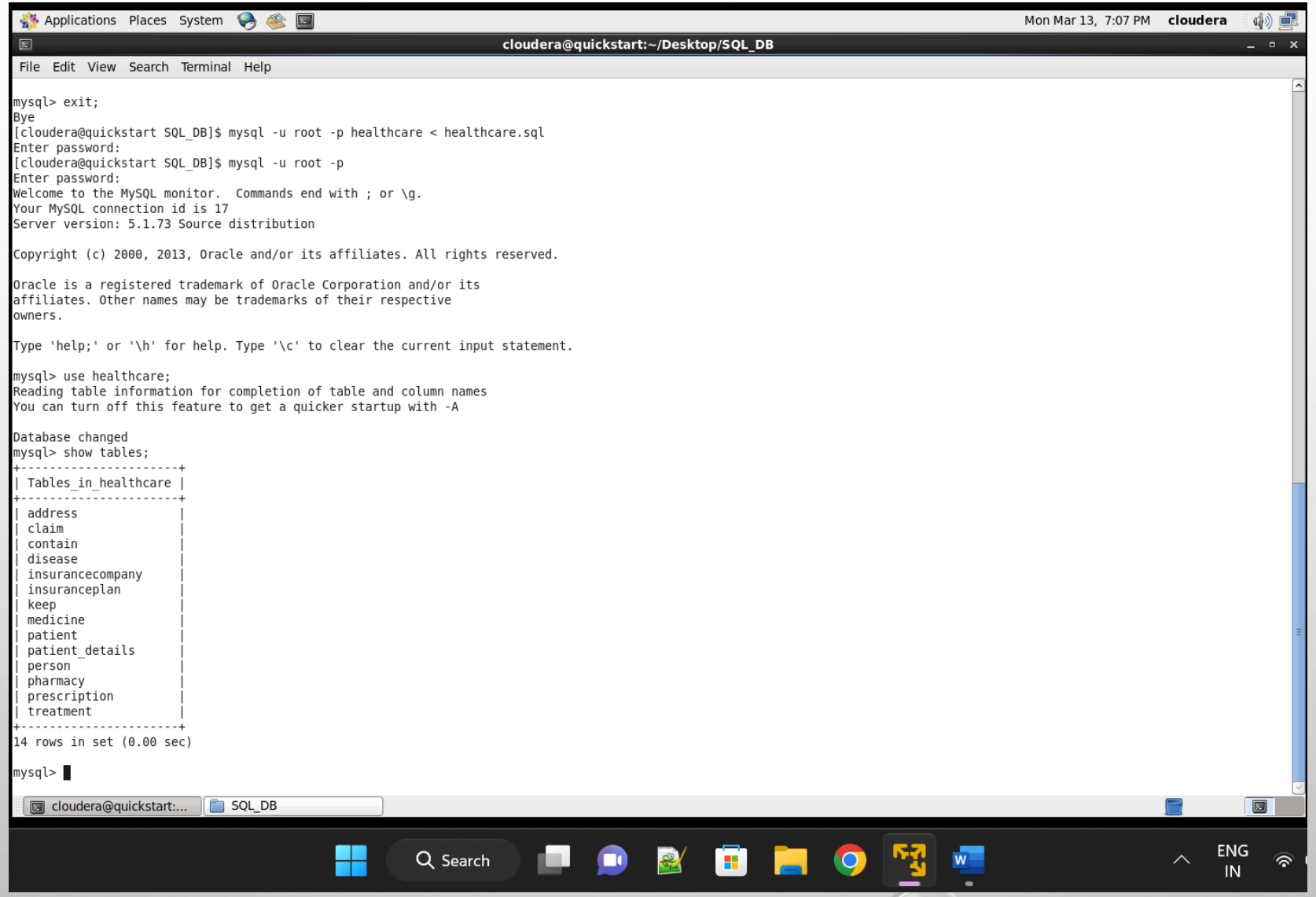
HIVE HEALTHCARE ANALYTICS

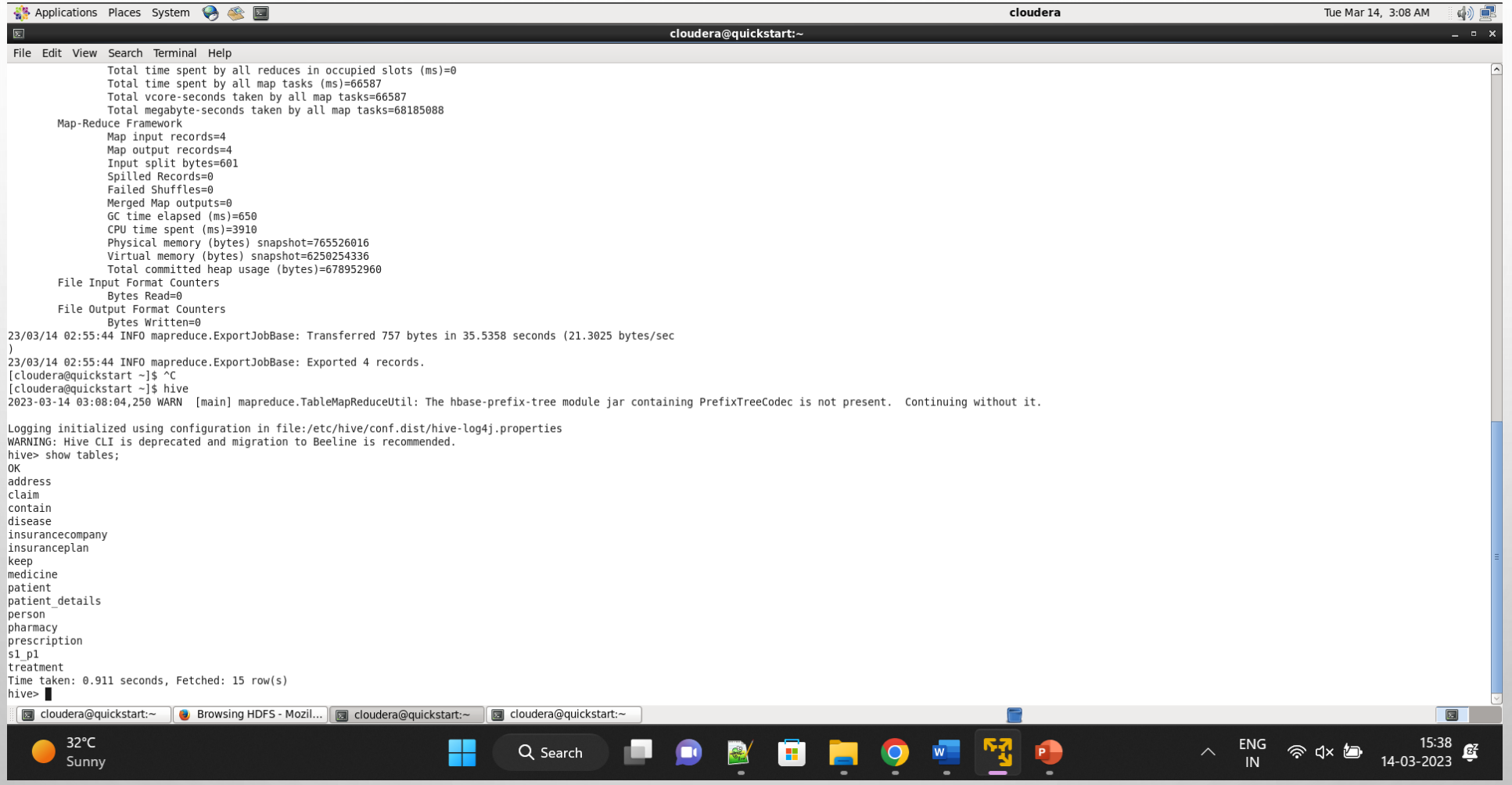
Import the SQL dump file in Cloudera MySQL Environment.

mysql -u root -p healthcare < healthcaredb_backUP.sql

Import all tables from MySQL to hive.

sqoop import-all-tables --connect jdbc:mysql://localhost:3306/healthcare --username root --hive-import -m 1


Applications Places System
Mon Mar 13, 7:07 PM cloudera
cloudera@quickstart:~/Desktop/SQL_DB
File Edit View Search Terminal Help
mysql> exit;
Bye
[cloudera@quickstart SQL_DB]$ mysql -u root -p healthcare < healthcare.sql
Enter password:
[cloudera@quickstart SQL_DB]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 5.1.73 Source distribution
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use healthcare;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
Database changed
mysql> show tables;
+----------------------+
| Tables_in_healthcare |
+----------------------+
| address |
| claim |
| contain |
| disease |
| insurancecompany |
| insuranceplan |
| keep |
| medicine |
| patient |
| patient_details |
| person |
| pharmacy |
| prescription |
| treatment |
+----------------------+
14 rows in set (0.00 sec)
mysql>
cloudera@quickstart:...
SQL_DB
Search
ENG
IN

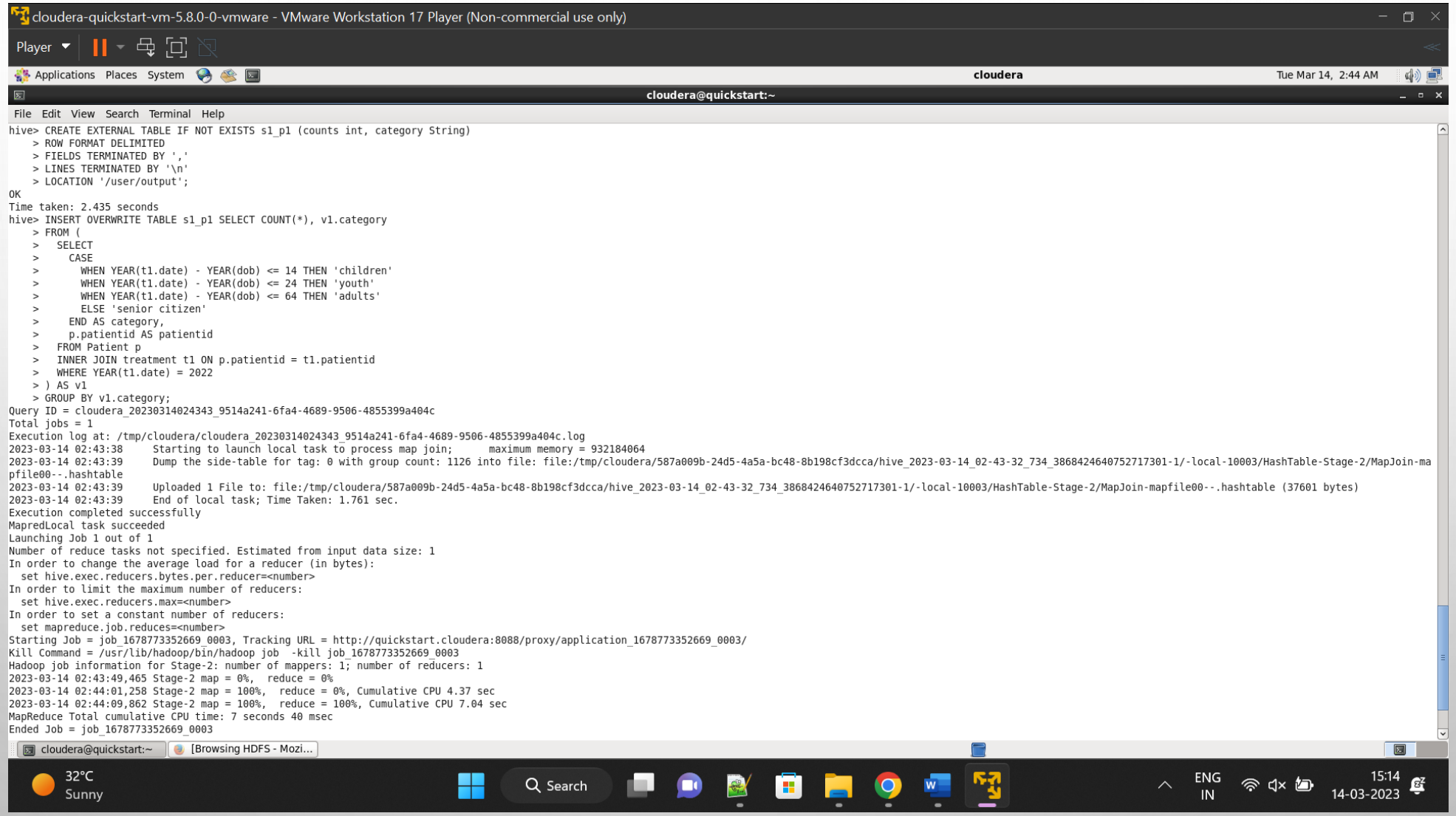

Applications Places System
cloudera
Tue Mar 14, 3:08 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
Total time spent by all reduces in occupied slots (ms)=0
Total time spent by all map tasks (ms)=66587
Total vcore-seconds taken by all map tasks=66587
Total megabyte-seconds taken by all map tasks=68185088
Map-Reduce Framework
Map input records=4
Map output records=4
Input split bytes=601
Spilled Records=0
Failed Shuffles=0
Merged Map outputs=0
GC time elapsed (ms)=650
CPU time spent (ms)=3910
Physical memory (bytes) snapshot=765526016
Virtual memory (bytes) snapshot=6250254336
Total committed heap usage (bytes)=678952960
File Input Format Counters
Bytes Read=0
File Output Format Counters
Bytes Written=0
23/03/14 02:55:44 INFO mapreduce.ExportJobBase: Transferred 757 bytes in 35.5358 seconds (21.3025 bytes/sec
)
23/03/14 02:55:44 INFO mapreduce.ExportJobBase: Exported 4 records.
[cloudera@quickstart ~]$ ^C
[cloudera@quickstart ~]$ hive
2023-03-14 03:08:04,250 WARN [main] mapreduce.TableMapReduceUtil: The hbase-prefix-tree module jar containing PrefixTreeCodec is not present. Continuing without it.
Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> show tables;
OK
address
claim
contain
disease
insurancecompany
insuranceplan
keep
medicine
patient
patient_details
person
pharmacy
prescription
s1_p1
treatment
Time taken: 0.911 seconds, Fetched: 15 row(s)
hive>
cloudera@quickstart:~
Browsing HDFS - Mozil...
cloudera@quickstart:~
cloudera@quickstart:~
32°C
Sunny
Search
ENG
IN
15:38
14-03-2023

Problem Statement S1_P1: 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.

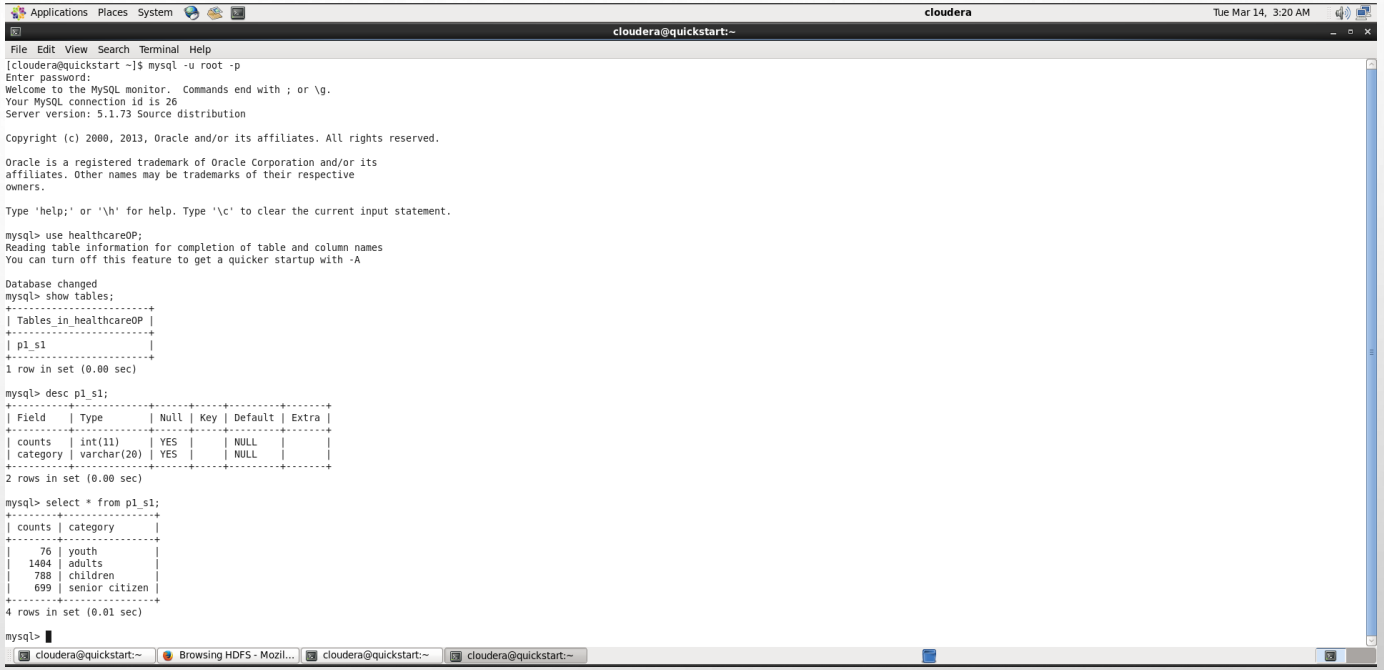
```
hive> CREATE EXTERNAL TABLE IF NOT EXISTS s1_p1 (counts int, category String)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/output';
OK
Time taken: 2.435 seconds
```

```
HIVE> INSERT OVERWRITE TABLE S1_P1 SELECT COUNT(*), V1.CATEGORY
    > FROM (
    >   SELECT
    >     CASE
    >       WHEN YEAR(T1.DATE) - YEAR(DOB) <= 14 THEN 'CHILDREN'
    >       WHEN YEAR(T1.DATE) - YEAR(DOB) <= 24 THEN 'YOUTH'
    >       WHEN YEAR(T1.DATE) - YEAR(DOB) <= 64 THEN 'ADULTS'
    >       ELSE 'SENIOR CITIZEN'
    >     END AS CATEGORY,
    >     P.PATIENTID AS PATIENTID
    >   FROM PATIENT P
    >   INNER JOIN TREATMENT T1 ON P.PATIENTID = T1.PATIENTID
    >   WHERE YEAR(T1.DATE) = 2022
    > ) AS V1
    > GROUP BY V1.CATEGORY;
```


cloudera-quickstart-vm-5.8.0-0-vmware - VMware Workstation 17 Player (Non-commercial use only)
Player
Applications Places System
cloudera
Tue Mar 14, 2:44 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
hive> CREATE EXTERNAL TABLE IF NOT EXISTS s1_p1 (counts int, category String)
> ROW FORMAT DELIMITED
> FIELDS TERMINATED BY ','
> LINES TERMINATED BY '\n'
> LOCATION '/user/output';
OK
Time taken: 2.435 seconds
hive> INSERT OVERWRITE TABLE s1_p1 SELECT COUNT(*), v1.category
> FROM (
> SELECT
> CASE
> WHEN YEAR(t1.date) - YEAR(dob) <= 14 THEN 'children'
> WHEN YEAR(t1.date) - YEAR(dob) <= 24 THEN 'youth'
> WHEN YEAR(t1.date) - YEAR(dob) <= 64 THEN 'adults'
> ELSE 'senior citizen'
> END AS category,
> p.patientid AS patientid
> FROM Patient p
> INNER JOIN treatment t1 ON p.patientid = t1.patientid
> WHERE YEAR(t1.date) = 2022
>) AS v1
> GROUP BY v1.category;
Query ID = cloudera_20230314024343_9514a241-6fa4-4689-9506-4855399a404c
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230314024343_9514a241-6fa4-4689-9506-4855399a404c.log
2023-03-14 02:43:38 Starting to launch local task to process map join; maximum memory = 932184064
2023-03-14 02:43:39 Dump the side-table for tag: 0 with group count: 1126 into file: file:/tmp/cloudera/587a009b-24d5-4a5a-bc48-8b198cf3dcca/hive_2023-03-14_02-43-32_734_3868424640752717301-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile00--.hashtable
2023-03-14 02:43:39 Uploaded 1 File to: file:/tmp/cloudera/587a009b-24d5-4a5a-bc48-8b198cf3dcca/hive_2023-03-14_02-43-32_734_3868424640752717301-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile00--.hashtable (37601 bytes)
2023-03-14 02:43:39 End of local task; Time Taken: 1.761 sec.
Execution completed successfully
MapredLocal task succeeded
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_1678773352669_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678773352669_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678773352669_0003
Hadoop job information for Stage-2: number of mappers: 1; number of reducers: 1
2023-03-14 02:43:49,465 Stage-2 map = 0%, reduce = 0%
2023-03-14 02:44:01,258 Stage-2 map = 100%, reduce = 0%, Cumulative CPU 4.37 sec
2023-03-14 02:44:09,862 Stage-2 map = 100%, reduce = 100%, Cumulative CPU 7.04 sec
MapReduce Total cumulative CPU time: 7 seconds 40 msec
Ended Job = job_1678773352669_0003
cloudera@quickstart:~
[Browsing HDFS - Mozi...
32°C
Sunny
Search
ENG
IN
15:14
14-03-2023

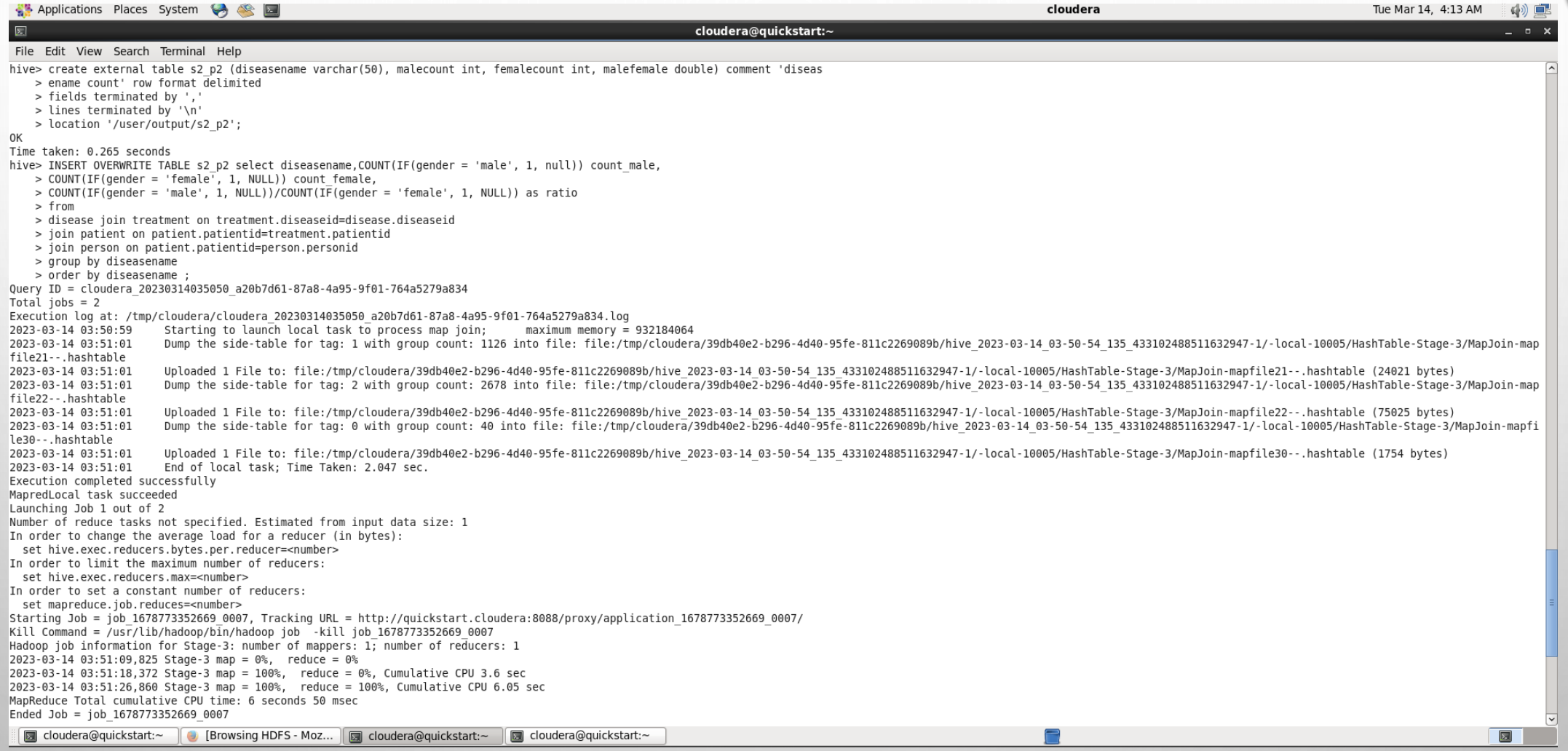
MYSQL> CREATE TABLE P1_S1(COUNTS INT,CATEGORY VARCHAR(20));

[CLOUDERA@QUICKSTART ~]$ SQOOP EXPORT --CONNECT JDBC:MYSQL://LOCALHOST:3306/HEALTHCAREOP --USERNAME ROOT --P --TABLE P1_S1 --EXPORT-DIR /USER/OUTPUT/000000_0 --INPUT-FIELDS-TERMINATED-BY ',';



Problem Statement S1_P2: 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.

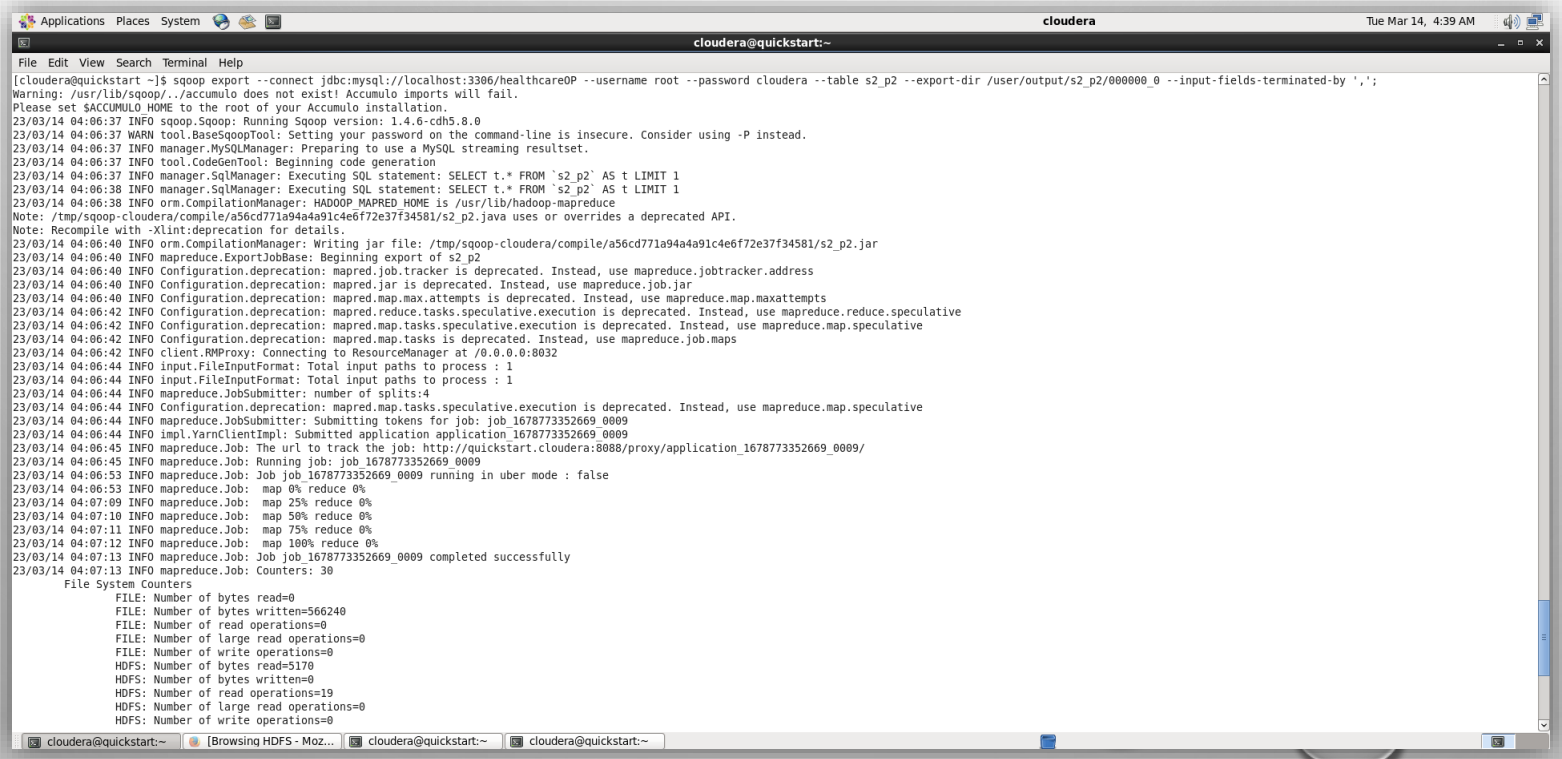
```
hive> create external table s2_p2 (diseasename varchar(50), malecount int, femalecount int, malefemale double) >
ename count' row format delimited
    > fields terminated by ','
    > lines terminated by '\n'
    > location '/user/output/s2_p2';

hive> INSERT OVERWRITE TABLE s2_p2 select diseasename,COUNT(IF(gender = 'male', 1, null)) count_male,
   > COUNT(IF(gender = 'female', 1, NULL)) count_female,
   > COUNT(IF(gender = 'male', 1, NULL))/COUNT(IF(gender = 'female', 1, NULL)) as ratio
   > from
   > disease join treatment on treatment.diseaseid=disease.diseaseid
   > join patient on patient.patientid=treatment.patientid
   > join person on patient.patientid=person.personid
   > group by diseasename
   > order by diseasename ;
```

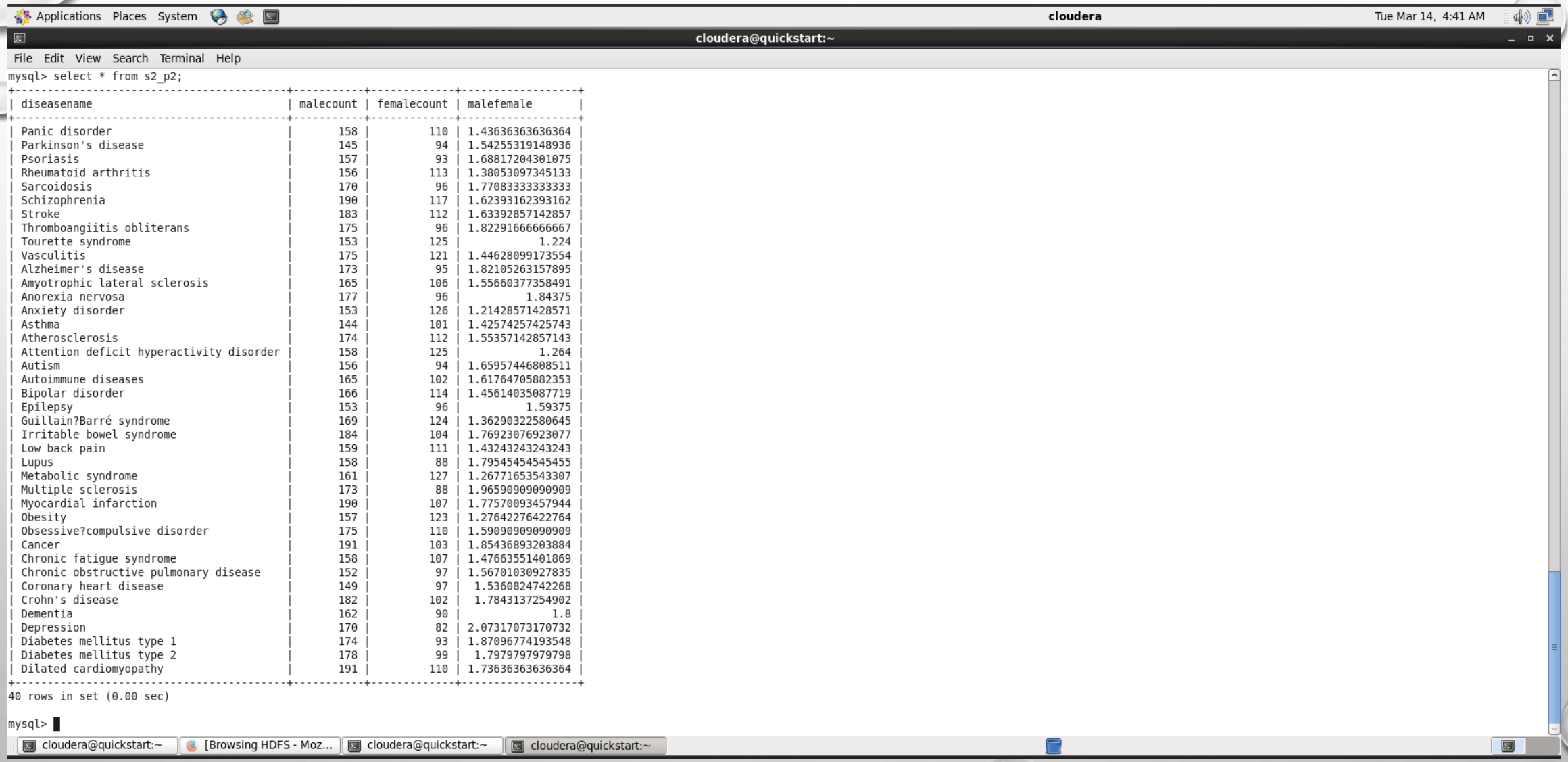

Applications Places System
cloudera
Tue Mar 14, 4:13 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
hive> create external table s2_p2 (diseasename varchar(50), malecount int, femalecount int, malefemale double) comment 'diseas
> ename count' row format delimited
> fields terminated by ','
> lines terminated by '\n'
> location '/user/output/s2_p2';
OK
Time taken: 0.265 seconds
hive> INSERT OVERWRITE TABLE s2_p2 select diseasename,COUNT(IF(gender = 'male', 1, null)) count_male,
> COUNT(IF(gender = 'female', 1, NULL)) count_female,
> COUNT(IF(gender = 'male', 1, NULL))/COUNT(IF(gender = 'female', 1, NULL)) as ratio
> from
> disease join treatment on treatment.diseaseid=disease.diseaseid
> join patient on patient.patientid=treatment.patientid
> join person on patient.patientid=person.personid
> group by diseasename
> order by diseasename ;
Query ID = cloudera_20230314035050_a20b7d61-87a8-4a95-9f01-764a5279a834
Total jobs = 2
Execution log at: /tmp/cloudera/cloudera_20230314035050_a20b7d61-87a8-4a95-9f01-764a5279a834.log
2023-03-14 03:50:59 Starting to launch local task to process map join; maximum memory = 932184064
2023-03-14 03:51:01 Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/39db40e2-b296-4d40-95fe-811c2269089b/hive_2023-03-14_03-50-54_135_433102488511632947-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile21--.hashtable
2023-03-14 03:51:01 Uploaded 1 File to: file:/tmp/cloudera/39db40e2-b296-4d40-95fe-811c2269089b/hive_2023-03-14_03-50-54_135_433102488511632947-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile21--.hashtable (24021 bytes)
2023-03-14 03:51:01 Dump the side-table for tag: 2 with group count: 2678 into file: file:/tmp/cloudera/39db40e2-b296-4d40-95fe-811c2269089b/hive_2023-03-14_03-50-54_135_433102488511632947-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile22--.hashtable
2023-03-14 03:51:01 Uploaded 1 File to: file:/tmp/cloudera/39db40e2-b296-4d40-95fe-811c2269089b/hive_2023-03-14_03-50-54_135_433102488511632947-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile22--.hashtable (75025 bytes)
2023-03-14 03:51:01 Dump the side-table for tag: 0 with group count: 40 into file: file:/tmp/cloudera/39db40e2-b296-4d40-95fe-811c2269089b/hive_2023-03-14_03-50-54_135_433102488511632947-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile30--.hashtable
2023-03-14 03:51:01 Uploaded 1 File to: file:/tmp/cloudera/39db40e2-b296-4d40-95fe-811c2269089b/hive_2023-03-14_03-50-54_135_433102488511632947-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile30--.hashtable (1754 bytes)
2023-03-14 03:51:01 End of local task; Time Taken: 2.047 sec.
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_1678773352669_0007, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678773352669_0007/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678773352669_0007
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2023-03-14 03:51:09,825 Stage-3 map = 0%, reduce = 0%
2023-03-14 03:51:18,372 Stage-3 map = 100%, reduce = 0%, Cumulative CPU 3.6 sec
2023-03-14 03:51:26,860 Stage-3 map = 100%, reduce = 100%, Cumulative CPU 6.05 sec
MapReduce Total cumulative CPU time: 6 seconds 50 msec
Ended Job = job_1678773352669_0007
cloudera@quickstart:~
[Browsing HDFS - Moz...
cloudera@quickstart:~
cloudera@quickstart:~

mysql> create table s2_p2(diseasename varchar(50), malecount int, femalecount int, malefemale double);
Query OK, 0 rows affected (0.01 sec)

```
mysql> create table s2_p2(diseasename varchar(50), malecount int, femalecount in
t, malefemale double);
Query OK, 0 rows affected (0.01 sec)
```

[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcareOP --username root --password cloudera --table s2_p2 --export-dir /user/output/s2_p2/000000_0 --input-fields-terminated-by ',';




Applications Places System
cloudera
Tue Mar 14, 4:41 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
mysql> select * from s2_p2;
diseasename	malecount	femalecount	malefemale
Panic disorder	158	110	1.43636363636364
Parkinson's disease	145	94	1.54255319148936
Psoriasis	157	93	1.68817204301075
Rheumatoid arthritis	156	113	1.38053097345133
Sarcoidosis	170	96	1.77083333333333
Schizophrenia	190	117	1.62393162393162
Stroke	183	112	1.63392857142857
Thromboangiitis obliterans	175	96	1.82291666666667
Tourette syndrome	153	125	1.224
Vasculitis	175	121	1.44628099173554
Alzheimer's disease	173	95	1.82105263157895
Amyotrophic lateral sclerosis	165	106	1.55660377358491
Anorexia nervosa	177	96	1.84375
Anxiety disorder	153	126	1.21428571428571
Asthma	144	101	1.42574257425743
Atherosclerosis	174	112	1.55357142857143
Attention deficit hyperactivity disorder	158	125	1.264
Autism	156	94	1.65957446808511
Autoimmune diseases	165	102	1.61764705882353
Bipolar disorder	166	114	1.45614035087719
Epilepsy	153	96	1.59375
Guillain?Barré syndrome	169	124	1.36290322580645
Irritable bowel syndrome	184	104	1.76923076923077
Low back pain	159	111	1.43243243243243
Lupus	158	88	1.79545454545455
Metabolic syndrome	161	127	1.26771653543307
Multiple sclerosis	173	88	1.96590909090909
Myocardial infarction	190	107	1.77570093457944
Obesity	157	123	1.27642276422764
Obsessive?compulsive disorder	175	110	1.59090909090909
Cancer	191	103	1.85436893203884
Chronic fatigue syndrome	158	107	1.47663551401869
Chronic obstructive pulmonary disease	152	97	1.56701030927835
Coronary heart disease	149	97	1.5360824742268
Crohn's disease	182	102	1.7843137254902
Dementia	162	90	1.8
Depression	170	82	2.07317073170732
Diabetes mellitus type 1	174	93	1.87096774193548
Diabetes mellitus type 2	178	99	1.7979797979798
Dilated cardiomyopathy	191	110	1.73636363636364
40 rows in set (0.00 sec)
mysql>
cloudera@quickstart:~
[Browsing HDFS - Moz...
cloudera@quickstart:~
cloudera@quickstart:~

Problem Statement S2_P5: An Insurance company wants a state wise report of the treatments to claim ratio

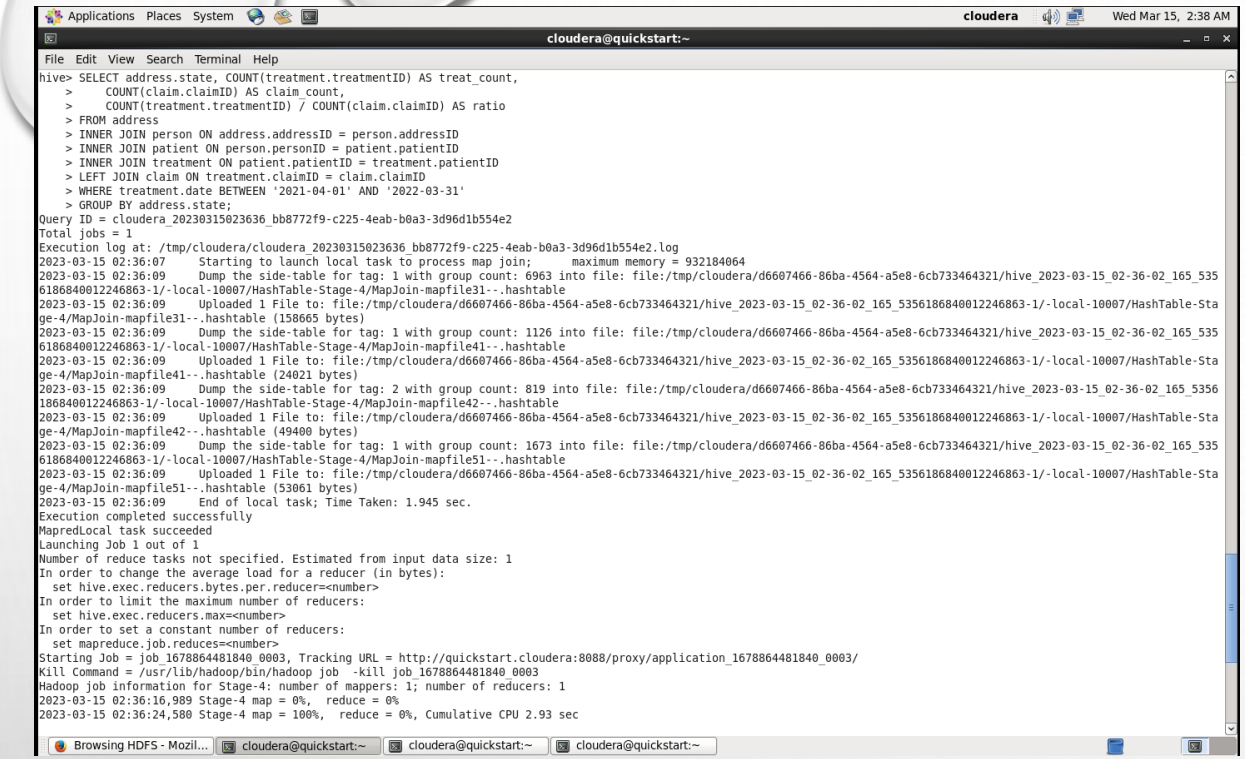
between 1st April 2021 and 31st March 2022 (days both included).

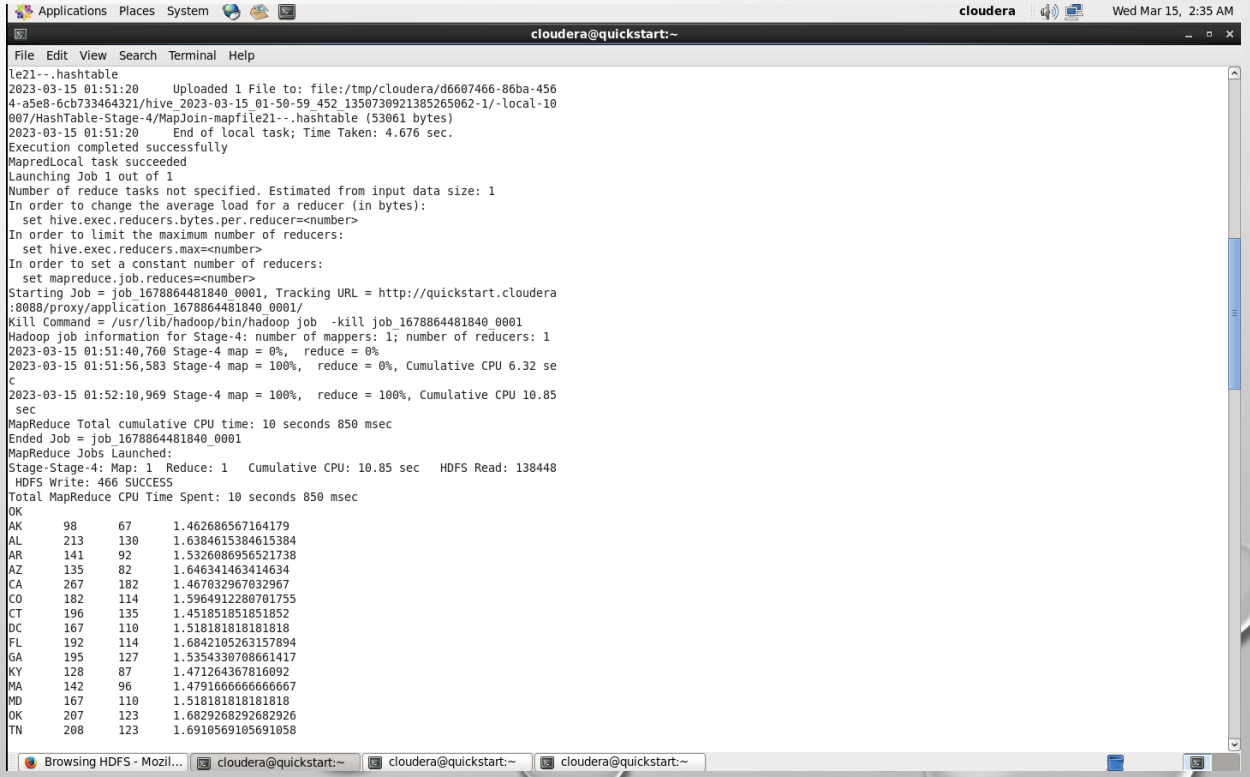
Assist them to create such a report.

SELECT address.state, COUNT(treatment.treatmentID) AS treat_count, COUNT(claim.claimID) AS claim_count, COUNT(treatment.treatmentID) / COUNT(claim.claimID) AS ratioFROM addressINNER JOIN person ON address.addressID = person.addressIDINNER JOIN patient ON person.personID = patient.patientIDINNER JOIN treatment ON patient.patientID = treatment.patientIDLEFT JOIN claim ON treatment.claimID = claim.claimIDWHERE treatment.date BETWEEN '2021-04-01' AND '2022-03-31'GROUP BY address.state;

create table address_part1 (addressid int , address1 string, city string, zip int) partitioned by (state string);

insert into address_part1 partition(state) select addressid, address1,city, zip,state from address;

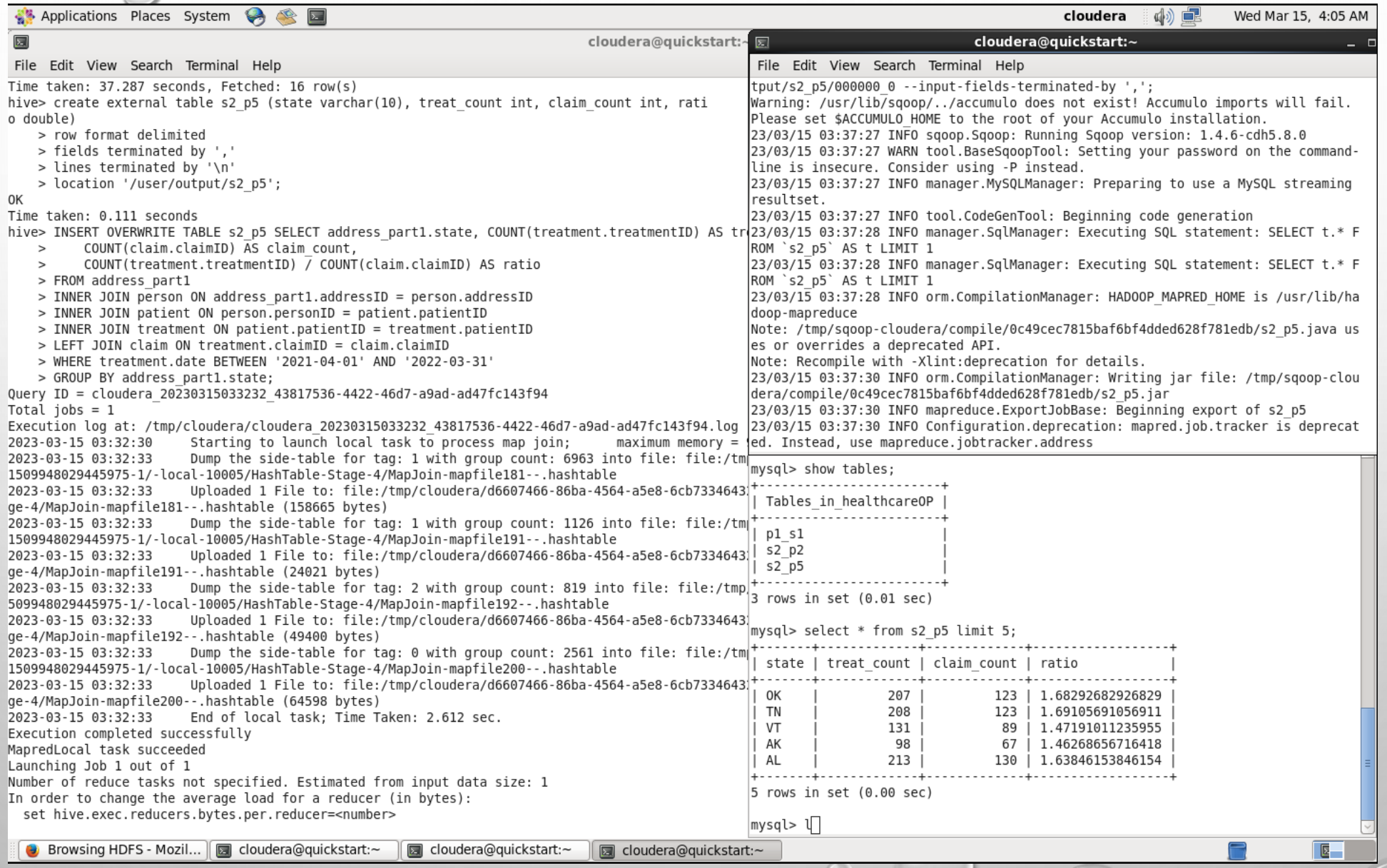

Applications Places System
cloudera
Wed Mar 15, 2:38 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
hive> SELECT address.state, COUNT(treatment.treatmentID) AS treat_count,
> COUNT(claim.claimID) AS claim_count,
> COUNT(treatment.treatmentID) / COUNT(claim.claimID) AS ratio
> FROM address
> INNER JOIN person ON address.addressID = person.addressID
> INNER JOIN patient ON person.personID = patient.patientID
> INNER JOIN treatment ON patient.patientID = treatment.patientID
> LEFT JOIN claim ON treatment.claimID = claim.claimID
> WHERE treatment.date BETWEEN '2021-04-01' AND '2022-03-31'
> GROUP BY address.state;
Query ID = cloudera_20230315023636_bb8772f9-c225-4eab-b0a3-3d96d1b554e2
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230315023636_bb8772f9-c225-4eab-b0a3-3d96d1b554e2.log
2023-03-15 02:36:07 Starting to launch local task to process map join; maximum memory = 932184064
2023-03-15 02:36:09 Dump the side-table for tag: 1 with group count: 6963 into file: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile31--.hashtable
2023-03-15 02:36:09 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile31--.hashtable (158665 bytes)
2023-03-15 02:36:09 Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile41--.hashtable
2023-03-15 02:36:09 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile41--.hashtable (24021 bytes)
2023-03-15 02:36:09 Dump the side-table for tag: 2 with group count: 819 into file: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile42--.hashtable
2023-03-15 02:36:09 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile42--.hashtable (49400 bytes)
2023-03-15 02:36:09 Dump the side-table for tag: 1 with group count: 1673 into file: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile51--.hashtable
2023-03-15 02:36:09 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb733464321/hive_2023-03-15_02-36-02_165_5356186840012246863-1/-local-10007/HashTable-Stage-4/MapJoin-mapfile51--.hashtable (53061 bytes)
2023-03-15 02:36:09 End of local task; Time Taken: 1.945 sec.
Execution completed successfully
MapredLocal task succeeded
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_1678864481840_0003, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678864481840_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678864481840_0003
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2023-03-15 02:36:16,989 Stage-4 map = 0%, reduce = 0%
2023-03-15 02:36:24,580 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 2.93 sec
Browsing HDFS - Mozil...
cloudera@quickstart:~
cloudera@quickstart:~
cloudera@quickstart:~


Applications Places System
cloudera
Wed Mar 15, 2:35 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
le21--.hashtable
2023-03-15 01:51:20 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-456
4-a5e8-6cb733464321/hive_2023-03-15_01-50-59_452_1350730921385265062-1/-local-10
007/HashTable-Stage-4/MapJoin-mapfile21--.hashtable (53061 bytes)
2023-03-15 01:51:20 End of local task; Time Taken: 4.676 sec.
Execution completed successfully
MapredLocal task succeeded
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_1678864481840_0001, Tracking URL = http://quickstart.cloudera
:8088/proxy/application_1678864481840_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678864481840_0001
Hadoop job information for Stage-4: number of mappers: 1; number of reducers: 1
2023-03-15 01:51:40,760 Stage-4 map = 0%, reduce = 0%
2023-03-15 01:51:56,583 Stage-4 map = 100%, reduce = 0%, Cumulative CPU 6.32 se
c
2023-03-15 01:52:10,969 Stage-4 map = 100%, reduce = 100%, Cumulative CPU 10.85
sec
MapReduce Total cumulative CPU time: 10 seconds 850 msec
Ended Job = job_1678864481840_0001
MapReduce Jobs Launched:
Stage-Stage-4: Map: 1 Reduce: 1 Cumulative CPU: 10.85 sec HDFS Read: 138448
HDFS Write: 466 SUCCESS
Total MapReduce CPU Time Spent: 10 seconds 850 msec
OK
AK 98 67 1.462686567164179
AL 213 130 1.6384615384615384
AR 141 92 1.5326086956521738
AZ 135 82 1.6463414634146634
CA 267 182 1.467032967032967
CO 182 114 1.5964912280701755
CT 196 135 1.451851851851852
DC 167 110 1.5181818181818181
FL 192 114 1.6842105263157894
GA 195 127 1.5354330708661417
KY 128 87 1.471264367816092
MA 142 96 1.4791666666666667
MD 167 110 1.5181818181818181
OK 207 123 1.6829268292682926
TN 208 123 1.6910569105691058
Browsing HDFS - Mozil...
cloudera@quickstart:~
cloudera@quickstart:~
cloudera@quickstart:~

```
create external table s2_p5 (state varchar(10), treat_count int, claim_count int, ratio double)
    > row format delimited
    > fields terminated by ','
    > lines terminated by '\n'
    > location '/user/output/s2_p5';
OK
Time taken: 0.111 seconds
hive> INSERT OVERWRITE TABLE s2_p5 SELECT address_part1.state, COUNT(treatment.treatmentID) AS treat_count,
    >     COUNT(claim.claimID) AS claim_count,
    >     COUNT(treatment.treatmentID) / COUNT(claim.claimID) AS ratio
    > FROM address_part1
    > INNER JOIN person ON address_part1.addressID = person.addressID
    > INNER JOIN patient ON person.personID = patient.patientID
    > INNER JOIN treatment ON patient.patientID = treatment.patientID
    > LEFT JOIN claim ON treatment.claimID = claim.claimID
    > WHERE treatment.date BETWEEN '2021-04-01' AND '2022-03-31'
    > GROUP BY address_part1.state;
```

```
mysql> create table s2_p5 (state varchar(10), treat_count int, claim_count int, ratio double);
```

```
[cloudera@quickstart ~]$ sqoop export --connect jdbc:mysql://localhost:3306/healthcareOP --username root --
password cloudera --table s2_p5 --export-dir /user/output/s2_p5/000000_0 --input-fields-terminated-by ',';
```


Applications Places System
cloudera
Wed Mar 15, 4:05 AM
cloudera@quickstart:~
File Edit View Search Terminal Help
Time taken: 37.287 seconds, Fetched: 16 row(s)
hive> create external table s2_p5 (state varchar(10), treat_count int, claim_count int, rati
o double)
> row format delimited
> fields terminated by ','
> lines terminated by '\n'
> location '/user/output/s2_p5';
OK
Time taken: 0.111 seconds
hive> INSERT OVERWRITE TABLE s2_p5 SELECT address_part1.state, COUNT(treatment.treatmentID) AS tr
> COUNT(claim.claimID) AS claim_count,
> COUNT(treatment.treatmentID) / COUNT(claim.claimID) AS ratio
> FROM address_part1
> INNER JOIN person ON address_part1.addressID = person.addressID
> INNER JOIN patient ON person.personID = patient.patientID
> INNER JOIN treatment ON patient.patientID = treatment.patientID
> LEFT JOIN claim ON treatment.claimID = claim.claimID
> WHERE treatment.date BETWEEN '2021-04-01' AND '2022-03-31'
> GROUP BY address_part1.state;
Query ID = cloudera_20230315033232_43817536-4422-46d7-a9ad-ad47fc143f94
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230315033232_43817536-4422-46d7-a9ad-ad47fc143f94.log
2023-03-15 03:32:30 Starting to launch local task to process map join; maximum memory =
2023-03-15 03:32:33 Dump the side-table for tag: 1 with group count: 6963 into file: file:/tm
1509948029445975-1/-local-10005/HashTable-Stage-4/MapJoin-mapfile181--.hashtable
2023-03-15 03:32:33 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb7334643
ge-4/MapJoin-mapfile181--.hashtable (158665 bytes)
2023-03-15 03:32:33 Dump the side-table for tag: 1 with group count: 1126 into file: file:/tm
1509948029445975-1/-local-10005/HashTable-Stage-4/MapJoin-mapfile191--.hashtable
2023-03-15 03:32:33 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb7334643
ge-4/MapJoin-mapfile191--.hashtable (24021 bytes)
2023-03-15 03:32:33 Dump the side-table for tag: 2 with group count: 819 into file: file:/tmp
509948029445975-1/-local-10005/HashTable-Stage-4/MapJoin-mapfile192--.hashtable
2023-03-15 03:32:33 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb7334643
ge-4/MapJoin-mapfile192--.hashtable (49400 bytes)
2023-03-15 03:32:33 Dump the side-table for tag: 0 with group count: 2561 into file: file:/tm
1509948029445975-1/-local-10005/HashTable-Stage-4/MapJoin-mapfile200--.hashtable
2023-03-15 03:32:33 Uploaded 1 File to: file:/tmp/cloudera/d6607466-86ba-4564-a5e8-6cb7334643
ge-4/MapJoin-mapfile200--.hashtable (64598 bytes)
2023-03-15 03:32:33 End of local task; Time Taken: 2.612 sec.
Execution completed successfully
MapredLocal task succeeded
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>
cloudera@quickstart:~
File Edit View Search Terminal Help
tput/s2_p5/000000_0 --input-fields-terminated-by ',';
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/03/15 03:37:27 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
23/03/15 03:37:27 WARN tool.BaseSqoopTool: Setting your password on the command-
line is insecure. Consider using -P instead.
23/03/15 03:37:27 INFO manager.MySQLManager: Preparing to use a MySQL streaming
resultset.
23/03/15 03:37:27 INFO tool.CodeGenTool: Beginning code generation
23/03/15 03:37:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* F
ROM `s2_p5` AS t LIMIT 1
23/03/15 03:37:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* F
ROM `s2_p5` AS t LIMIT 1
23/03/15 03:37:28 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/ha
doop-mapreduce
Note: /tmp/sqoop-cloudera/compile/0c49cec7815baf6bf4dded628f781edb/s2_p5.java us
es or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
23/03/15 03:37:30 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-clou
dera/compile/0c49cec7815baf6bf4dded628f781edb/s2_p5.jar
23/03/15 03:37:30 INFO mapreduce.ExportJobBase: Beginning export of s2_p5
23/03/15 03:37:30 INFO Configuration.deprecation: mapred.job.tracker is deprecat
ed. Instead, use mapreduce.jobtracker.address
mysql> show tables;
+-------------------------+
| Tables_in_healthcareOP |
+-------------------------+
| p1_s1 |
| s2_p2 |
| s2_p5 |
+-------------------------+
3 rows in set (0.01 sec)
mysql> select * from s2_p5 limit 5;
+-------+-------------+-------------+------------------+
| state | treat_count | claim_count | ratio |
+-------+-------------+-------------+------------------+
OK	207	123	1.68292682926829
TN	208	123	1.69105691056911
VT	131	89	1.47191011235955
AK	98	67	1.46268656716418
AL	213	130	1.63846153846154
+-------+-------------+-------------+------------------+
5 rows in set (0.00 sec)
mysql>
Browsing HDFS - Mozil...
cloudera@quickstart:~
cloudera@quickstart:~
cloudera@quickstart:~

/*

Problem Statement S1_P3: 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.

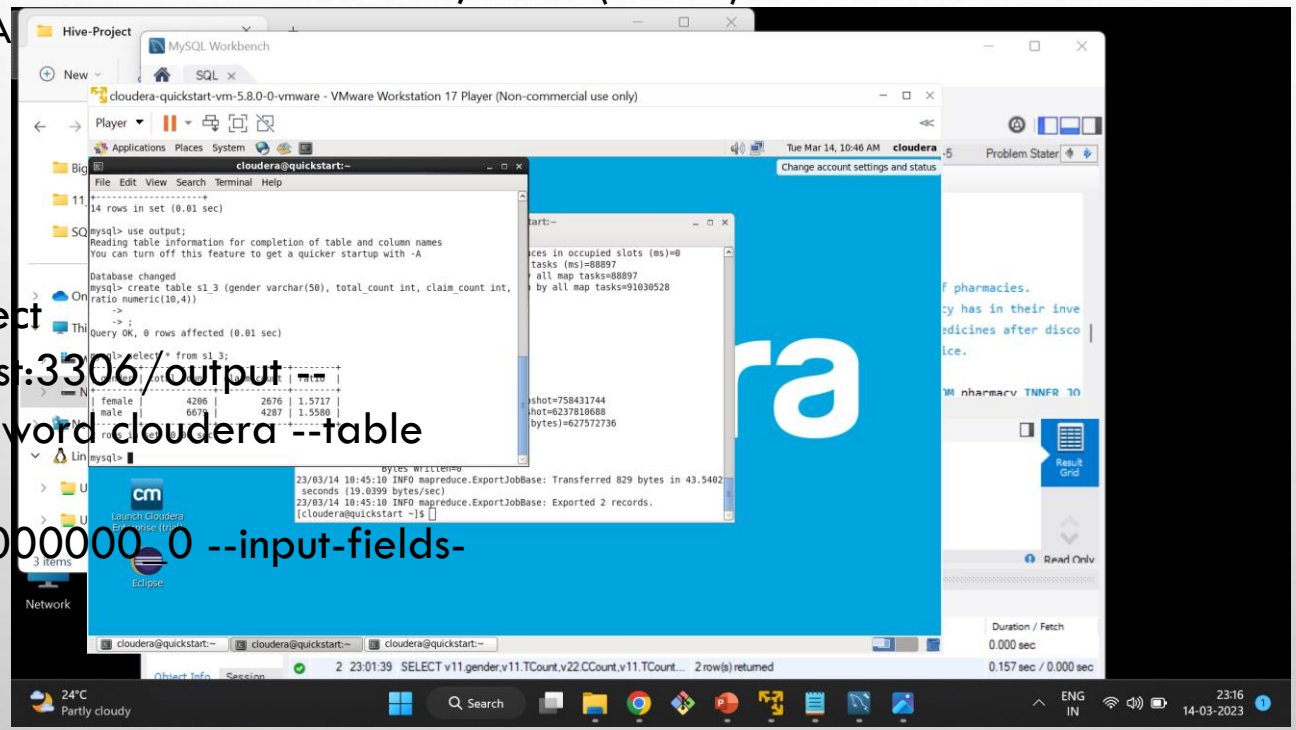
And notice if there is a significant difference between the treatment-to-claim ratio of male and female patients.

*/

```
create external table s1_3 (gender varchar(50), total_count int, claim_count int, ratio double)
row format delimited
fields terminated by ','
lines terminated by '\n'
location '/user/output/s1_3';
```

```
INSERT OVERWRITE TABLE S1_3
SELECT V11.GENDER,V11.TCOUNT,V22.CCOUNT,V11.TCOUNT/V22.CCOUNT AS RATIO FROM
(SELECT P.GENDER AS GENDER,COUNT(V1.DID) AS TCOUNT FROM PERSON P JOIN
(SELECT T.PATIENTID AS PID,T.DISEASEID AS DID  FROM TREATMENT T ) AS V1
ON P.PERSONID=V1.PID
GROUP BY P.GENDER)AS V11
JOIN
(SELECT P.GENDER AS GENDER,COUNT(V2.CID) AS CCOUNT FROM PERSON P JOIN
(SELECT T.PATIENTID A                                                    = C.CLAIMID ) AS V2
```

```
sqoop export --connect
jdbc:mysql://localhost:3306/output --
username root --password cloudera --table
s1_3 --export-dir
/user/output/s1_3/000000_0 --input-fields-
terminated-by ',';
```

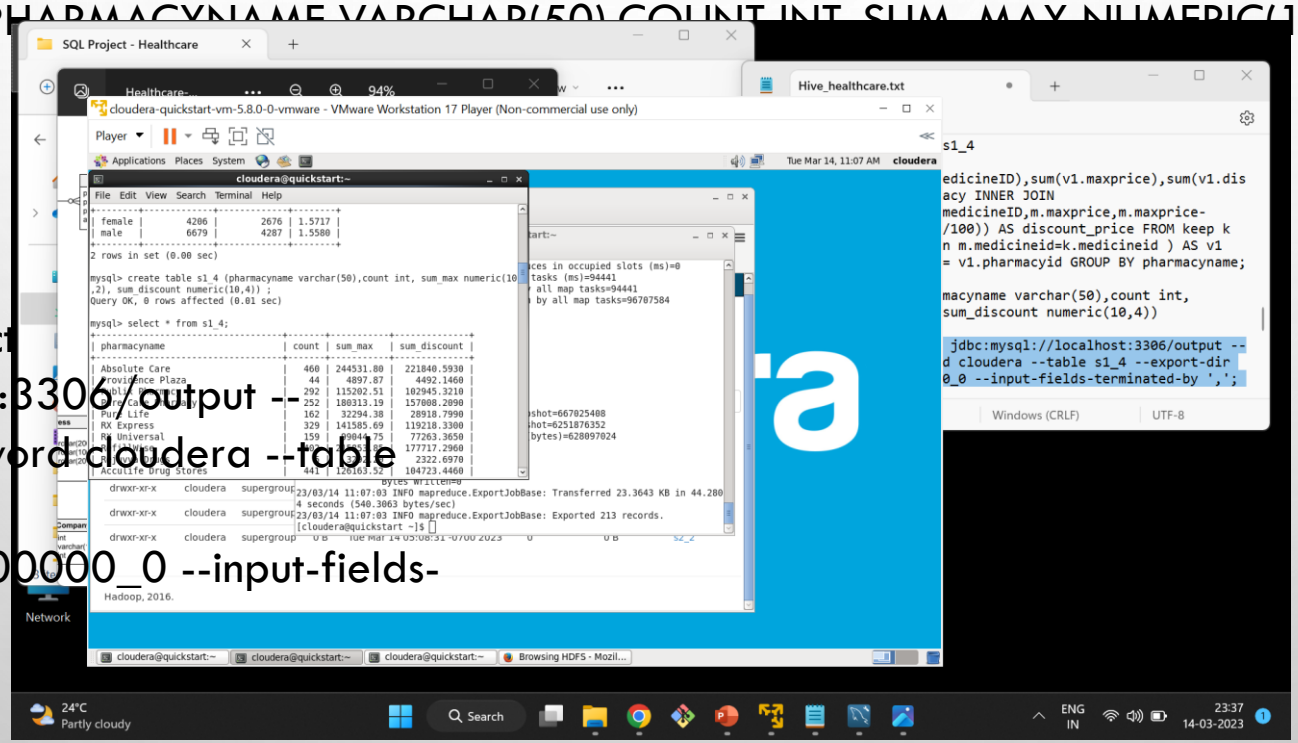


/*
Problem Statement S1_P4: The Healthcare department wants a report about the inventory of pharmacies.
Generate a report on their behalf that shows how many units of medicine each pharmacy has in their inventory,
the total maximum retail price of those medicines, and the total price of all the medicines after discount.
Note: discount field in keep signifies the percentage of discount on the maximum price.
*/

```
create external table s1_4 (pharmacyname varchar(50),count int, sum_max double, sum_discount double)
row format delimited
fields terminated by ','
lines terminated by '\n'
location '/user/output/s1_4';
```

```
INSERT OVERWRITE TABLE S1_4
SELECT PHARMACYNAME,COUNT(V1.MEDICINEID),SUM(V1.MAXPRICE),SUM(V1.DISCOUNT_PRICE) FROM
PHARMACY INNER JOIN
(SELECT K.PHARMACYID,M.MEDICINEID,M.MAXPRICE,M.MAXPRICE-(M.MAXPRICE*(K.DISCOUNT/100)) AS
DISCOUNT_PRICE FROM KEEP K INNER JOIN MEDICINE M ON M.MEDICINEID=K.MEDICINEID ) AS V1
ON PHARMACY.PHARMACYID = V1.PHARMACYID GROUP BY PHARMACYNAME;

CREATE TABLE S1_4 (PHARMACYNAME VARCHAR(50),COUNT INT,SUM_MAX NUMERIC(10,2), SUM_DISCOUNT
```

```
sqoop export --connect
jdbc:mysql://localhost:3306/output --
username root --password cloudera --table
s1_4 --export-dir
/user/output/s1_4/000000_0 --input-fields-
terminated-by ',';
```

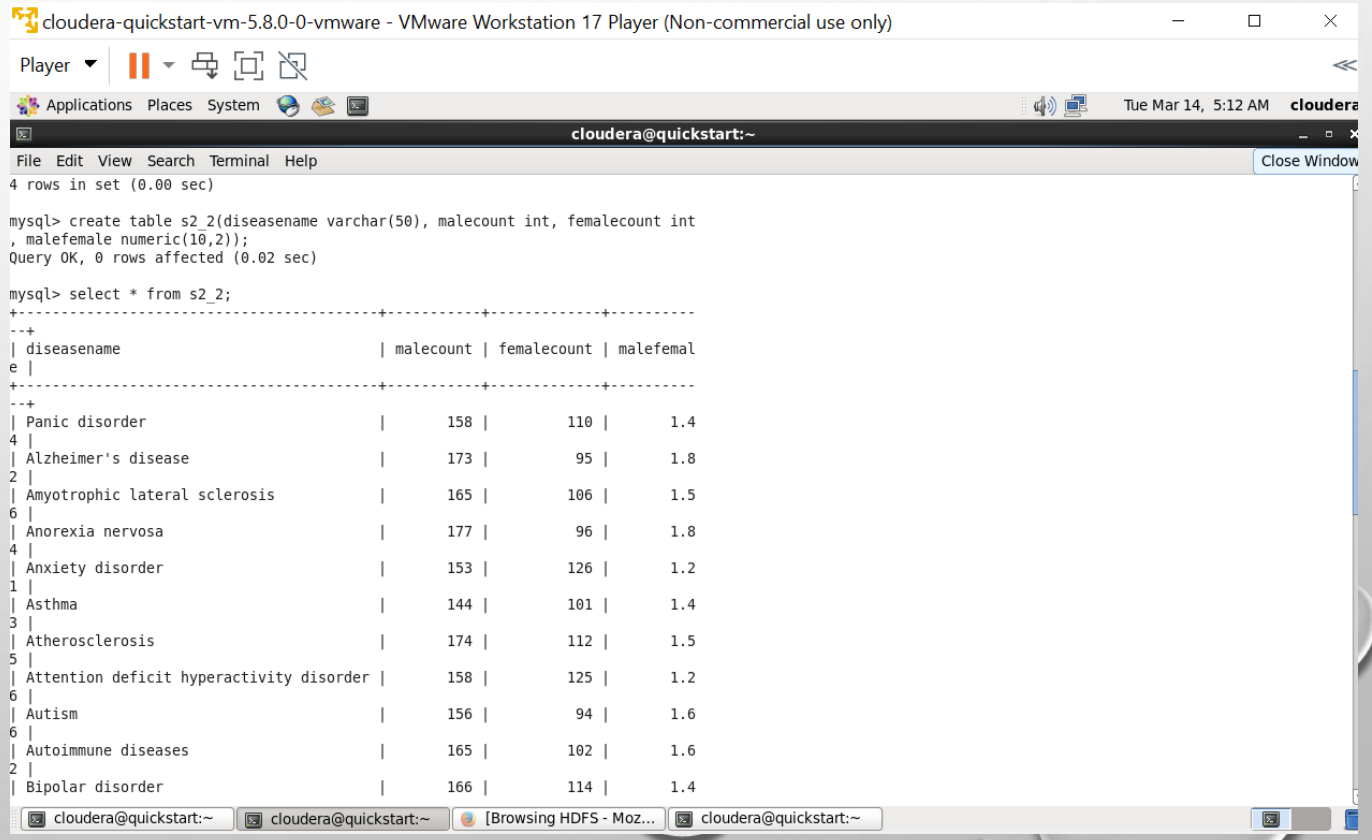


Problem Statement S2_P2: The State of Alabama (AL) is trying to manage its healthcare resources more efficiently.
For each city in their state, they need to identify the disease for which the maximum number of patients have gone for treatment.
Assist the state for this purpose.
Note: The state of Alabama is represented as AL in Address Table.

```
create external table s2_2 (diseasename varchar(50), malecount int, femalecount int, malefemale double) comment
'diseasename count'
row format delimited
fields terminated by ','
lines terminated by '\n'
location '/user/output/s2_2';
```

```
INSERT OVERWRITE TABLE S2_P2
select diseasename,COUNT(IF(gender = 'male', 1, null)) count_male,
COUNT(IF(gender = 'female', 1, NULL)) count_female,
COUNT(IF(gender = 'male', 1, NULL))/COUNT(IF(gender = 'female', 1, NULL)) as ratio
from
disease join treatment on treatment.diseaseid=disease.diseaseid
join patient on patient.patientid=treatment.patientid
join person on patient.patientid=person.personid
group by diseasename
order by diseasename ;
```

```
sqoop export --connect
jdbc:mysql://localhost:3306/output --
username root --P --table s2_P2 --export-
dir /user/output/S2_P2/000000_0 --
input-fields-terminated-by ',';
```



Problem Statement S4_P1:

“HealthDirect” pharmacy finds it difficult to deal with the product type of medicine being displayed in numerical form,
they want the product type in words. Also, they want to filter the medicines based on tax criteria. Display only the medicines of product categories 1, 2, and 3 for medicines that come under tax category I and medicines of product
categories 4, 5, and 6 for medicines that come under tax category II. Write a SQL query to solve this problem.

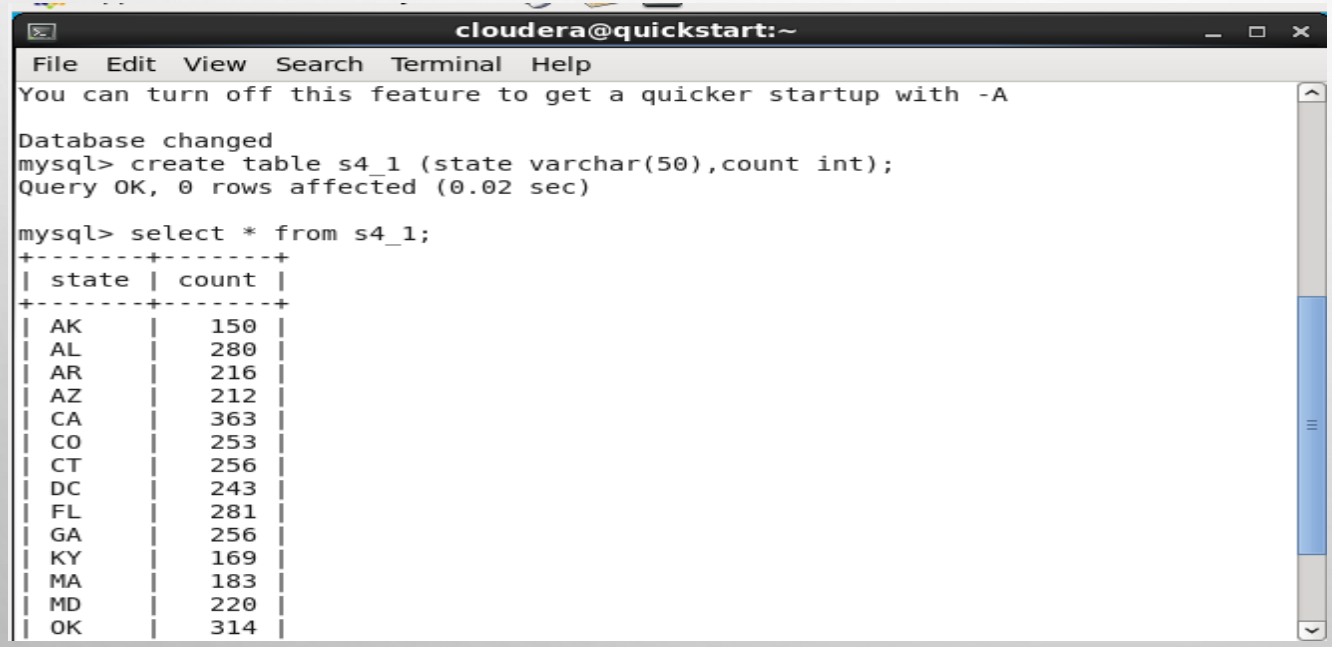
ProductType numerical form and ProductType in words are given by

1 - Generic,

2 - Patent,

```
create external table s4_P1 (state varchar(50),count int)
row format delimited
fields terminated by ','
lines terminated by '\n'
location '/user/output/s4_1';
```

```
insert overwrite table s4_P1
select a.state,count(*)
from treatment t left join claim c on t.claimid=c.claimid
left join patient p on t.patientid=p.patientid
left join person pe on p.patientid=pe.personid
left join address_part1 a on pe.addressid=a.addressid
where t.claimid IS NULL
group by a.state;
```




```
sqoop export --connect
jdbc:mysql://localhost:3306/output --
username root --password cloudera --table
s4_1 --export-dir
/user/output/s4_1/000000_0 --input-
fields-terminated-by ',';
```

Problem Statement S6_P4:

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.

```
create external table s6_p4 (state string, count int)
row format delimited
fields terminated by ','
lines terminated by '\n'
location '/user/output/s6_p4';
```

```
INSERT OVERWRITE TABLE s6_p4 select a.state,count(*)
from treatment t left join claim c on t.claimid=c.claimid
left join patient p on t.patientid=p.patientid
left join person pe on p.patientid=pe.personid
left join address a on pe.addressid=a.addressid
where t.claimid IS NULL
group by a.state;
```

```
Database changed
mysql> create table s6_p4 (state varchar(20), count int);
Query OK, 0 rows affected (0.01 sec)

mysql> select * from s6_p4;
+-------+-------+
| state | count |
+-------+-------+
| AK    |   150 |
| CO    |   253 |
| AL    |   280 |
| CT    |   256 |
| AR    |   216 |
| DC    |   243 |
| AZ    |   212 |
| FL    |   281 |
| CA    |   363 |
| GA    |   256 |
| KY    |   169 |
| MA    |   183 |
| MD    |   220 |
| OK    |   314 |
| TN    |   307 |
| VT    |   219 |
+-------+-------+
16 rows in set (0.00 sec)

mysql>
cloudera@quickstart:... cloudera@quickstart:~ cloudera@quickstart:~
```

```
create table s6_p4 (state varchar(20), count int);
```

```
sqoop export --connect
jdbc:mysql://localhost:3306/healthcareOP --username
root --password cloudera --table s6_p4 --export-dir
/user/output/s6_p4/000000_0 --input-fields-
terminated-by ',';
```