-610ba77a1dc2.log
2023-03-15 02:47:07     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-15 02:47:15     Dump the side-table for tag: 1 with group count: 213 into file: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-46-35_681_4308551305438668496-1/-local-10016/HashTable-Stage-10/MapJoin-ma
pfile101--.hashtable
2023-03-15 02:47:15     Uploaded 1 File to: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-46-35_681_4308551305438668496-1/-local-10016/HashTable-Stage-10/MapJoin-mapfile101--.hashtable (4540 bytes)
2023-03-15 02:47:15     Dump the side-table for tag: 1 with group count: 2646 into file: file:/tmp/cloudera/1c951b48-7b84-41f5-884c-ecd1e3871525/hive_2023-03-15_02-46-35_681_4308551305438668496-1/-local-10016/HashTable-Stage-10/MapJoin-m
```

EXPORT THE DATA TO SQL DATABASE:

sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root table p7 export-dir /user/hive/warehouse/out7/000000_0 --input-fields-terminated-by ','

```
        HDFS: Number of write operations=0
Job Counters
        Launched map tasks=4
        Data-local map tasks=4
        Total time spent by all maps in occupied slots (ms)=218103
        Total time spent by all reduces in occupied slots (ms)=0
        Total time spent by all map tasks (ms)=218103
        Total vcore-seconds taken by all map tasks=218103
        Total megabyte-seconds taken by all map tasks=223337472
Map-Reduce Framework
        Map input records=17
        Map output records=17
        Input split bytes=584
        Spilled Records=0
        Failed Shuffles=0
        Merged Map outputs=0
        GC time elapsed (ms)=4087
        CPU time spent (ms)=6670
        Physical memory (bytes) snapshot=509038592
        Virtual memory (bytes) snapshot=6015627264
        Total committed heap usage (bytes)=243531776
File Input Format Counters
        Bytes Read=0
File Output Format Counters
        Bytes Written=0
; 03:11:05 INFO mapreduce.ExportJobBase: Transferred 1.0801 KB in 86.6443 seconds (12.7648 bytes/sec)
; 03:11:05 INFO mapreduce.ExportJobBase: Exported 17 records.
```

Problem Statement 8:

The healthcare department wants a pharmacy report on the percentage of hospital-exclusive medicine prescribed in the year 2022.

Assist the healthcare department to view for each pharmacy, the pharmacy id, pharmacy name, total quantity of medicine prescribed in 2022, total quantity of hospital-exclusive medicine prescribed by the pharmacy in 2022, and the percentage of hospital-exclusive medicine to the total medicine prescribed in 2022.

Order the result in descending order of percentage found.

Query:

select distinct y.ph as pharmacyid,y.pn as pharmacyname,z.cnt2 as HospitalExclusiveMedQuantity,y.cnt1 as MedicineQuantity,cast((z.cnt2/y.cnt1)*100 as int) as Percentage_of_Hospital_exclusive_to_total_medicine from (SELECT p.pharmacyid as ph,p.pharmacyname as pn,count(c.medicineid) as cnt1 FROM PHARMACY P join PRESCRIPTION PR on p.pharmacyid=pr.pharmacyid join CONTAIN C on pr.prescriptionid=c.prescriptionid join MEDICINE M on c.medicineid=m.medicineid join treatment t on t.treatmentid=pr.treatmentid where year(t.`date`)=2022 group by p.pharmacyid,p.pharmacyname order by p.pharmacyid)y join (SELECT p.pharmacyid as ph,p.pharmacyname as pn,count(c.medicineid) as cnt2 FROM PHARMACY P join PRESCRIPTION PR on p.pharmacyid=pr.pharmacyid join CONTAIN C on pr.prescriptionid=c.prescriptionid join MEDICINE M on c.medicineid=m.medicineid join treatment t on t.treatmentid=pr.treatmentid where year(t.`date`)=2022 and m.hospitalexclusive="S" group by p.pharmacyid,p.pharmacyname order by p.pharmacyid)z where y.ph=z.ph order by pharmacyid;

CREATE EXTERNAL TABLE:

create external table out8(pharmacyid int,pharmacyName string,HospitalExclusiveQuantity int,TotalMedicineQuantity int,HospitalExclusive_to_TotalQuantity_percentage int) row format delimited fields terminated by "," lines terminated by "\n";

INSERT THE DATA INTO EXTERNAL TABLE:

insert overwrite table out9 select distinct y.ph as pharmacyid,y.pn as pharmacyname,z.cnt2 as HospitalExclusiveMedQuantity,y.cnt1 as MedicineQuantity,cast((z.cnt2/y.cnt1)*100 as int) as Percentage_of_Hospital_exclusive_to_total_medicine from (SELECT p.pharmacyid as ph,p.pharmacyname as pn,count(c.medicineid) as cnt1 FROM PHARMACY P join PRESCRIPTION PR on p.pharmacyid=pr.pharmacyid join CONTAIN C on pr.prescriptionid=c.prescriptionid join MEDICINE M on c.medicineid=m.medicineid join treatment t on t.treatmentid=pr.treatmentid where year(t.`date`)=2022 group by p.pharmacyid,p.pharmacyname order by p.pharmacyid)y join (SELECT p.pharmacyid as ph,p.pharmacyname as pn,count(c.medicineid) as cnt2 FROM PHARMACY P join PRESCRIPTION PR on p.pharmacyid=pr.pharmacyid join CONTAIN C on pr.prescriptionid=c.prescriptionid join MEDICINE M on c.medicineid=m.medicineid join treatment t on t.treatmentid=pr.treatmentid where year(t.`date`)=2022 and m.hospitalexclusive="S" group by p.pharmacyid,p.pharmacyname order by p.pharmacyid)z where y.ph=z.ph order by pharmacyid;

```
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678870392778_0027
Hadoop job information for Stage-9: number of mappers: 1; number of reducers: 1
2023-03-15 06:14:20,757 Stage-9 map = 0%,  reduce = 0%
2023-03-15 06:14:36,156 Stage-9 map = 100%,  reduce = 0%, Cumulative CPU 2.16 sec
2023-03-15 06:14:53,872 Stage-9 map = 100%,  reduce = 100%, Cumulative CPU 5.36 sec
MapReduce Total cumulative CPU time: 5 seconds 360 msec
Ended Job = job_1678870392778_0027
MapReduce Jobs Launched:
Stage-Stage-5: Map: 1  Reduce: 1   Cumulative CPU: 10.23 sec   HDFS Read: 340479 HDFS Write: 7826 SUCCESS
Stage-Stage-18: Map: 1  Reduce: 1   Cumulative CPU: 9.19 sec   HDFS Read: 340811 HDFS Write: 4609 SUCCESS
Stage-Stage-6: Map: 1  Reduce: 1   Cumulative CPU: 4.97 sec   HDFS Read: 12195 HDFS Write: 7826 SUCCESS
Stage-Stage-19: Map: 1  Reduce: 1   Cumulative CPU: 5.04 sec   HDFS Read: 8587 HDFS Write: 4609 SUCCESS
Stage-Stage-24: Map: 1   Cumulative CPU: 5.89 sec   HDFS Read: 13744 HDFS Write: 9763 SUCCESS
Stage-Stage-8: Map: 1  Reduce: 1   Cumulative CPU: 5.18 sec   HDFS Read: 15089 HDFS Write: 9763 SUCCESS
Stage-Stage-9: Map: 1  Reduce: 1   Cumulative CPU: 5.36 sec   HDFS Read: 15572 HDFS Write: 9228 SUCCESS
Total MapReduce CPU Time Spent: 45 seconds 860 msec
OK
1008    MobiMeds        12      66      18.181818181818183
1145    Spot Rx 9       73      12.32876712328767
1149    Modern Health   15      73      20.54794520547945
1194    Foundation Care 9       53      16.9811320754717
1204    Family Drug Mart        9       76      11.842105263157894
1248    New Era 7       45      15.555555555555555
1293    Fry?s Pharmacy  19      98      19.387755102040817
1332    Rite Aid        16      87      18.39080459770115
1354    HealthMart      10      54      18.51851851851852
1386    GenScripts      15      82      18.29268292682927
1396    Sand Point Pharmacy     12      78      15.384615384615385
1478    Pocketpills     14      82      17.073170731707318
1570    White Pigeon Pharmacy   10      71      14.084507042253522
1574    ScriptSite Specialty    14      71      19.718309859154928
1584    Sunwest 14      85      16.470588235294116
1609    The Compounding Pharmacy        11      68      16.176470588235293
1624    Caremark        14      70      20.0
1628    IDL Drug Stores 11      54      20.37037037037037
1724    Mediserv        14      93      15.053763440860216
1731    Thrifty Way Pharmacy    16      79      20.253164556962027
1755    Health Warehouse        17      82      20.73170731707317
1766    Pharma Best     7       52      13.461538461538462
1795    Pure Life       9       50      18.0
1882    PersonalRX      13      76      17.105263157894736
1891    Welltrack       11      92      11.956521739130435
1925    Everyday Drugs  11      75      14.666666666666666
1987    Healthbest      13      63      20.634920634920633
2060    Prescription Hope       12      67      17.91044776119403
2066    Drug Blend      14      85      16.470588235294116
```

EXPORT DATA TO THE SQL DATABASE:

sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root –table p8 --export-dir /user/hive/warehouse/out8/000000_0 --input-fields-terminated-by','

```
23/03/15 06:51:32 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=566572
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=16225
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=19
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=4
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=214425
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=214425
                Total vcore-seconds taken by all map tasks=214425
                Total megabyte-seconds taken by all map tasks=219571200
        Map-Reduce Framework
                Map input records=213
                Map output records=213
                Input split bytes=666
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=3659
                CPU time spent (ms)=6920
                Physical memory (bytes) snapshot=444801024
                Virtual memory (bytes) snapshot=6015488000
                Total committed heap usage (bytes)=243531776
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
23/03/15 06:51:32 INFO mapreduce.ExportJobBase: Transferred 15.8447 KB in 84.8234 seconds (191.2797 bytes/sec)
23/03/15 06:51:32 INFO mapreduce.ExportJobBase: Exported 213 records.
```

Problem Statement 9:

Manish, from the healthcare department, wants to know how many registered people are registered as patients as well, in each city. Generate a report that shows each city that has 10 or more registered people belonging to it and the number of patients from that city as well as the percentage of the patient with respect to the registered people.

Query:

select x.city as city,y.regpatient as Registered_patient,x.regperson as Registered_person,(y.regpatient/x.regperson)*100 as PERCENTAGE FROM (select a.city as city,count(p.personid) as regperson from address_part a join person p on a.addressid = p.addressid group by city having count(p.personid)>=10)x join (select a.city as city,count(DISTINCT t.patientid) as regpatient from address_part a join person p on a.addressid = p.addressid join treatment t on p.personid=t.patientid group by a.city)y ON X.CITY=Y.CITY ORDER BY city;

```
Ended Job = job_1678703927?8_0013
MapReduce Jobs Launched:
Stage-Stage-2: Map: 1  Reduce: 1   Cumulative CPU: 8.05 sec   HDFS Read: 134970 HDFS Write: 983 SUCCESS
Stage-Stage-8: Map: 1  Reduce: 1   Cumulative CPU: 8.83 sec   HDFS Read: 137079 HDFS Write: 5295 SUCCESS
Stage-Stage-12: Map: 1   Cumulative CPU: 3.73 sec   HDFS Read: 10436 HDFS Write: 1262 SUCCESS
Stage-Stage-4: Map: 1  Reduce: 1   Cumulative CPU: 5.19 sec   HDFS Read: 6363 HDFS Write: 975 SUCCESS
Total MapReduce CPU Time Spent: 25 seconds 800 msec
OK
Anchorage       47      135     34.81481481481482
Annapolis       14      35      40.0
Arvada  57      144     39.5833333333333
Burlington      2       10      20.0
Calhoun 7       10      70.0
Castro Valley   3       10      30.0
Edmond  11      18      61.111111111111114
Fayetteville    63      149     42.28187919463087 5
Fremont 12      26      46.15384615384615
Glen Burnie     10      23      43.47826086956522
Glendale        57      153     37.254901960784316
Hayward 10      21      47.61904761904761
Livermore       5       12      41.66666666666667
Louisville      46      131     35.11450381679389
Lynn Haven      5       12      41.66666666666667
Manchester      63      168     37.5
Montgomery      73      176     41.47727272727273
Nashville       65      130     50.0
Norman  7       18      38.88888888888889
Oakland 12      31      38.70967741935484
Oklahoma City   37      81      45.67901234567901
Panama City     43      95      45.26315789473684
Panama City Beach       12      31      38.70967741935484
Pasadena        7       15      46.666666666666664
Pooler  6       16      37.5
San Leandro     5       13      38.46153846153847
Savannah        48      132     36.36363636363637
Severn  5       13      38.46153846153847
Severna Park    3       13      23.076923076923077
Union City      4       11      36.36363636363637
Washington      71      184     38.58695652173913
Time taken: 323.074 seconds, Fetched: 31 row(s)
```

creating external table and insert data into the table:

hive> create external table if not exists out9(city string,registered_patient int,registered_person int,percentage double) row format delimited fields terminated by "," lines terminated by "\n";

OK

Time taken: 0.097 seconds

hive> select x.city as city,y.regpatient as Registered_patient,x.regperson as Registered_person,(y.regpatient/x.regperson)*100 as PERCENTAGE FROM (select a.city as city,count(p.personid) as regperson from address_part a join person p on a.addressid = p.addressid group by city having count(p.personid)>=10)x join (select a.city as city,count(DISTINCT t.patientid) as regpatient from address_part a join person p on a.addressid = p.addressid join treatment t on p.personid=t.patientid group by a.city)y ON X.CITY=Y.CITY ORDER BY city;

Export the data to the SQL database:

sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root –table p9 –export-dir /user/hive/warehouse/out9/000000_0 –input-fields-terminated-by’,’

```
23/03/15 04:53:38 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=566304
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2476
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=16
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=4
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=78476
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=78476
                Total vcore-seconds taken by all map tasks=78476
                Total megabyte-seconds taken by all map tasks=80359424
        Map-Reduce Framework
                Map input records=31
                Map output records=31
                Input split bytes=584
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=1233
                CPU time spent (ms)=3070
                Physical memory (bytes) snapshot=470683648
                Virtual memory (bytes) snapshot=6015369216
                Total committed heap usage (bytes)=243531776
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
23/03/15 04:53:38 INFO mapreduce.ExportJobBas                                    in 38.1802 seconds (64.8504 bytes/sec)
```

Cloudera Live : Welcome! - Cloudera ExportJobBas Task Manager

Problem Statement 10:

Mack, From HealthDirect Pharmacy, wants to get a list of all the affordable and costly, hospital-exclusive medicines in the database. Where affordable medicines are the medicines that have a maximum price of less than 50% of the avg maximum price of all the medicines in the database, and costly medicines are the medicines that have a maximum price of more than double the avg maximum price of all the medicines in the database. Mack wants clear text next to each medicine name to be displayed that identifies the medicine as affordable or costly. The medicines that do not fall under either of the two categories need not be displayed.

Write a SQL query for Mack for this requirement.

Query:

select m.medicineid,sum(k.quantity) as Quantity ,"High Quantity" as Quantity_Category,"Low" as Discount from pharmacy p join keep k on k.pharmacyid=p.pharmacyid join medicine m on m.medicineid=k.medicineid where P.pharmacyname="Spot Rx" and k.discount=0 group by m.medicineid having sum(k.quantity)>7500 union all select m.medicineid,sum(k.quantity) as Quantity,"low Quantity" as Quantity_Category,"High" as Discount from pharmacy p join keep k on k.pharmacyid=p.pharmacyid join medicine m on m.medicineid=k.medicineid where P.pharmacyname="Spot Rx" and k.discount>=30 group by m.medicineid having sum(k.quantity)<1000;

```
Starting Job = job_1678870392778_0038, Tracking URL = http://quickstart.cloudera:8088/proxy/application_16
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678870392778_0038
Hadoop job information for Stage-4: number of mappers: 2; number of reducers: 0
2023-03-15 09:43:49,559 Stage-4 map = 0%,  reduce = 0%
2023-03-15 09:44:22,840 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 5.96 sec
MapReduce Total cumulative CPU time: 5 seconds 960 msec
Ended Job = job_1678870392778_0038
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 9.47 sec   HDFS Read: 1035515 HDFS Write: 998 SUCCESS
Stage-Stage-9: Map: 1  Reduce: 1   Cumulative CPU: 9.44 sec   HDFS Read: 1035633 HDFS Write: 463 SUCCESS
Stage-Stage-4: Map: 2   Cumulative CPU: 5.96 sec   HDFS Read: 7365 HDFS Write: 881 SUCCESS
Total MapReduce CPU Time Spent: 24 seconds 870 msec
OK
807     8575    High Quantity   Low
2791    8924    High Quantity   Low
5529    8474    High Quantity   Low
9192    8512    High Quantity   Low
9530    9994    High Quantity   Low
15999   7790    High Quantity   Low
17172   7504    High Quantity   Low
19571   7756    High Quantity   Low
25319   8821    High Quantity   Low
26749   7835    High Quantity   Low
31111   9810    High Quantity   Low
32313   9495    High Quantity   Low
35997   7853    High Quantity   Low
36453   9185    High Quantity   Low
37372   9939    High Quantity   Low
39816   7664    High Quantity   Low
41404   7560    High Quantity   Low
43387   9611    High Quantity   Low
43598   8327    High Quantity   Low
50031   8094    High Quantity   Low
50220   8939    High Quantity   Low
53209   7618    High Quantity   Low
8237    384     low Quantity    High
14240   746     low Quantity    High
15687   680     low Quantity    High
20038   721     low Quantity    High
23972   568     low Quantity    High
25132   321     low Quantity    High
38118   586     low Quantity    High
39536   493     low Quantity    High
41511   8       low Quantity    High
Time taken: 293.3 seconds, Fetched: 31 row(s)
```

Create External table to store the output data:

create external table if not exists out10(medicineid int,Quantity int,Quantity_Category string,discount string)

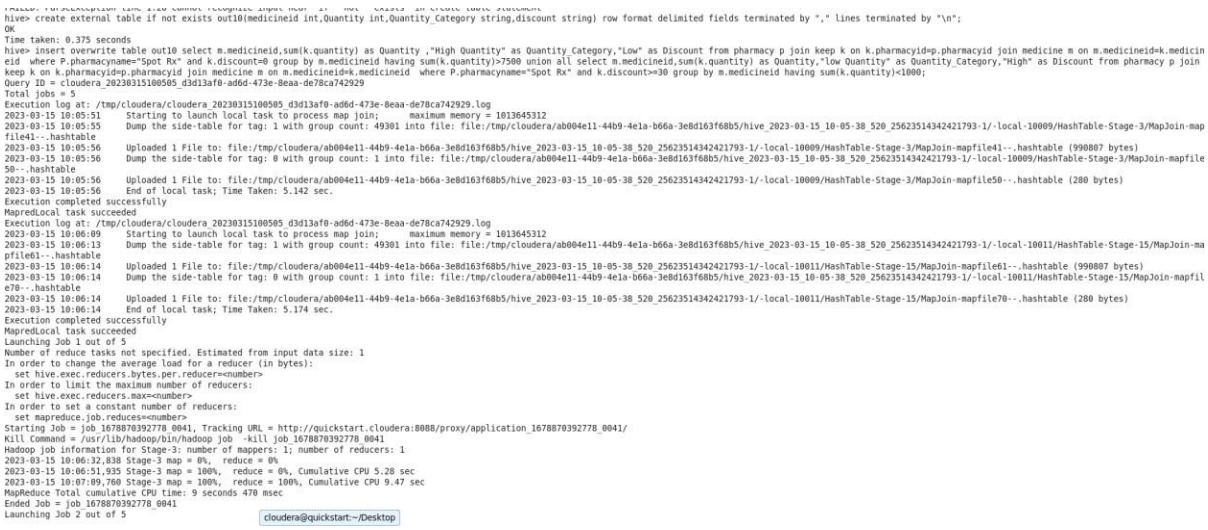
row format delimited

fields terminated by ","

lines terminated by "\n";

INSERT THE OUTPUT DATA INTO EXTERNAL TABLE:

Insert overwrite table out10 select m.medicineid,sum(k.quantity) as Quantity ,"High Quantity" as Quantity_Category,"Low" as Discount from pharmacy p join keep k on k.pharmacyid=p.pharmacyid join medicine m on m.medicineid=k.medicineid where P.pharmacyname="Spot Rx" and k.discount=0 group by m.medicineid having sum(k.quantity)>7500 union all select m.medicineid,sum(k.quantity) as Quantity,"low Quantity"as Quantity_Category,"High" as Discount from pharmacy p join keep k on k.pharmacyid=p.pharmacyid join medicine m on m.medicineid=k.medicineid where P.pharmacyname="Spot Rx" and k.discount>=30 group by m.medicineid having sum(k.quantity)<1000;



Export the data to SQL DATABASE:

sqoop export --connect jdbc:mysql://localhost:3306/healthcare --username root --table p10 --export-dir /user/hive/warehouse/out10/000000_0 --input-fields-terminated-by ','

```
23/03/15 10:35:36 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=566304
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=3001
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=19
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=0
        Job Counters
                Launched map tasks=4
                Data-local map tasks=4
                Total time spent by all maps in occupied slots (ms)=211893
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=211893
                Total vcore-seconds taken by all map tasks=211893
                Total megabyte-seconds taken by all map tasks=216978432
        Map-Reduce Framework
                Map input records=31
                Map output records=31
                Input split bytes=671
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=3639
                CPU time spent (ms)=6680
                Physical memory (bytes) snapshot=444518400
                Virtual memory (bytes) snapshot=6015369216
                Total committed heap usage (bytes)=243531776
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=0
23/03/15 10:35:36 INFO mapreduce.ExportJobBase: Transferred 2.9307 KB in 82.6733 seconds (36.2995 bytes/sec)
23/03/15 10:35:36 INFO mapreduce.ExportJobBase: Exported 31 records.
```