Hive Assignment

Architecture:

Source: ClientDB (MySQL)   jdbc:mysql://localhost:3306/healthcare

ETL: SQOOP (Importing all the tables from Source to Hive for Analysing via SQOOP (ETL tool))

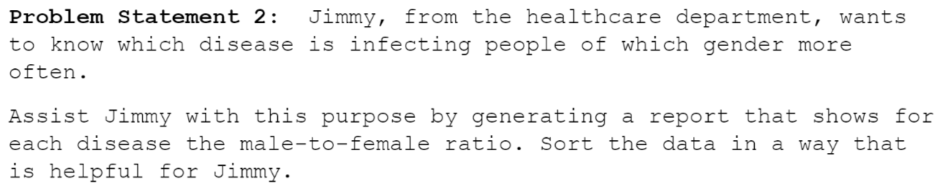
sqoop import-all-tables --connect jdbc:mysql://localhost/healthcare --username root \

--password cloudera --hive-import --hive-overwrite --m 1

All the tables have been imported to Hive database.

Analysing:

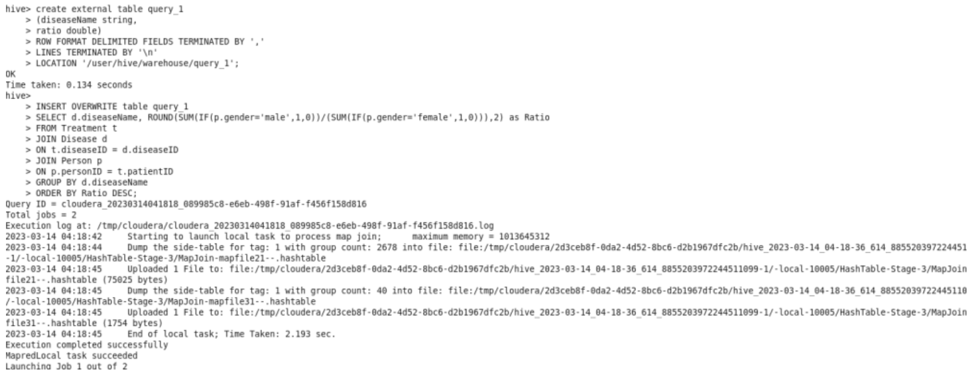
Query1 :



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

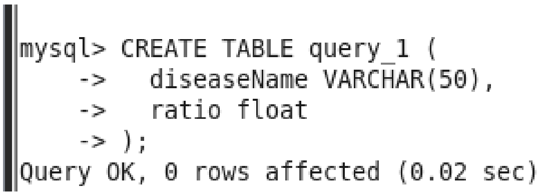
Assist Jimmy with this purpose by generating a report that shows for each disease the male-to-female ratio. Sort the data in a way that is helpful for Jimmy.

The Query Results are stored in External Table. (The data is stored in HDFS)



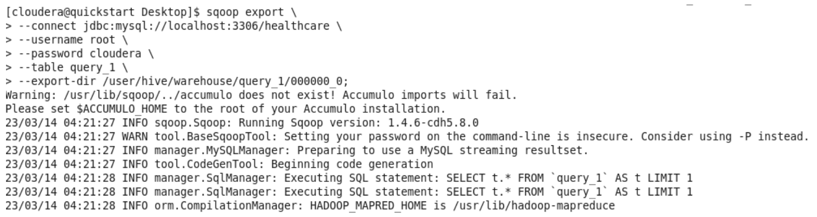
```
hive> create external table query_1
    > (diseaseName string,
    > ratio double)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/hive/warehouse/query_1';
OK
Time taken: 0.134 seconds
hive>
    > INSERT OVERWRITE table query_1
    > SELECT d.diseaseName, ROUND(SUM(IF(p.gender='male',1,0))/(SUM(IF(p.gender='female',1,0))),2) as Ratio
    > FROM Treatment t
    > JOIN Disease d
    > ON t.diseaseID = d.diseaseID
    > JOIN Person p
    > ON p.personID = t.patientID
    > GROUP BY d.diseaseName
    > ORDER BY Ratio DESC;
Query ID = cloudera_20230314041818_089985c8-e6eb-498f-91af-f456f158d816
Total jobs = 2
Execution log at: /tmp/cloudera/cloudera_20230314041818_089985c8-e6eb-498f-91af-f456f158d816.log
2023-03-14 04:18:42     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-14 04:18:44     Dump the side-table for tag: 1 with group count: 2678 into file: file:/tmp/cloudera/2d3ceb8f-0da2-4d52-8bc6-d2b1967dfc2b/hive_2023-03-14_04-18-36_614_885520397224451
-1/-local-10005/HashTable-Stage-3/MapJoin-mapfile21--.hashtable
2023-03-14 04:18:45     Uploaded 1 File to: file:/tmp/cloudera/2d3ceb8f-0da2-4d52-8bc6-d2b1967dfc2b/hive_2023-03-14_04-18-36_614_8855203972244511099-1/-local-10005/HashTable-Stage-3/MapJoin
file21--.hashtable (75025 bytes)
2023-03-14 04:18:45     Dump the side-table for tag: 1 with group count: 40 into file: file:/tmp/cloudera/2d3ceb8f-0da2-4d52-8bc6-d2b1967dfc2b/hive_2023-03-14_04-18-36_614_88552039722445110
/-local-10005/HashTable-Stage-3/MapJoin-mapfile31--.hashtable
2023-03-14 04:18:45     Uploaded 1 File to: file:/tmp/cloudera/2d3ceb8f-0da2-4d52-8bc6-d2b1967dfc2b/hive_2023-03-14_04-18-36_614_8855203972244511099-1/-local-10005/HashTable-Stage-3/MapJoin
file31--.hashtable (1754 bytes)
2023-03-14 04:18:45     End of local task; Time Taken: 2.193 sec.
Execution completed successfully
MapredLocal task succeeded
Launching Job 1 out of 2
```

Creating a Table in ClientDB (Target), since the table must be exported from Hive to MySOL using SQOOP.



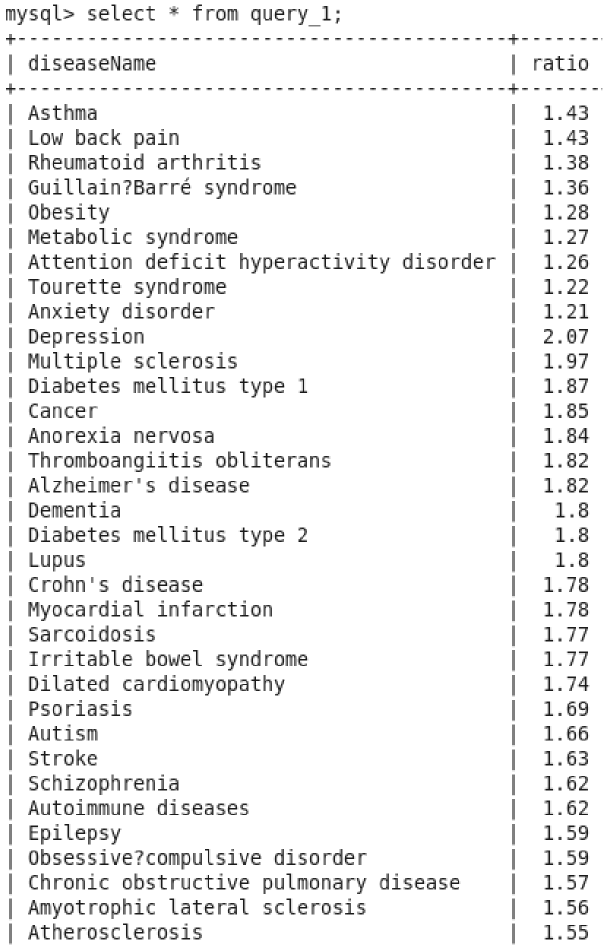
```
mysql> CREATE TABLE query_1 (
    ->   diseaseName VARCHAR(50),
    ->   ratio float
    -> );
Query OK, 0 rows affected (0.02 sec)
```

SQOOP Command: (The data from hive/warehouse is sent to the ClientDB (MySQL))



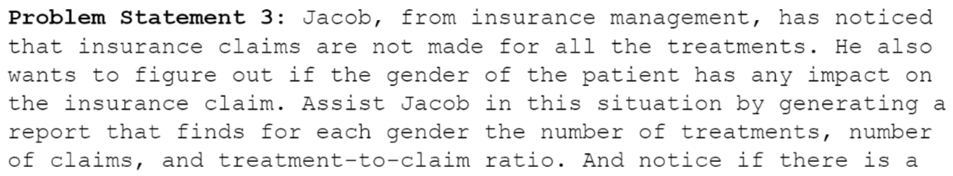
```
[cloudera@quickstart Desktop]$ sqoop export \
> --connect jdbc:mysql://localhost:3306/healthcare \
> --username root \
> --password cloudera \
> --table query_1 \
> --export-dir /user/hive/warehouse/query_1/000000_0;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/03/14 04:21:27 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
23/03/14 04:21:27 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
23/03/14 04:21:27 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/03/14 04:21:27 INFO tool.CodeGenTool: Beginning code generation
23/03/14 04:21:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `query_1` AS t LIMIT 1
23/03/14 04:21:28 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `query_1` AS t LIMIT 1
23/03/14 04:21:28 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
```

Here the data is exported from the same location in HDFS, while creating external table.

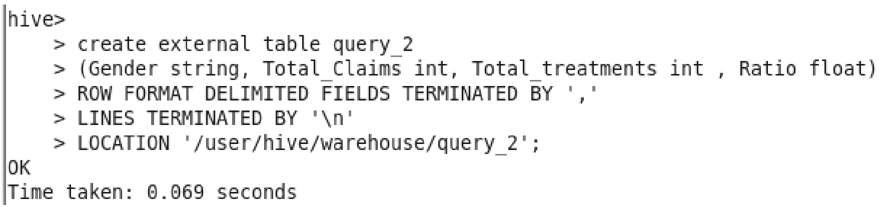


```
mysql> select * from query_1;
+------------------------------------------+-------
| diseaseName                              | ratio
+------------------------------------------+-------
| Asthma                                   |  1.43
| Low back pain                            |  1.43
| Rheumatoid arthritis                     |  1.38
| Guillain?Barré syndrome                  |  1.36
| Obesity                                  |  1.28
| Metabolic syndrome                       |  1.27
| Attention deficit hyperactivity disorder |  1.26
| Tourette syndrome                        |  1.22
| Anxiety disorder                         |  1.21
| Depression                               |  2.07
| Multiple sclerosis                       |  1.97
| Diabetes mellitus type 1                 |  1.87
| Cancer                                   |  1.85
| Anorexia nervosa                         |  1.84
| Thromboangiitis obliterans               |  1.82
| Alzheimer's disease                      |  1.82
| Dementia                                 |   1.8
| Diabetes mellitus type 2                 |   1.8
| Lupus                                    |   1.8
| Crohn's disease                          |  1.78
| Myocardial infarction                    |  1.78
| Sarcoidosis                              |  1.77
| Irritable bowel syndrome                 |  1.77
| Dilated cardiomyopathy                   |  1.74
| Psoriasis                                |  1.69
| Autism                                   |  1.66
| Stroke                                   |  1.63
| Schizophrenia                            |  1.62
| Autoimmune diseases                      |  1.62
| Epilepsy                                 |  1.59
| Obsessive?compulsive disorder            |  1.59
| Chronic obstructive pulmonary disease    |  1.57
| Amyotrophic lateral sclerosis            |  1.56
| Atherosclerosis                          |  1.55
```

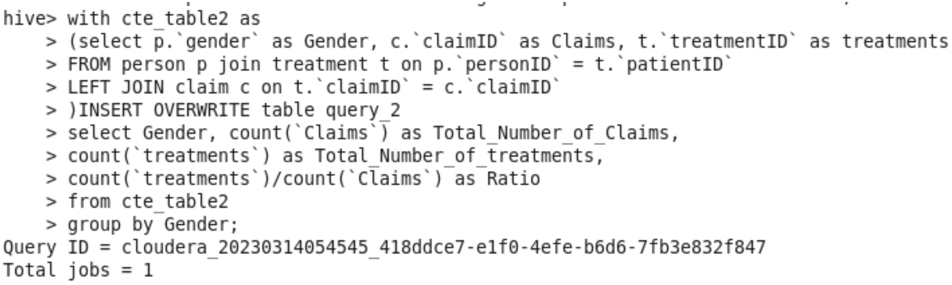
Query2:



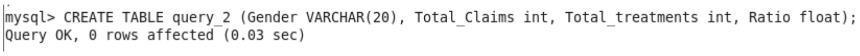
Problem Statement 3: Jacob, from insurance management, has noticed that insurance claims are not made for all the treatments. He also wants to figure out if the gender of the patient has any impact on the insurance claim. Assist Jacob in this situation by generating a report that finds for each gender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



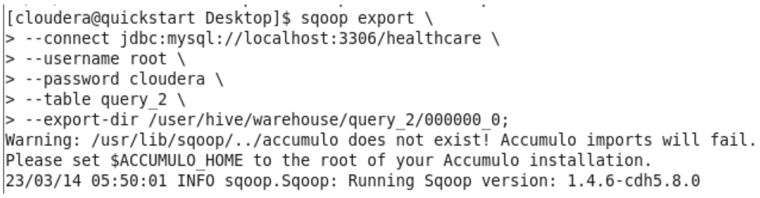
```
hive>
    > create external table query_2
    > (Gender string, Total_Claims int, Total_treatments int , Ratio float)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/hive/warehouse/query_2';
OK
Time taken: 0.069 seconds
```



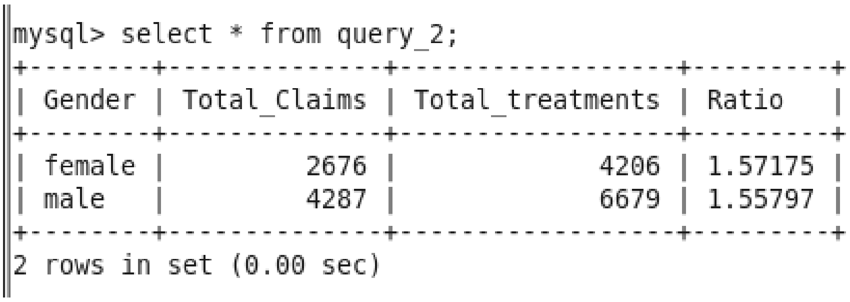
```
hive> with cte_table2 as
    > (select p.`gender` as Gender, c.`claimID` as Claims, t.`treatmentID` as treatments
    > FROM person p join treatment t on p.`personID` = t.`patientID`
    > LEFT JOIN claim c on t.`claimID` = c.`claimID`
    > )INSERT OVERWRITE table query_2
    > select Gender, count(`Claims`) as Total_Number_of_Claims,
    > count(`treatments`) as Total_Number_of_treatments,
    > count(`treatments`)/count(`Claims`) as Ratio
    > from cte_table2
    > group by Gender;
Query ID = cloudera_20230314054545_418ddce7-e1f0-4efe-b6d6-7fb3e832f847
Total jobs = 1
```



```
mysql> CREATE TABLE query_2 (Gender VARCHAR(20), Total_Claims int, Total_treatments int, Ratio float);
Query OK, 0 rows affected (0.03 sec)
```

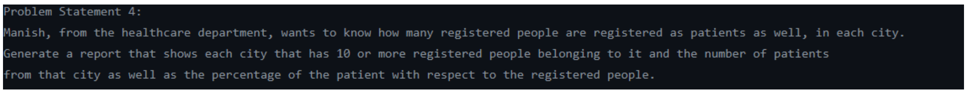


```
[cloudera@quickstart Desktop]$ sqoop export \
> --connect jdbc:mysql://localhost:3306/healthcare \
> --username root \
> --password cloudera \
> --table query_2 \
> --export-dir /user/hive/warehouse/query_2/000000_0;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/03/14 05:50:01 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
```

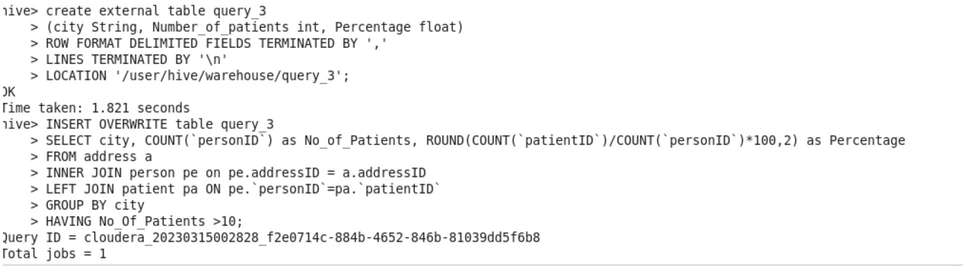


```
mysql> select * from query_2;
+--------+--------------+------------------+---------+
| Gender | Total_Claims | Total_treatments | Ratio   |
+--------+--------------+------------------+---------+
| female |         2676 |             4206 | 1.57175 |
| male   |         4287 |             6679 | 1.55797 |
+--------+--------------+------------------+---------+
2 rows in set (0.00 sec)
```

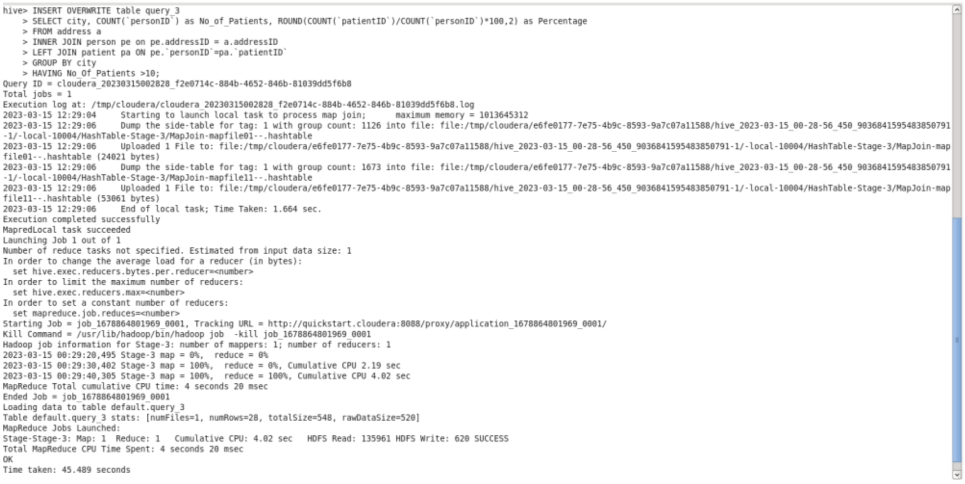
Query3:



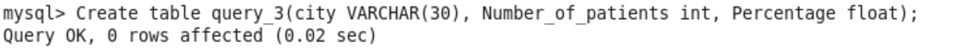
Problem Statement 4:
Manish, from the healthcare department, wants to know how many registered people are registered as patients as well, in each city.
Generate a report that shows each city that has 10 or more registered people belonging to it and the number of patients
from that city as well as the percentage of the patient with respect to the registered people.



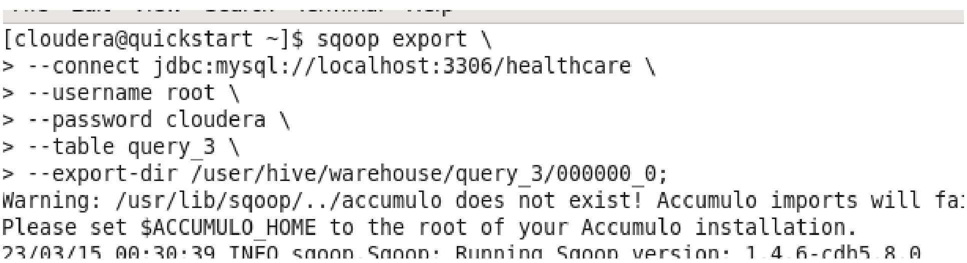
```
ıive> create external table query_3
    > (city String, Number_of_patients int, Percentage float)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/hive/warehouse/query_3';
ЭK
Time taken: 1.821 seconds
ıive> INSERT OVERWRITE table query_3
    > SELECT city, COUNT(`personID`) as No_of_Patients, ROUND(COUNT(`patientID`)/COUNT(`personID`)*100,2) as Percentage
    > FROM address a
    > INNER JOIN person pe on pe.addressID = a.addressID
    > LEFT JOIN patient pa ON pe.`personID`=pa.`patientID`
    > GROUP BY city
    > HAVING No_Of_Patients >10;
Query ID = cloudera_20230315002828_f2e0714c-884b-4652-846b-81039dd5f6b8
Total jobs = 1
```



```
hive> INSERT OVERWRITE table query_3
    > SELECT city, COUNT(`personID`) as No_of_Patients, ROUND(COUNT(`patientID`)/COUNT(`personID`)*100,2) as Percentage
    > FROM address a
    > INNER JOIN person pe on pe.addressID = a.addressID
    > LEFT JOIN patient pa ON pe.`personID`=pa.`patientID`
    > GROUP BY city
    > HAVING No_Of_Patients >10;
Query ID = cloudera_20230315002828_f2e0714c-884b-4652-846b-81039dd5f6b8
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230315002828_f2e0714c-884b-4652-846b-81039dd5f6b8.log
2023-03-15 12:29:04     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-15 12:29:06     Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/e6fe0177-7e75-4b9c-8593-9a7c07a11588/hive_2023-03-15_00-28-56_450_9036841595483850791
-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile01--.hashtable
2023-03-15 12:29:06     Uploaded 1 File to: file:/tmp/cloudera/e6fe0177-7e75-4b9c-8593-9a7c07a11588/hive_2023-03-15_00-28-56_450_9036841595483850791-1/-local-10004/HashTable-Stage-3/MapJoin-map
file01--.hashtable (24021 bytes)
2023-03-15 12:29:06     Dump the side-table for tag: 1 with group count: 1673 into file: file:/tmp/cloudera/e6fe0177-7e75-4b9c-8593-9a7c07a11588/hive_2023-03-15_00-28-56_450_9036841595483850791
-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile11--.hashtable
2023-03-15 12:29:06     Uploaded 1 File to: file:/tmp/cloudera/e6fe0177-7e75-4b9c-8593-9a7c07a11588/hive_2023-03-15_00-28-56_450_9036841595483850791-1/-local-10004/HashTable-Stage-3/MapJoin-map
file11--.hashtable (53061 bytes)
2023-03-15 12:29:06     End of local task; Time Taken: 1.664 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_1678864801969_0001, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678864801969_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678864801969_0001
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2023-03-15 00:29:20,495 Stage-3 map = 0%,  reduce = 0%
2023-03-15 00:29:30,402 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 2.19 sec
2023-03-15 00:29:40,305 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 4.02 sec
MapReduce Total cumulative CPU time: 4 seconds 20 msec
Ended Job = job_1678864801969_0001
Loading data to table default.query_3
Table default.query_3 stats: [numFiles=1, numRows=28, totalSize=548, rawDataSize=520]
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 4.02 sec   HDFS Read: 135961 HDFS Write: 620 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 20 msec
OK
Time taken: 45.489 seconds
```

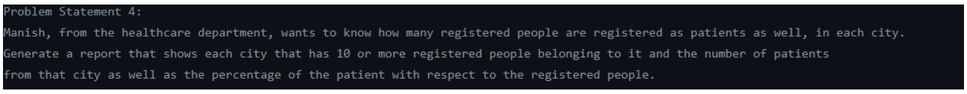


```
mysql> Create table query_3(city VARCHAR(30), Number_of_patients int, Percentage float);
Query OK, 0 rows affected (0.02 sec)
```

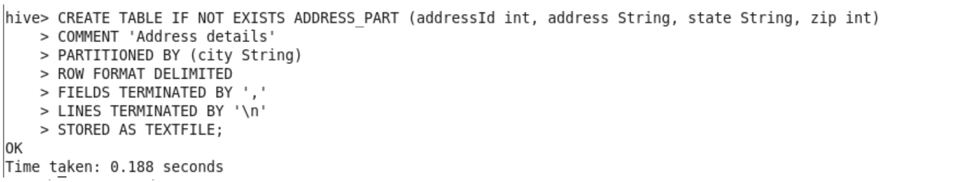


```
[cloudera@quickstart ~]$ sqoop export \
> --connect jdbc:mysql://localhost:3306/healthcare \
> --username root \
> --password cloudera \
> --table query_3 \
> --export-dir /user/hive/warehouse/query_3/000000_0;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fa
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/03/15 00:30:39 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
```

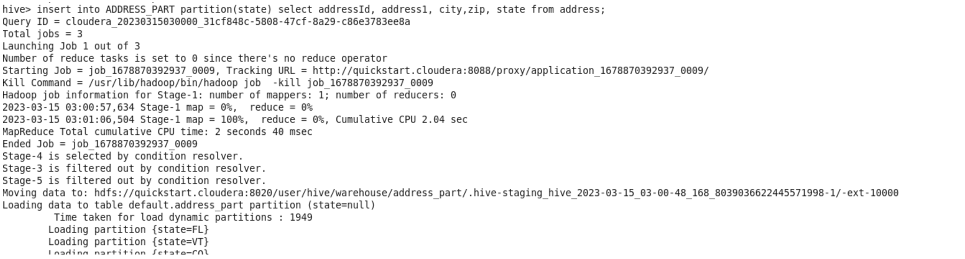
Query4: (Previous Query with Partitions)



```
Problem Statement 4:
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.
```

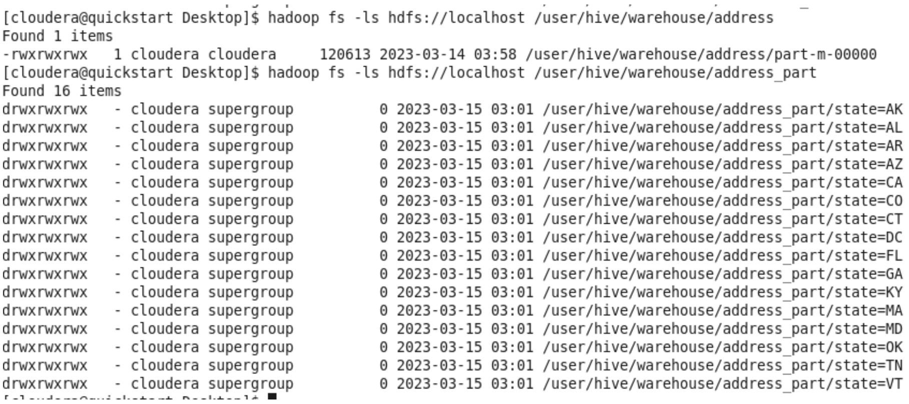


```
hive> CREATE TABLE IF NOT EXISTS ADDRESS_PART (addressId int, address String, state String, zip int)
    > COMMENT 'Address details'
    > PARTITIONED BY (city String)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > STORED AS TEXTFILE;
OK
Time taken: 0.188 seconds
```

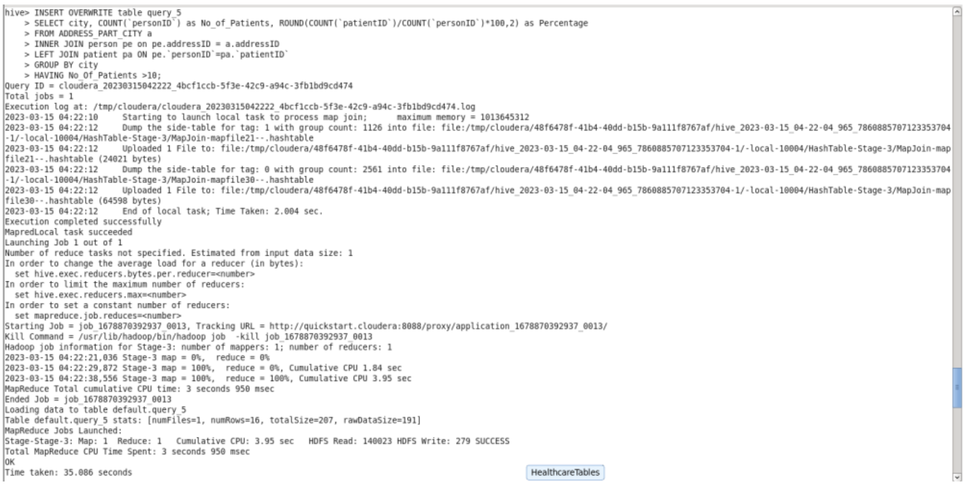


```
hive> insert into ADDRESS_PART partition(state) select addressId, address1, city,zip, state from address;
Query ID = cloudera_20230315030000_31cf848c-5808-47cf-8a29-c86e3783ee8a
Total jobs = 3
Launching Job 1 out of 3
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1678870392937_0009, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678870392937_0009/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678870392937_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2023-03-15 03:00:57,634 Stage-1 map = 0%,  reduce = 0%
2023-03-15 03:01:06,504 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.04 sec
MapReduce Total cumulative CPU time: 2 seconds 40 msec
Ended Job = job_1678870392937_0009
Stage-4 is selected by condition resolver.
Stage-3 is filtered out by condition resolver.
Stage-5 is filtered out by condition resolver.
Moving data to: hdfs://quickstart.cloudera:8020/user/hive/warehouse/address_part/.hive-staging_hive_2023-03-15_03-00-48_168_8039036622445571998-1/-ext-10000
Loading data to table default.address_part partition (state=null)
         Time taken for load dynamic partitions : 1949
        Loading partition {state=FL}
        Loading partition {state=VT}
        Loading partition {state=CO}
```

The inserted data has been stored in different directories in hive/warehouse.



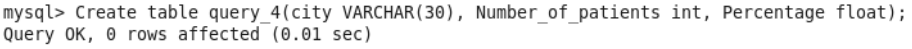
```
[cloudera@quickstart Desktop]$ hadoop fs -ls hdfs://localhost /user/hive/warehouse/address
Found 1 items
-rwxrwxrwx   1 cloudera cloudera     120613 2023-03-14 03:58 /user/hive/warehouse/address/part-m-00000
[cloudera@quickstart Desktop]$ hadoop fs -ls hdfs://localhost /user/hive/warehouse/address_part
Found 16 items
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=AK
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=AL
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=AR
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=AZ
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=CA
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=CO
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=CT
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=DC
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=FL
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=GA
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=KY
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=MA
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=MD
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=OK
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=TN
drwxrwxrwx   - cloudera supergroup          0 2023-03-15 03:01 /user/hive/warehouse/address_part/state=VT
```



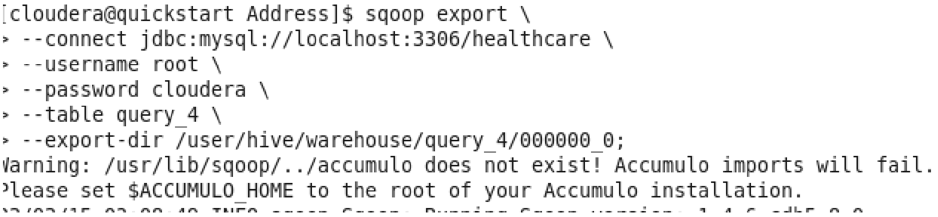
```
hive> INSERT OVERWRITE table query_5
    > SELECT city, COUNT(`personID`) as No_of_Patients, ROUND(COUNT(`patientID`)/COUNT(`personID`)*100,2) as Percentage
    > FROM ADDRESS_PART_CITY a
    > INNER JOIN person pe on pe.addressID = a.addressID
    > LEFT JOIN patient pa ON pe.`personID`=pa.`patientID`
    > GROUP BY city
    > HAVING No_Of_Patients >10;
Query ID = cloudera_20230315042222_4bcf1ccb-5f3e-42c9-a94c-3fb1bd9cd474
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230315042222_4bcf1ccb-5f3e-42c9-a94c-3fb1bd9cd474.log
2023-03-15 04:22:10     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-15 04:22:12     Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/48f6478f-41b4-40dd-b15b-9a111f8767af/hive_2023-03-15_04-22-04_965_7860885707123353704
-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile21--.hashtable
2023-03-15 04:22:12     Uploaded 1 File to: file:/tmp/cloudera/48f6478f-41b4-40dd-b15b-9a111f8767af/hive_2023-03-15_04-22-04_965_7860885707123353704-1/-local-10004/HashTable-Stage-3/MapJoin-map
file21--.hashtable (24021 bytes)
2023-03-15 04:22:12     Dump the side-table for tag: 0 with group count: 2561 into file: file:/tmp/cloudera/48f6478f-41b4-40dd-b15b-9a111f8767af/hive_2023-03-15_04-22-04_965_7860885707123353704
-1/-local-10004/HashTable-Stage-3/MapJoin-mapfile30--.hashtable
2023-03-15 04:22:12     Uploaded 1 File to: file:/tmp/cloudera/48f6478f-41b4-40dd-b15b-9a111f8767af/hive_2023-03-15_04-22-04_965_7860885707123353704-1/-local-10004/HashTable-Stage-3/MapJoin-map
file30--.hashtable (64598 bytes)
2023-03-15 04:22:12     End of local task; Time Taken: 2.004 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_1678870392937_0013, Tracking URL = http://quickstart.cloudera:8088/proxy/application_1678870392937_0013/
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_1678870392937_0013
Hadoop job information for Stage-3: number of mappers: 1; number of reducers: 1
2023-03-15 04:22:21,036 Stage-3 map = 0%,  reduce = 0%
2023-03-15 04:22:29,872 Stage-3 map = 100%,  reduce = 0%, Cumulative CPU 1.84 sec
2023-03-15 04:22:38,556 Stage-3 map = 100%,  reduce = 100%, Cumulative CPU 3.95 sec
MapReduce Total cumulative CPU time: 3 seconds 950 msec
Ended Job = job_1678870392937_0013
Loading data to table default.query_5
Table default.query_5 stats: [numFiles=1, numRows=16, totalSize=207, rawDataSize=191]
MapReduce Jobs Launched:
Stage-Stage-3: Map: 1  Reduce: 1   Cumulative CPU: 3.95 sec   HDFS Read: 140023 HDFS Write: 279 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 950 msec
OK
Time taken: 35.086 seconds
```

The result has been reduced by 10 seconds with partitions added in the address table.

The max partitions supported by HIVE is 100. If it exceeds 100 partitions it throws number of dynamic partitions exceeded.



```
mysql> Create table query_4(city VARCHAR(30), Number_of_patients int, Percentage float);
Query OK, 0 rows affected (0.01 sec)
```



```
[cloudera@quickstart Address]$ sqoop export \
> --connect jdbc:mysql://localhost:3306/healthcare \
> --username root \
> --password cloudera \
> --table query_4 \
> --export-dir /user/hive/warehouse/query_4/000000_0;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
```

<u>**Query5:**</u>

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.

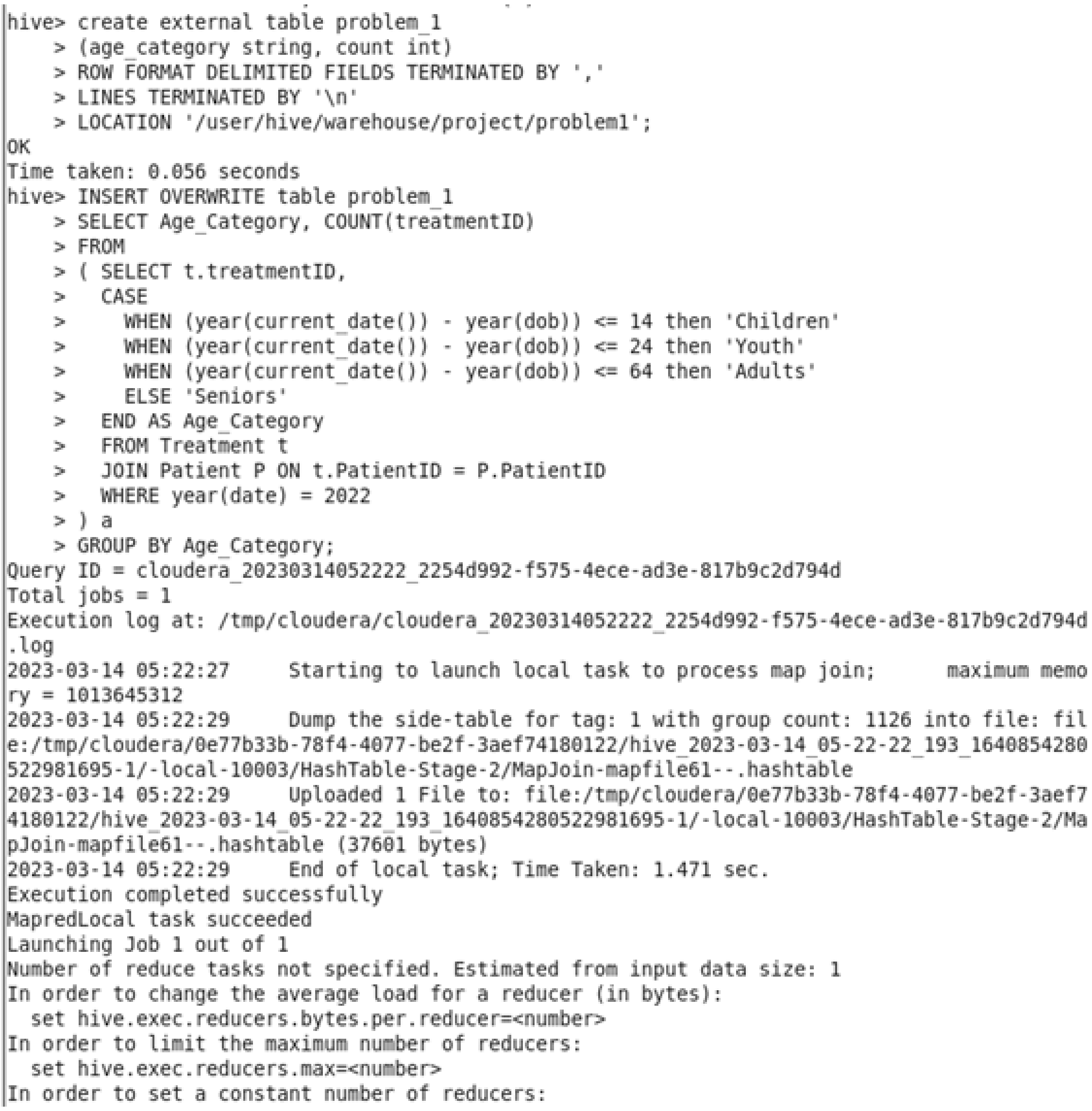
Create External Table

create external table problem_1

```
(age_category string, count int)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
LOCATION '/user/hive/warehouse/project/problem1';
```

Insert Data Into External Table In Hive

```
INSERT OVERWRITE table problem_1
SELECT Age_Category, COUNT(treatmentID)
FROM
( SELECT t.treatmentID,
  CASE
    WHEN (year(current_date()) - year(dob)) <= 14 then 'Children'
    WHEN (year(current_date()) - year(dob)) <= 24 then 'Youth'
    WHEN (year(current_date()) - year(dob)) <= 64 then 'Adults'
    ELSE 'Seniors'
  END AS Age_Category
  FROM Treatment t
  JOIN Patient P ON t.PatientID = P.PatientID
  WHERE year(date) = 2022
) a
GROUP BY Age_Category;
```



```
hive> create external table problem_1
    > (age_category string, count int)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/hive/warehouse/project/problem1';
OK
Time taken: 0.056 seconds
hive> INSERT OVERWRITE table problem_1
    > SELECT Age_Category, COUNT(treatmentID)
    > FROM
    > ( SELECT t.treatmentID,
    >   CASE
    >     WHEN (year(current_date()) - year(dob)) <= 14 then 'Children'
    >     WHEN (year(current_date()) - year(dob)) <= 24 then 'Youth'
    >     WHEN (year(current_date()) - year(dob)) <= 64 then 'Adults'
    >     ELSE 'Seniors'
    >   END AS Age_Category
    >   FROM Treatment t
    >   JOIN Patient P ON t.PatientID = P.PatientID
    >   WHERE year(date) = 2022
    > ) a
    > GROUP BY Age_Category;
Query ID = cloudera_20230314052222_2254d992-f575-4ece-ad3e-817b9c2d794d
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230314052222_2254d992-f575-4ece-ad3e-817b9c2d794d
.log
2023-03-14 05:22:27     Starting to launch local task to process map join;      maximum memo
ry = 1013645312
2023-03-14 05:22:29     Dump the side-table for tag: 1 with group count: 1126 into file: fil
e:/tmp/cloudera/0e77b33b-78f4-4077-be2f-3aef74180122/hive_2023-03-14_05-22-22_193_1640854280
522981695-1/-local-10003/HashTable-Stage-2/MapJoin-mapfile61--.hashtable
2023-03-14 05:22:29     Uploaded 1 File to: file:/tmp/cloudera/0e77b33b-78f4-4077-be2f-3aef7
4180122/hive_2023-03-14_05-22-22_193_1640854280522981695-1/-local-10003/HashTable-Stage-2/Ma
pJoin-mapfile61--.hashtable (37601 bytes)
2023-03-14 05:22:29     End of local task; Time Taken: 1.471 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:
```

<u>Create Output Table in Client DB</u>

CREATE TABLE problem_1(

  age_category VARCHAR(20),

  count int

);

<u>Move Data to Client DB using Sqoop Export</u>

sqoop export \

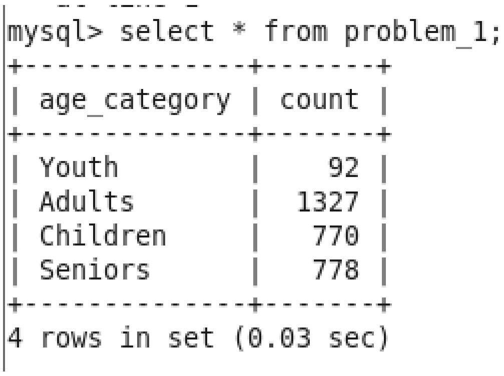
--connect jdbc:mysql://localhost:3306/healthcare \

--username root \

--password cloudera \

--table problem_1 \

--export-dir /user/hive/warehouse/project/problem1/000000_0;



```
mysql> select * from problem_1;
+--------------+-------+
| age_category | count |
+--------------+-------+
| Youth        |    92 |
| Adults       |  1327 |
| Children     |   770 |
| Seniors      |   778 |
+--------------+-------+
4 rows in set (0.03 sec)
```

—---

QUERY_6

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.

Create External Table

create external table query_2

(diseaseName string, ratio double)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

LOCATION '/user/hive/warehouse/project/problem2';

Insert Data Into External Table In Hive

INSERT OVERWRITE table query_2

SELECT d.diseaseName, ROUND(SUM(IF(p.gender='male',1,0))/(SUM(IF(p.gender='female',1,0))),2) as Ratio

FROM Treatment t

JOIN Disease d

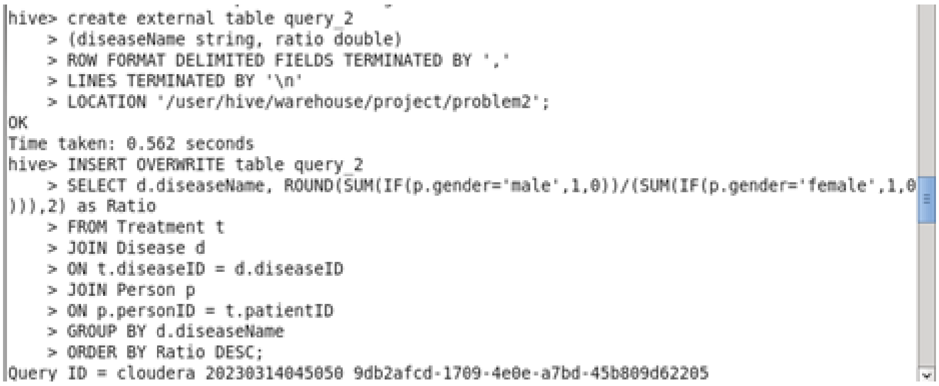
ON t.diseaseID = d.diseaseID

JOIN Person p

ON p.personID = t.patientID

GROUP BY d.diseaseName

ORDER BY Ratio DESC;



```
hive> create external table query_2
    > (diseaseName string, ratio double)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/hive/warehouse/project/problem2';
OK
Time taken: 0.562 seconds
hive> INSERT OVERWRITE table query_2
    > SELECT d.diseaseName, ROUND(SUM(IF(p.gender='male',1,0))/(SUM(IF(p.gender='female',1,0
))),2) as Ratio
    > FROM Treatment t
    > JOIN Disease d
    > ON t.diseaseID = d.diseaseID
    > JOIN Person p
    > ON p.personID = t.patientID
    > GROUP BY d.diseaseName
    > ORDER BY Ratio DESC;
Query ID = cloudera_20230314045050_9db2afcd-1709-4e0e-a7bd-45b809d62205
```

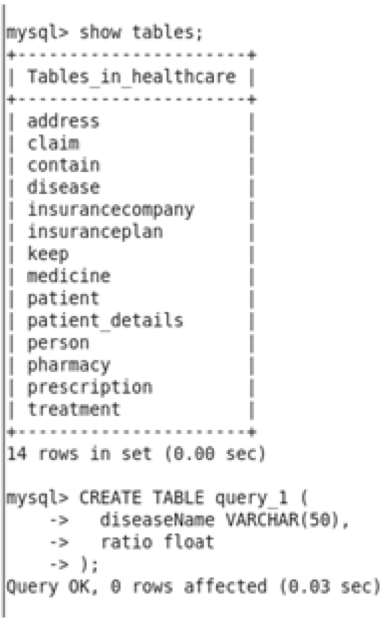
Create Output Table in Client DB

CREATE TABLE query_2 (

  diseaseName VARCHAR(50),

  ratio float

);



```
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> CREATE TABLE query_1 (
    ->   diseaseName VARCHAR(50),
    ->   ratio float
    -> );
Query OK, 0 rows affected (0.03 sec)
```

Move Data to Client DB using Sqoop Export

sqoop export \

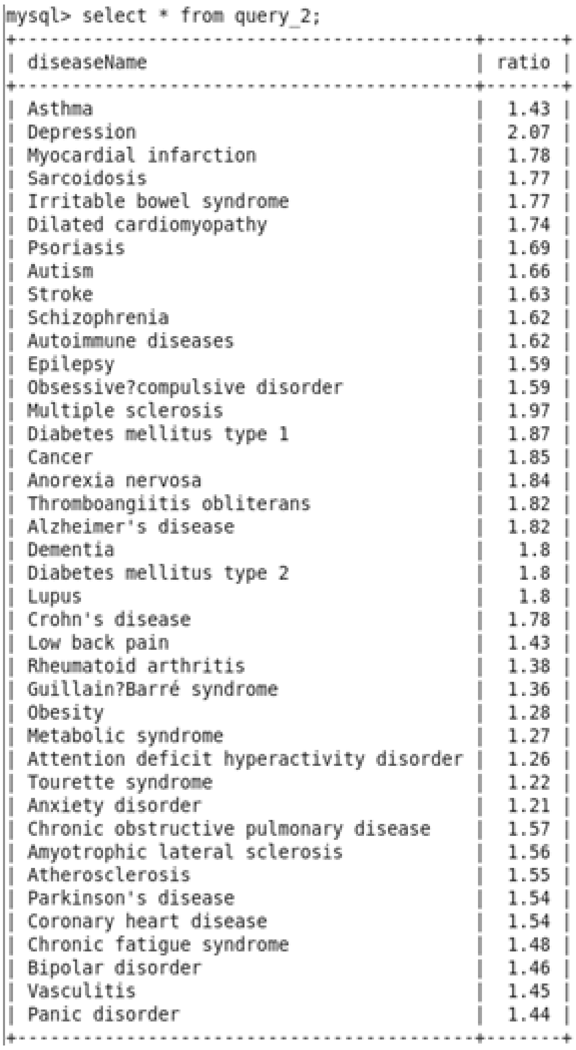
--connect jdbc:mysql://localhost:3306/healthcare \

--username root \

--password cloudera \

--table query_2 \

--export-dir /user/hive/warehouse/project/problem2/000000_0;



```
mysql> select * from query_2;
+-----------------------------------------+-------+
| diseaseName                             | ratio |
+-----------------------------------------+-------+
| Asthma                                  |  1.43 |
| Depression                              |  2.07 |
| Myocardial infarction                   |  1.78 |
| Sarcoidosis                             |  1.77 |
| Irritable bowel syndrome                |  1.77 |
| Dilated cardiomyopathy                  |  1.74 |
| Psoriasis                               |  1.69 |
| Autism                                  |  1.66 |
| Stroke                                  |  1.63 |
| Schizophrenia                           |  1.62 |
| Autoimmune diseases                     |  1.62 |
| Epilepsy                                |  1.59 |
| Obsessive?compulsive disorder           |  1.59 |
| Multiple sclerosis                      |  1.97 |
| Diabetes mellitus type 1                |  1.87 |
| Cancer                                  |  1.85 |
| Anorexia nervosa                        |  1.84 |
| Thromboangiitis obliterans              |  1.82 |
| Alzheimer's disease                     |  1.82 |
| Dementia                                |   1.8 |
| Diabetes mellitus type 2                |   1.8 |
| Lupus                                   |   1.8 |
| Crohn's disease                         |  1.78 |
| Low back pain                           |  1.43 |
| Rheumatoid arthritis                    |  1.38 |
| Guillain?Barré syndrome                 |  1.36 |
| Obesity                                 |  1.28 |
| Metabolic syndrome                      |  1.27 |
| Attention deficit hyperactivity disorder |  1.26 |
| Tourette syndrome                       |  1.22 |
| Anxiety disorder                        |  1.21 |
| Chronic obstructive pulmonary disease   |  1.57 |
| Amyotrophic lateral sclerosis           |  1.56 |
| Atherosclerosis                         |  1.55 |
| Parkinson's disease                     |  1.54 |
| Coronary heart disease                  |  1.54 |
| Chronic fatigue syndrome                |  1.48 |
| Bipolar disorder                        |  1.46 |
| Vasculitis                              |  1.45 |
| Panic disorder                          |  1.44 |
+-----------------------------------------+-------+
```

—--

QUERY7

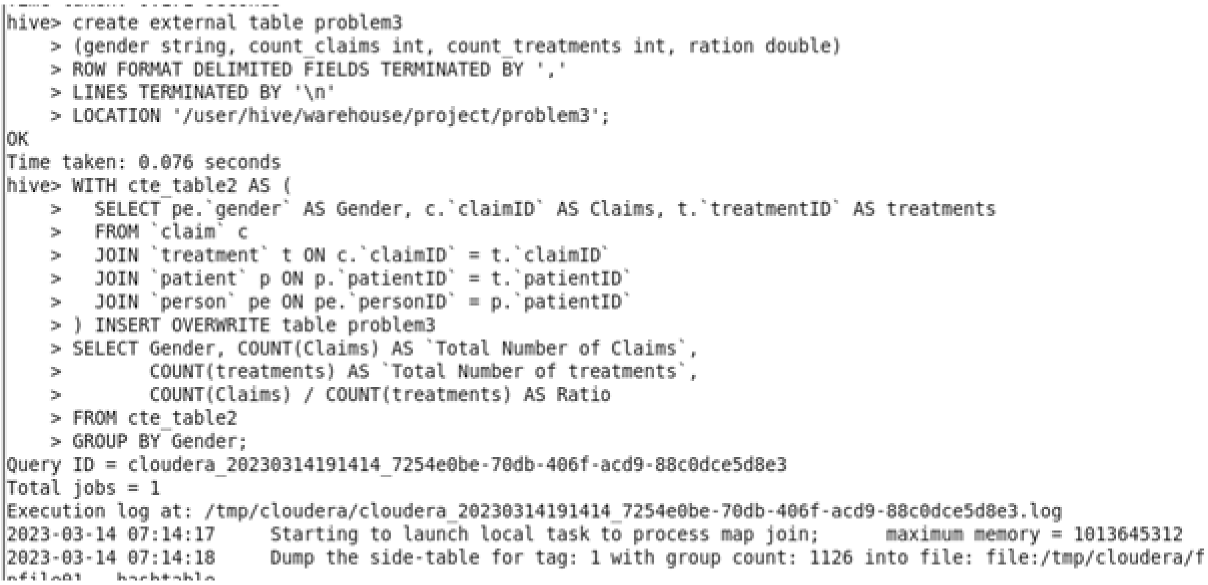
Problem Statement 3: Jacob, from insurance management, has noticed that insurance claims are not made for all the treatments. He also wants to figure out if the gender of the patient has any impact on the insurance claim. Assist Jacob in this situation by generating a report that finds for each gender 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.

Create External Table

```
create external table problem3
(gender string, count_claims int, count_treatments int, ration double)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LINES TERMINATED BY '\n'
LOCATION '/user/hive/warehouse/project/problem3';
```

Insert Data Into External Table In Hive

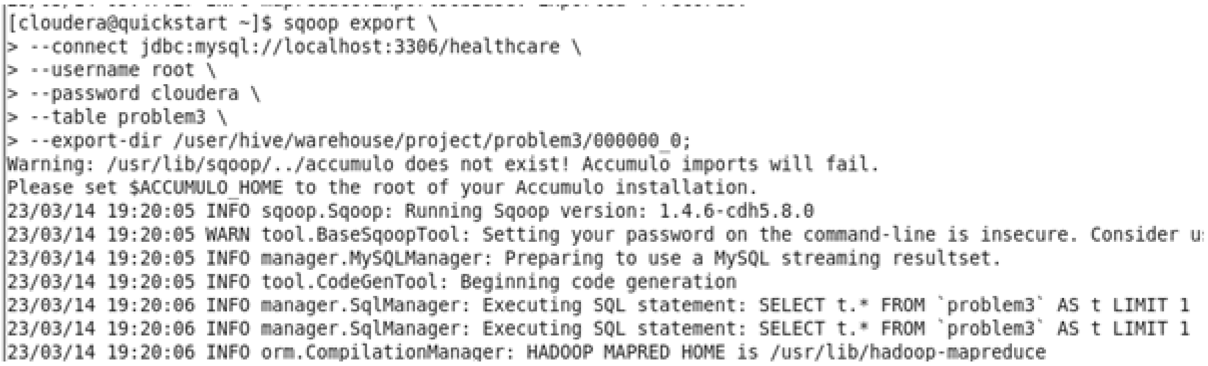
```
WITH cte_table2 AS (
  SELECT pe.`gender` AS Gender, c.`claimID` AS Claims, t.`treatmentID` AS treatments
  FROM `claim` c
  JOIN `treatment` t ON c.`claimID` = t.`claimID`
  JOIN `patient` p ON p.`patientID` = t.`patientID`
  JOIN `person` pe ON pe.`personID` = p.`patientID`
) INSERT OVERWRITE table problem3
SELECT Gender, COUNT(Claims) AS `Total Number of Claims`,
       COUNT(treatments) AS `Total Number of treatments`,
       COUNT(Claims) / COUNT(treatments) AS Ratio
FROM cte_table2
GROUP BY Gender;
```



```
hive> create external table problem3
    > (gender string, count_claims int, count_treatments int, ration double)
    > ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
    > LINES TERMINATED BY '\n'
    > LOCATION '/user/hive/warehouse/project/problem3';
OK
Time taken: 0.076 seconds
hive> WITH cte_table2 AS (
    >   SELECT pe.`gender` AS Gender, c.`claimID` AS Claims, t.`treatmentID` AS treatments
    >   FROM `claim` c
    >   JOIN `treatment` t ON c.`claimID` = t.`claimID`
    >   JOIN `patient` p ON p.`patientID` = t.`patientID`
    >   JOIN `person` pe ON pe.`personID` = p.`patientID`
    > ) INSERT OVERWRITE table problem3
    > SELECT Gender, COUNT(Claims) AS `Total Number of Claims`,
    >        COUNT(treatments) AS `Total Number of treatments`,
    >        COUNT(Claims) / COUNT(treatments) AS Ratio
    > FROM cte_table2
    > GROUP BY Gender;
Query ID = cloudera_20230314191414_7254e0be-70db-406f-acd9-88c0dce5d8e3
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230314191414_7254e0be-70db-406f-acd9-88c0dce5d8e3.log
2023-03-14 07:14:17     Starting to launch local task to process map join;      maximum memory = 1013645312
2023-03-14 07:14:18     Dump the side-table for tag: 1 with group count: 1126 into file: file:/tmp/cloudera/f
```

Create Output Table in Client DB

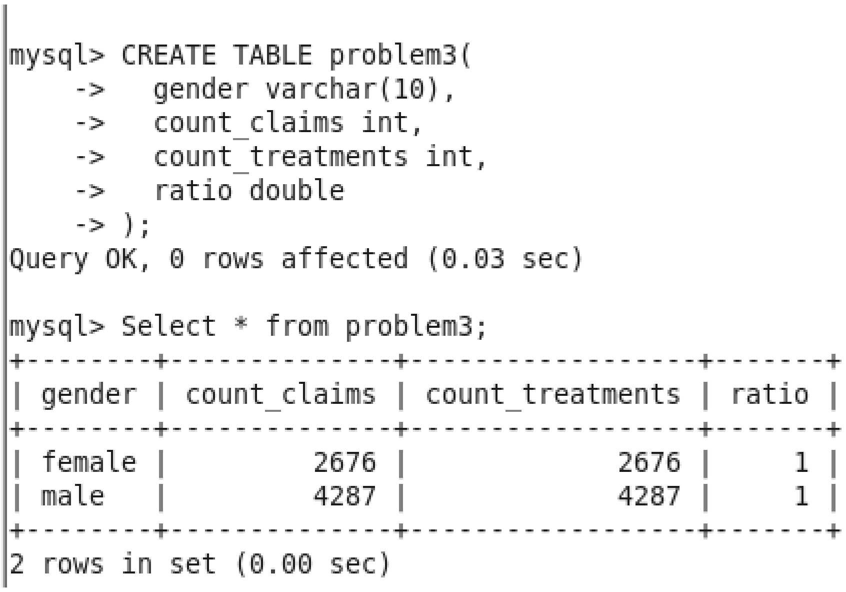
```
CREATE TABLE problem3(
  gender varchar(10),
  count_claims int,
  count_treatments int,
  ratio double
);
```



```
[cloudera@quickstart ~]$ sqoop export \
> --connect jdbc:mysql://localhost:3306/healthcare \
> --username root \
> --password cloudera \
> --table problem3 \
> --export-dir /user/hive/warehouse/project/problem3/000000_0;
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
23/03/14 19:20:05 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.8.0
23/03/14 19:20:05 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider u
23/03/14 19:20:05 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
23/03/14 19:20:05 INFO tool.CodeGenTool: Beginning code generation
23/03/14 19:20:06 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `problem3` AS t LIMIT 1
23/03/14 19:20:06 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `problem3` AS t LIMIT 1
23/03/14 19:20:06 INFO orm.CompilationManager: HADOOP MAPRED HOME is /usr/lib/hadoop-mapreduce
```

Move Data to Client DB using Sqoop Export

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table problem_1 \
--export-dir /user/hive/warehouse/project/problem1/000000_0;
```



```
mysql> CREATE TABLE problem3(
    ->   gender varchar(10),
    ->   count_claims int,
    ->   count_treatments int,
    ->   ratio double
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> Select * from problem3;
+--------+--------------+------------------+-------+
| gender | count_claims | count_treatments | ratio |
+--------+--------------+------------------+-------+
| female |         2676 |             2676 |     1 |
| male   |         4287 |             4287 |     1 |
+--------+--------------+------------------+-------+
2 rows in set (0.00 sec)
```

—--

QUERY8:

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.

Create External Table

create external table problem4

(pharmacyName String, count_medicines int, total_price double, total_discounted_price double)

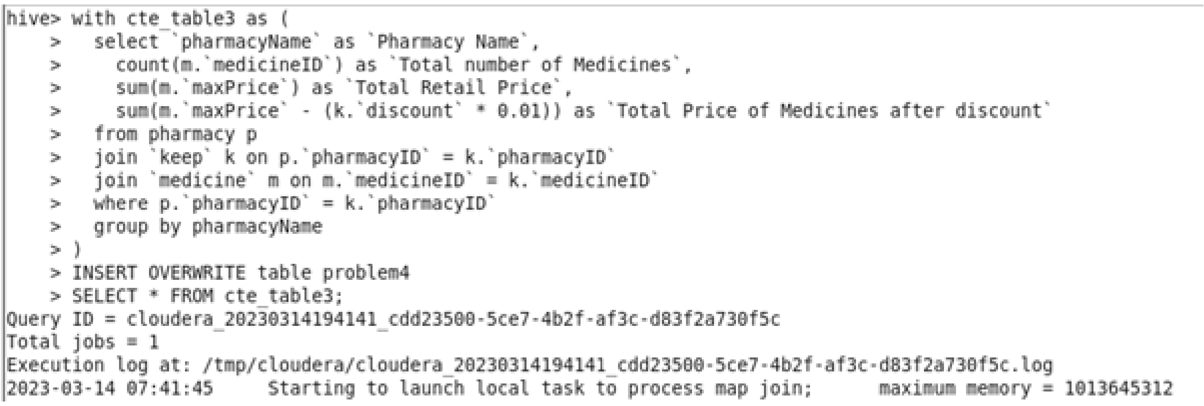
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

LINES TERMINATED BY '\n'

LOCATION '/user/hive/warehouse/project/problem4';

Insert Data Into External Table In Hive

```
with cte_table3 as (
  select `pharmacyName` as `Pharmacy Name`,
    count(m.`medicineID`) as `Total number of Medicines`,
    sum(m.`maxPrice`) as `Total Retail Price`,
    sum(m.`maxPrice` - (k.`discount` * 0.01)) as `Total Price of Medicines after discount`
  from pharmacy p
  join `keep` k on p.`pharmacyID` = k.`pharmacyID`
  join `medicine` m on m.`medicineID` = k.`medicineID`
  where p.`pharmacyID` = k.`pharmacyID`
  group by pharmacyName
)
INSERT OVERWRITE table problem4
SELECT * FROM cte_table3;
```



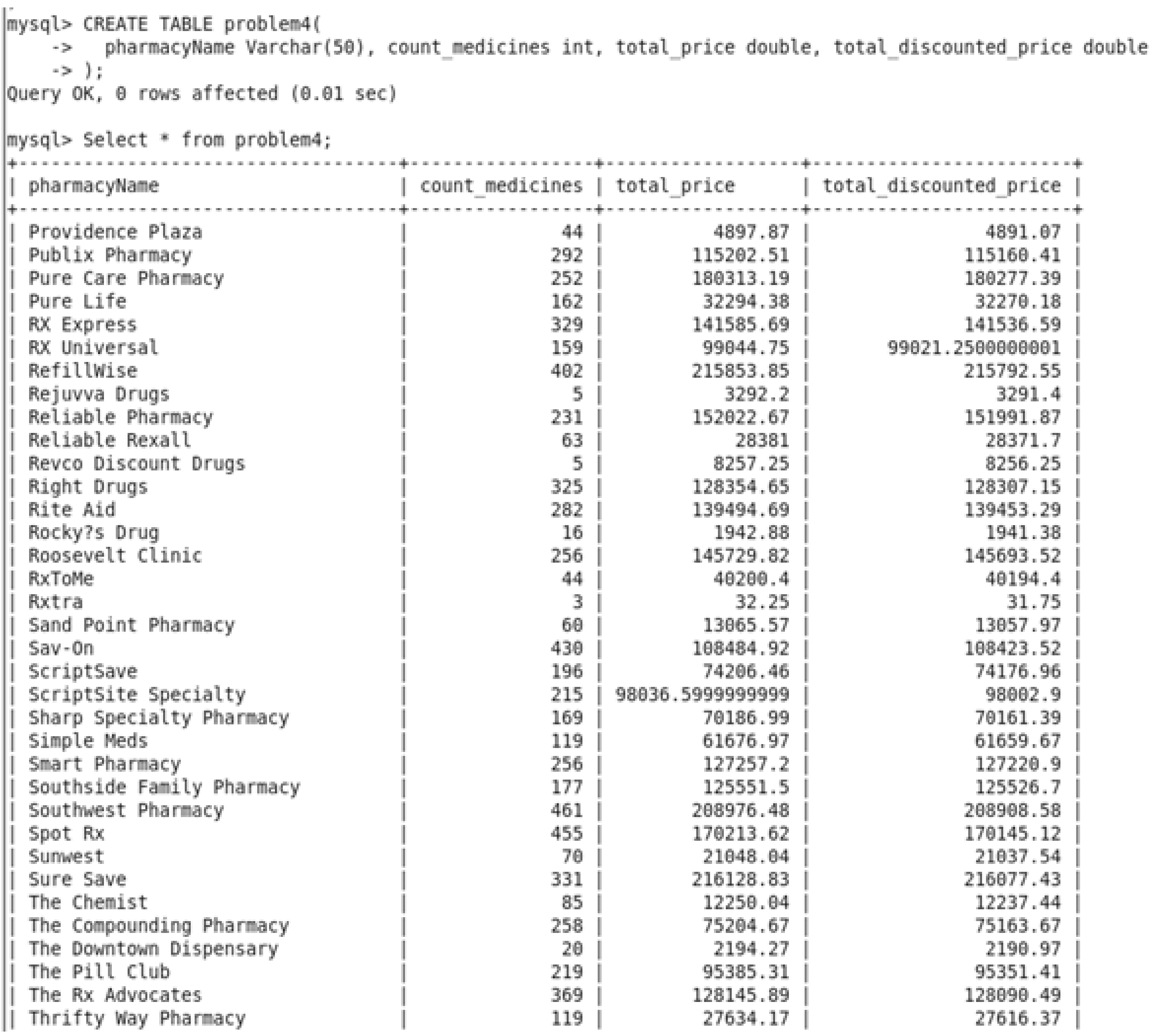
```
hive> with cte_table3 as (
    >   select `pharmacyName` as `Pharmacy Name`,
    >     count(m.`medicineID`) as `Total number of Medicines`,
    >     sum(m.`maxPrice`) as `Total Retail Price`,
    >     sum(m.`maxPrice` - (k.`discount` * 0.01)) as `Total Price of Medicines after discount`
    >   from pharmacy p
    >   join `keep` k on p.`pharmacyID` = k.`pharmacyID`
    >   join `medicine` m on m.`medicineID` = k.`medicineID`
    >   where p.`pharmacyID` = k.`pharmacyID`
    >   group by pharmacyName
    > )
    > INSERT OVERWRITE table problem4
    > SELECT * FROM cte_table3;
Query ID = cloudera_20230314194141_cdd23500-5ce7-4b2f-af3c-d83f2a730f5c
Total jobs = 1
Execution log at: /tmp/cloudera/cloudera_20230314194141_cdd23500-5ce7-4b2f-af3c-d83f2a730f5c.log
2023-03-14 07:41:45     Starting to launch local task to process map join;      maximum memory = 1013645312
```

<u>Create Output Table in Client DB</u>

```
CREATE TABLE problem4(
pharmacyName Varchar(50),
count_medicines int,
total_price double,
total_discounted_price double
);
```

<u>Move Data to Client DB using Sqoop Export</u>

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table problem4 \
--export-dir /user/hive/warehouse/project/problem4/000000_0;
```



```
mysql> CREATE TABLE problem4(
    ->   pharmacyName Varchar(50), count_medicines int, total_price double, total_discounted_price double
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql> Select * from problem4;
+-----------------------------------+-----------------+------------------+------------------------+
| pharmacyName                      | count_medicines | total_price      | total_discounted_price |
+-----------------------------------+-----------------+------------------+------------------------+
| Providence Plaza                  |              44 |          4897.87 |                4891.07 |
| Publix Pharmacy                   |             292 |        115202.51 |              115160.41 |
| Pure Care Pharmacy                |             252 |        180313.19 |              180277.39 |
| Pure Life                         |             162 |         32294.38 |               32270.18 |
| RX Express                        |             329 |        141585.69 |              141536.59 |
| RX Universal                      |             159 |         99044.75 |       99021.2500000001 |
| RefillWise                        |             402 |        215853.85 |              215792.55 |
| Rejuvva Drugs                     |               5 |           3292.2 |                 3291.4 |
| Reliable Pharmacy                 |             231 |        152022.67 |              151991.87 |
| Reliable Rexall                   |              63 |            28381 |                28371.7 |
| Revco Discount Drugs              |               5 |          8257.25 |                8256.25 |
| Right Drugs                       |             325 |        128354.65 |              128307.15 |
| Rite Aid                          |             282 |        139494.69 |              139453.29 |
| Rocky?s Drug                      |              16 |          1942.88 |                1941.38 |
| Roosevelt Clinic                  |             256 |        145729.82 |              145693.52 |
| RxToMe                            |              44 |          40200.4 |                40194.4 |
| Rxtra                             |               3 |            32.25 |                  31.75 |
| Sand Point Pharmacy               |              60 |         13065.57 |               13057.97 |
| Sav-On                            |             430 |        108484.92 |              108423.52 |
| ScriptSave                        |             196 |         74206.46 |               74176.96 |
| ScriptSite Specialty              |             215 | 98036.5999999999 |                98002.9 |
| Sharp Specialty Pharmacy          |             169 |         70186.99 |               70161.39 |
| Simple Meds                       |             119 |         61676.97 |               61659.67 |
| Smart Pharmacy                    |             256 |         127257.2 |               127220.9 |
| Southside Family Pharmacy         |             177 |         125551.5 |               125526.7 |
| Southwest Pharmacy                |             461 |        208976.48 |              208908.58 |
| Spot Rx                           |             455 |        170213.62 |              170145.12 |
| Sunwest                           |              70 |         21048.04 |               21037.54 |
| Sure Save                         |             331 |        216128.83 |              216077.43 |
| The Chemist                       |              85 |         12250.04 |               12237.44 |
| The Compounding Pharmacy          |             258 |         75204.67 |               75163.67 |
| The Downtown Dispensary           |              20 |          2194.27 |                2190.97 |
| The Pill Club                     |             219 |         95385.31 |               95351.41 |
| The Rx Advocates                  |             369 |        128145.89 |              128090.49 |
| Thrifty Way Pharmacy              |             119 |         27634.17 |               27616.37 |
```