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:

hive>select count(*) as count,e.category 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;

hive>create view ps1 as select count(*) as count,e.category 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;

hive>select * from ps1;

External Table for Hive :

hive>create external table et1(category string,count int);

hive> insert OVERWRITE table et1 select * from ps1;

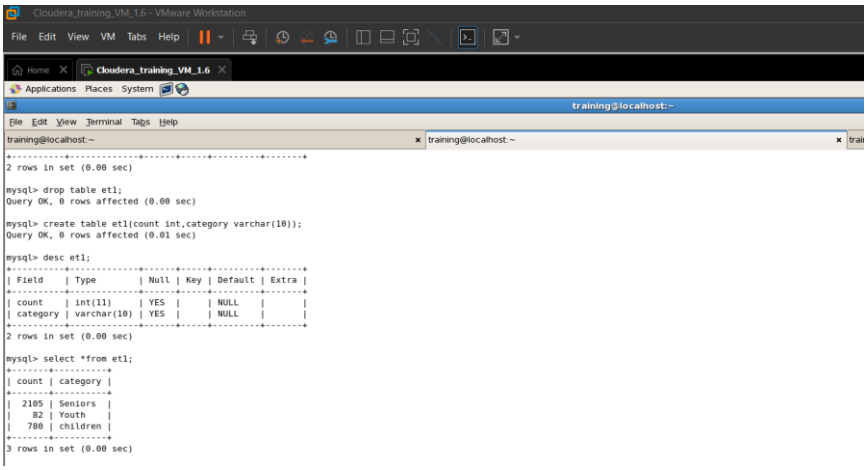
hive>select * from et1;

External Table for MYSQL :

mysql> create table et1(category varchar(10),count int);

SQOOP Export:

sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et1 --export-dir /user/hive/warehouse/et1/000000_0 --input-fields-terminated-by '\0001';



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:

```
create view male1 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;

create view female1 as select a.d diseasename, b.cnt male, a.cnt female

from (select diseasename d ,cnt from male1 where gender = 'female')a

join (select diseasename d, cnt from male1 where gender='male')b

on a.d=b.d;
```

External Table for Hive :

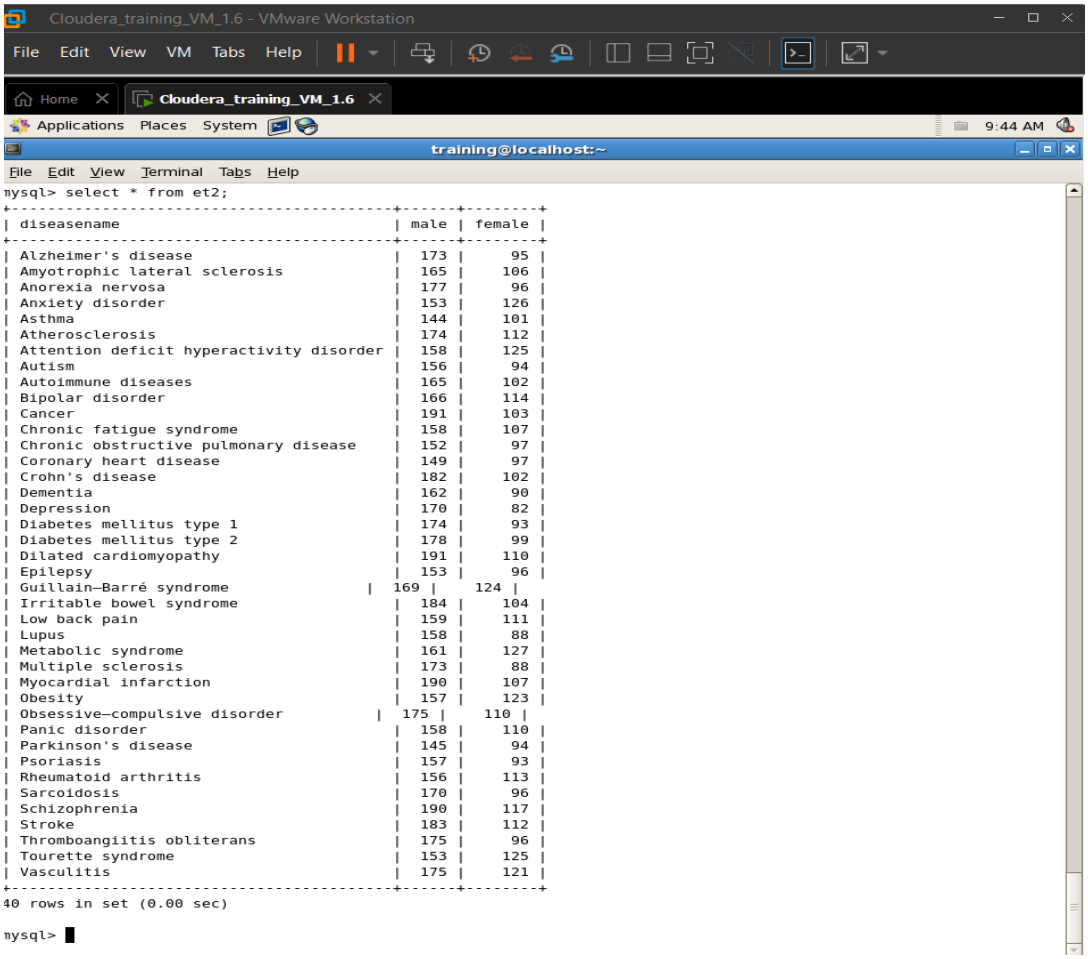
```
hive> create external table et2(diseasename string, male int,female int);

hive> insert overwrite table et2 select * from female1;
```

External Table Mysql :

```
mysql> create table et2(diseasename varchar(50),male int, female int);
```

SQOOP Export:

```
sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et2 --export-dir
/user/hive/warehouse/et2/000000_0 --input-fields-terminated-by '\0001';
```



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:

create view ps3 as 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;

External Table for Hive :

create external table et3(category string,count_tID int, count_cID int , ratio double);

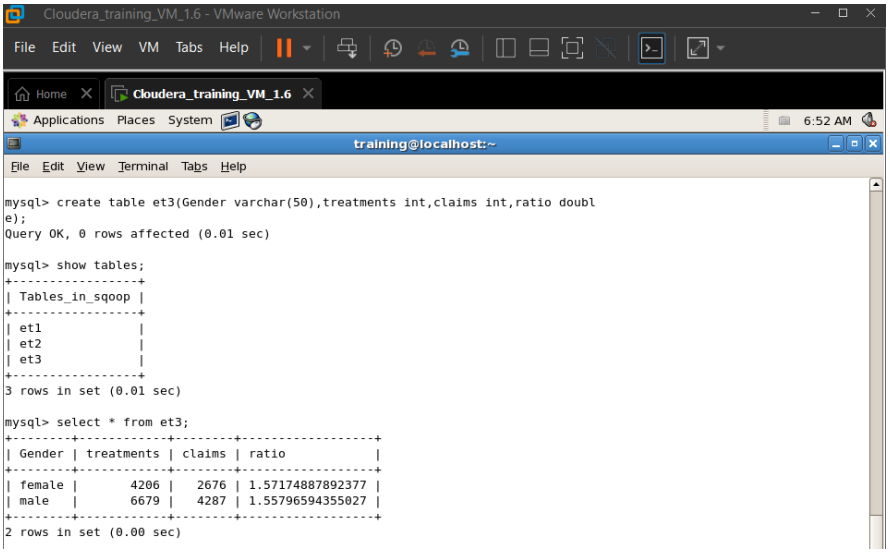
insert overwrite table et3 select * from ps3;

External Table for Mysql :

mysql>create table et3(Gender varchar(50),treatments int,claims int,ratio double);

SQOOP Export:

sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et3 --export-dir /user/hive/warehouse/et3/000000_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 :

create view ps4 as 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 for Hive :

create external table et4(pharmacyname string, Total double,Discount double);

insert overwrite table et4 select * from ps4;

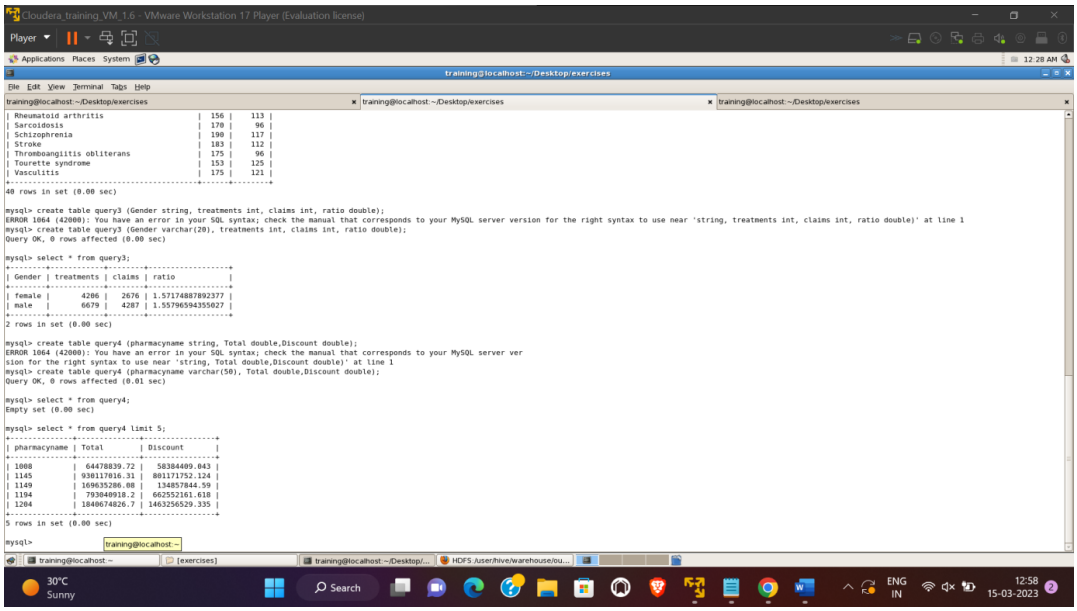
External Table for Mysql :

mysql> create table et4(pharmancyname varchar(50),Total double,Discount double);

SQOOP Export :

sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et4 --export-dir /user/hive/warehouse/et4/000000_0 --input-fields-terminated-by '\0001';

```
23/03/15 00:27:48 INFO mapred.JobClient: Running job: job_202303130454_0078
23/03/15 00:27:49 INFO mapred.JobClient:  map 0% reduce 0%
23/03/15 00:27:52 INFO mapred.JobClient:  map 100% reduce 0%
23/03/15 00:27:52 INFO mapred.JobClient: Job complete: job_202303130454_0078
23/03/15 00:27:52 INFO mapred.JobClient: Counters: 12
23/03/15 00:27:52 INFO mapred.JobClient:   Job Counters
23/03/15 00:27:52 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=2586
23/03/15 00:27:52 INFO mapred.JobClient:     Total time spent by all reduces
waiting after reserving slots (ms)=0
23/03/15 00:27:52 INFO mapred.JobClient:     Total time spent by all maps wai
ting after reserving slots (ms)=0
23/03/15 00:27:52 INFO mapred.JobClient:     Launched map tasks=1
23/03/15 00:27:52 INFO mapred.JobClient:     Data-local map tasks=1
23/03/15 00:27:52 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
23/03/15 00:27:52 INFO mapred.JobClient:   FileSystemCounters
23/03/15 00:27:52 INFO mapred.JobClient:     HDFS_BYTES_READ=7745
23/03/15 00:27:52 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=65656
23/03/15 00:27:52 INFO mapred.JobClient:   Map-Reduce Framework
23/03/15 00:27:52 INFO mapred.JobClient:     Map input records=213
23/03/15 00:27:52 INFO mapred.JobClient:     Spilled Records=0
23/03/15 00:27:52 INFO mapred.JobClient:     Map output records=213
23/03/15 00:27:52 INFO mapred.JobClient:     SPLIT_RAW_BYTES=126
23/03/15 00:27:52 INFO mapreduce.ExportJobBase: Transferred 7.5635 KB in 4.43
85 seconds (1.7041 KB/sec)
23/03/15 00:27:53 INFO mapreduce.ExportJobBase: Exported 213 records.
```



Problem Statement 5: The healthcare department suspects that some pharmacies prescribe more medicines than others in a single prescription, for them, generate a report that finds for each pharmacy the maximum, minimum and average number of medicines prescribed in their prescriptions.

Query :

create view ps5 as select pharmacyname, max(cnt) `max`, min(cnt) `min`,avg(cnt) `avg` from (select p1.pharmacyname, p.prescriptionid, sum(c.quantity) cnt from

prescription p join pharmacy p1

on p1.pharmacyid = p.pharmacyid

join contain c on c.prescriptionid = p.prescriptionid

group by p1.pharmacyname,p.prescriptionid)a

group by pharmacyname;

External Table for Hive :

create external table et5(pharmacyname string,`max` int, `min` int,avg double);

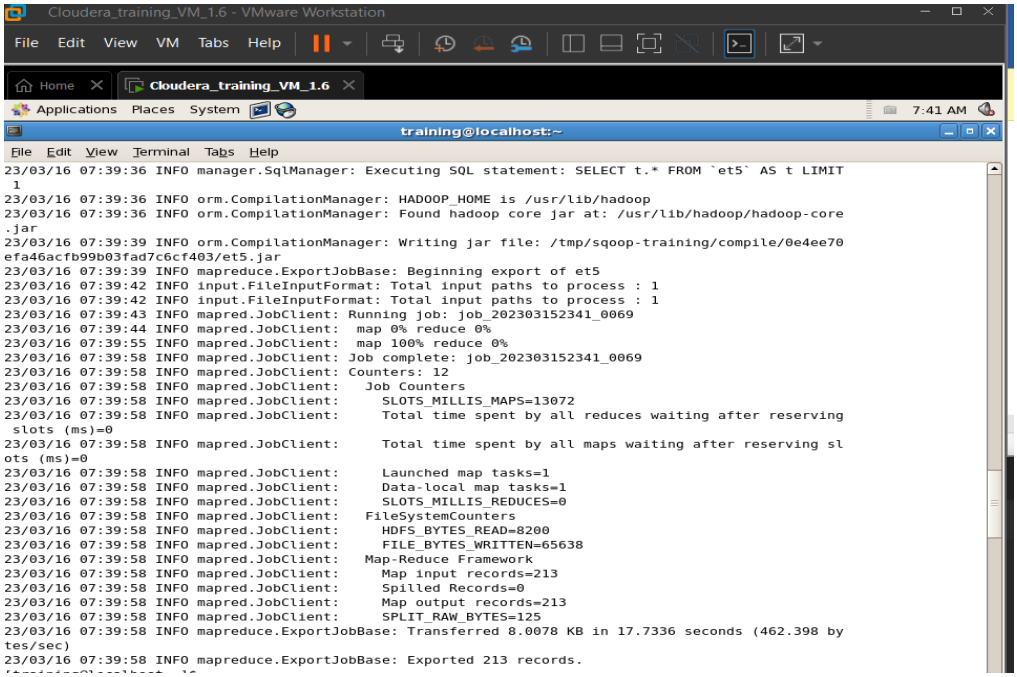
insert overwrite table et5 select * from ps5;

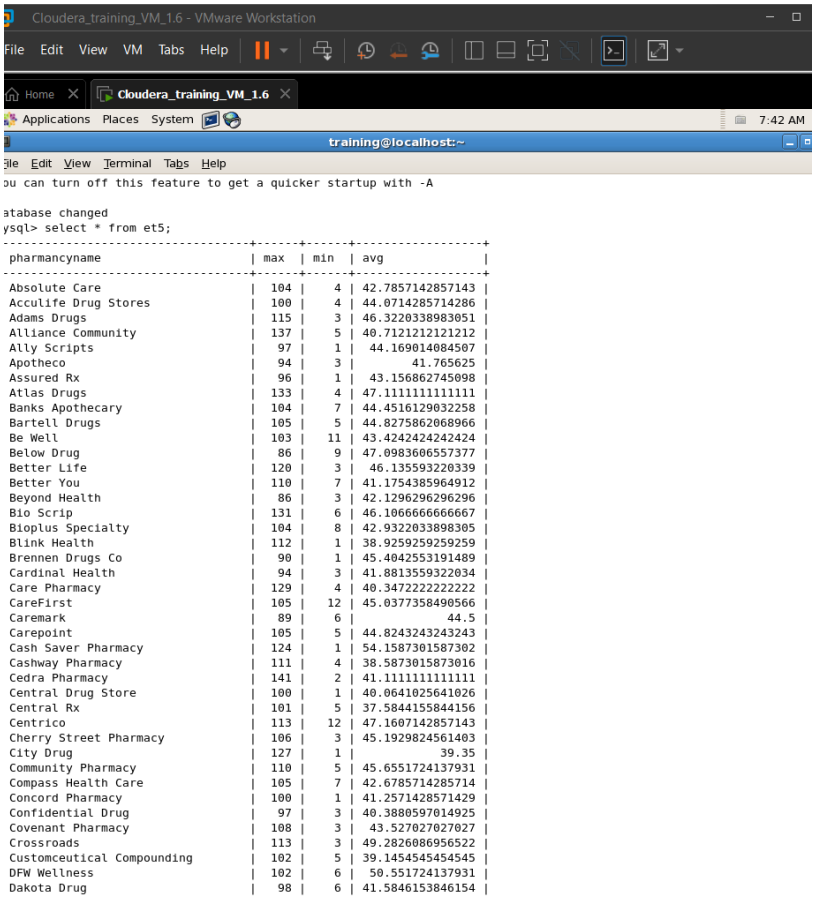
External Table for Mysql :

mysql> create table et5(pharmancyname varchar(50),max int,min int,avg double);

SQOOP Export :

sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et5 --export-dir /user/hive/warehouse/et5/000000_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 :

```
create view ps6 as 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 CNT)X

join Patient PA on PA.PATIENTID=X.PATIENTID

join Person P on P.PERSONID=PA.PATIENTID

order by TREATMENTCOUNT desc;
```

External Table for Hive :

create external table et6(personname string,Tcount int,age int);

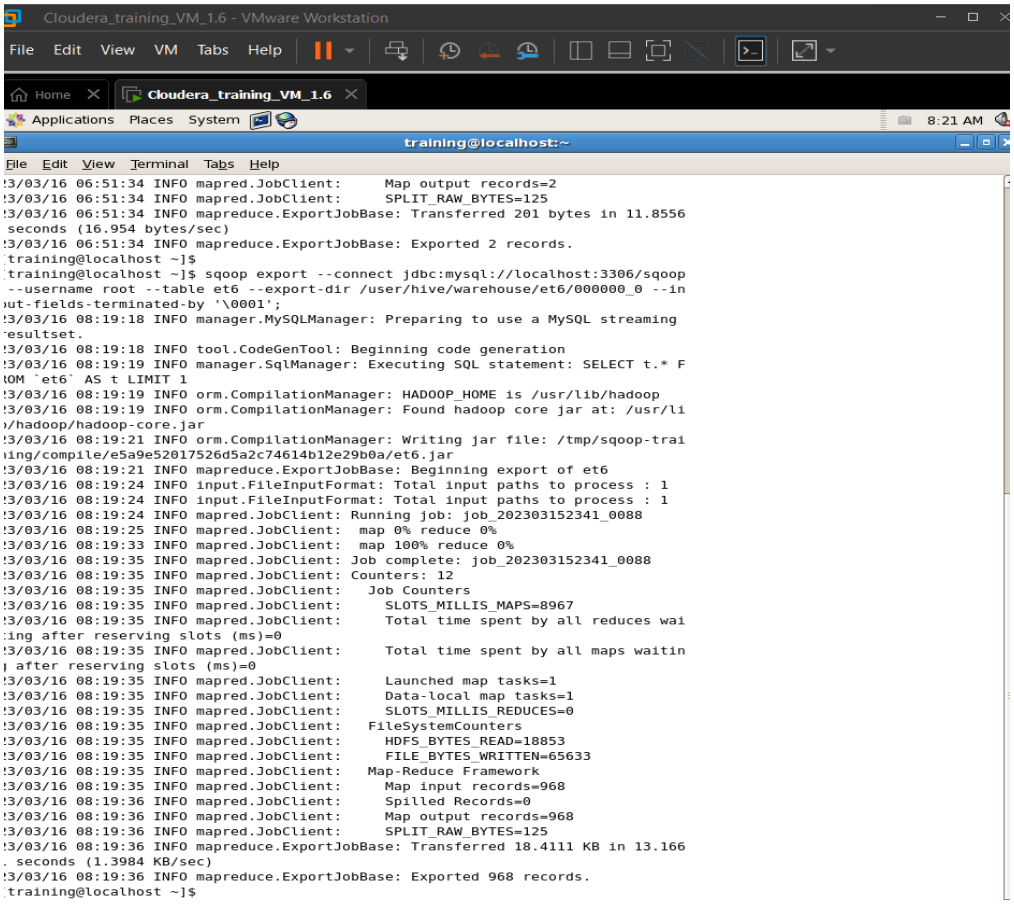
insert overwrite table et6 select * from ps6;

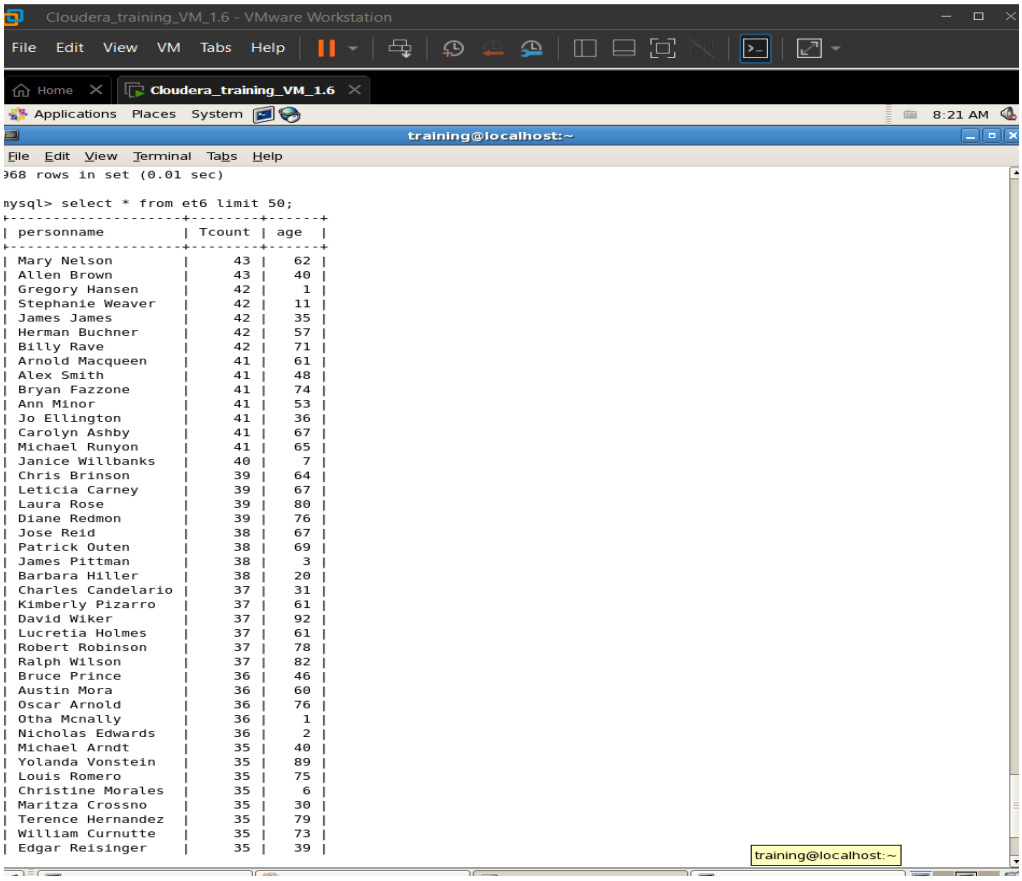
External Table for Mysql :

mysql>create table et6(personname varchar(50),Tcount int,age int);

SQOOP Export :

sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et6 --export-dir /user/hive/warehouse/et6/000000_0 --input-fields-terminated-by '\0001';





Problem statement 7: The State of Alabama (AL) is trying to manage its healthcare resources more efficiently. For each city in their state, they need to identify the disease for which the maximum number of patients have gone for treatment. Assist the state for this purpose.

Note: The state of Alabama is represented as AL in Address Table.

Query :

CREATE TABLE IF NOT EXISTS address_PART (addressid int, address1 String, city String,zip int)

COMMENT 'address_PART details'

PARTITIONED BY (state String)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

STORED AS TEXTFILE;

create view ps7 as select a.city,d.diseasename,count(t.patientid) as counTT from address_part a join person p on p.addressid=a.addressid join treatment t on t.patientID=p.personid join disease d on t.diseaseid=d.diseaseid where a.state='AL' group by a.city,d.diseasename;

External Table for Hive :

create external table et7(city string,diseasename string,counTT double);

insert overwrite table et7 select a.c,a.d,a.co from (select city as c,diseasename as d,counTT as count,ROW_NUMBER() partition by city order by counTT desc as rn from ps8) as a where a.rn=1;

External Table for Mysql :

create table et7(city varchar(50),diseasename varchar(50),count double);

SQOOP Export :

sqoop export --connect jdbc:mysql://localhost:3306/sqoop --username root --table et7 --export-dir /user/hive/warehouse/et7/000000_0 --input-fields-terminated-by '\0001';

```
23/03/14 08:30:55 INFO mapreduce.JobSubmitter: number of splits:4
23/03/14 08:30:55 INFO Configuration.deprecation: mapred.map.tasks.speculative.execution is deprecated. Instead, use mapreduce.map.speculative
23/03/14 08:30:55 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678798225978_0026
23/03/14 08:30:55 INFO impl.YarnClientImpl: Submitted application application_1678798225978_0026
23/03/14 08:30:55 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1678798225978_0026/
23/03/14 08:30:55 INFO mapreduce.Job: Running job: job_1678798225978_0026
23/03/14 08:31:04 INFO mapreduce.Job: Job job_1678798225978_0026 running in uber mode : false
23/03/14 08:31:04 INFO mapreduce.Job:  map 0% reduce 0%
23/03/14 08:31:25 INFO mapreduce.Job:  map 25% reduce 0%
23/03/14 08:31:29 INFO mapreduce.Job:  map 50% reduce 0%
23/03/14 08:31:30 INFO mapreduce.Job:  map 100% reduce 0%
23/03/14 08:31:31 INFO mapreduce.Job: Job job_1678798225978_0026 completed successfully
23/03/14 08:31:31 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=566056
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=886
                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)=84953
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=84953
                Total vcore-seconds taken by all map tasks=84953
                Total megabyte-seconds taken by all map tasks=86991872
        Map-Reduce Framework
                Map input records=3
                Map output records=3
                Input split bytes=584
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=1373
                CPU time spent (ms)=3350
                Physical memory (bytes) snapshot=499470336
                Virtual memory (bytes) snapshot=6015205376
                Total committed heap usage (bytes)=243531776
```

Problem statement 8: Some complaints have been lodged by patients that they have been prescribed hospital-exclusive medicine that they can't find elsewhere and facing problems due to that. Joshua, from the pharmacy management, wants to get a report of which pharmacies have prescribed hospital-exclusive medicines the most in the years 2021 and 2022. Assist Joshua to generate the report so that the pharmacies who prescribe hospital-exclusive medicine more often are advised to avoid such practice if possible.

Query :

Create view v8 as select ph.pharmacyid,count(c.medicineid) as counTT from treatment t join Prescription ph on t.treatmentid=ph.treatmentid join contain c on c.prescriptionid=ph.prescriptionid join medicine m on c.medicineid=m.medicineid where m.hospitalexclusive ='S' and year(t.date) in (2021,2022) group by ph.pharmacyid order by counTT desc;

External Table for Hive :

create table query8 (pharmacyid int, count int);

insert overwrite table out9 as select * from v8;

SQOOP Export :

sqoop export --connect jdbc:mysql://localhost:3306/client_DB --username root --table query8 --export-dir /user/hive/warehouse/out9/000000_0 --input-fields-terminated-by '\0001'

```
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 2
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_1678816063243_0002, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678816063243_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678816063243_0002
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2023-03-14 11:13:08,937 Stage-4 map = 0%,  reduce = 0%
2023-03-14 11:13:30,547 Stage-4 map = 100%,  reduce = 0%, Cumulative CPU 7.46 sec
2023-03-14 11:13:47,553 Stage-4 map = 100%,  reduce = 100%, Cumulative CPU 10.5 sec
MapReduce Total cumulative CPU time: 10 seconds 500 msec
Ended Job = job_1678816063243_0002
Launching Job 2 out of 2
```