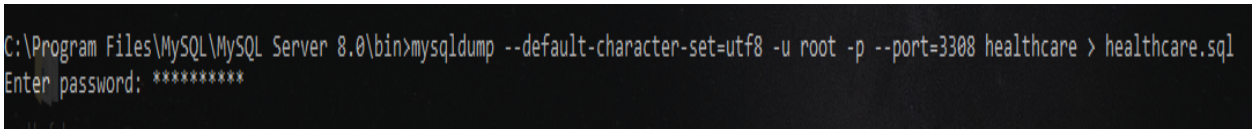
Create SQL dump for healthcare database



Create a healthcare database in Cloudera MySQL

```
mysql> create database healthcare
    -> ;
Query OK, 1 row affected (0.00 sec)
```

Dump SQL file into healthcare DB in Cloudera

```
[cloudera@quickstart ~]$ mysql -u root -p healthcare < /home/cloudera/Desktop/he
althcare.sql
Enter password:
```

```
mysql> show tables;
+----------------------+
| Tables_in_healthcare |
+----------------------+
| address              |
| claim                |
| contain              |
| disease              |
| insurancecompany     |
| insuranceplan        |
| keep                 |
| medicine             |
| patient              |
| person               |
| pharmacy             |
| prescription         |
| treatment            |
+----------------------+
13 rows in set (0.00 sec)
```

Sqoop import data into hive

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

```
.ogging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo
ı-1.1.0-cdh5.8.0.jar!/hive-log4j.properties
)K
ime taken: 0.082 seconds
.oading data to table default.pharmacy
:hgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehous
:/pharmacy/part-m-00000': User does not belong to supergroup
able default.pharmacy stats: [numFiles=1, totalSize=8105]
)K
ime taken: 0.239 seconds
lote: /tmp/sqoop-cloudera/compile/2dd58ee96afd81ac2b90d3c2d7453d00/prescription.
ava uses or overrides a deprecated API.
lote: Recompile with -Xlint:deprecation for details.

.ogging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo
ı-1.1.0-cdh5.8.0.jar!/hive-log4j.properties
)K
ime taken: 0.075 seconds
.oading data to table default.prescription
:hgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehous
:/prescription/part-m-00000': User does not belong to supergroup
able default.prescription stats: [numFiles=1, totalSize=322272]
)K
ime taken: 0.459 seconds
lote: /tmp/sqoop-cloudera/compile/2dd58ee96afd81ac2b90d3c2d7453d00/treatment.jav
ı uses or overrides a deprecated API.
lote: Recompile with -Xlint:deprecation for details.

.ogging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo
ı-1.1.0-cdh5.8.0.jar!/hive-log4j.properties
)K
ime taken: 0.077 seconds
.oading data to table default.treatment
:hgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehous
:/treatment/part-m-00000': User does not belong to supergroup
able default.treatment stats: [numFiles=1, totalSize=409446]
)K
ime taken: 0.355 seconds
cloudera@quickstart ~]$
```

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.

```
SELECT c.patientID, c.age, t.date AS treatment_date,
  CASE
    WHEN c.age BETWEEN 0 AND 14 THEN 'Children'
    WHEN c.age BETWEEN 15 AND 24 THEN 'Youth'
    WHEN c.age BETWEEN 25 AND 64 THEN 'Adults'
    ELSE 'Seniors'
  END AS age_category
FROM (
  SELECT patientID, ssn, dob, FLOOR(DATEDIFF(FROM_UNIXTIME(UNIX_TIMESTAMP()),
dob)/365.25) AS Age
  FROM patient
) as c
JOIN treatment t ON c.patientID = t.patientID
WHERE YEAR(t.date) = 2022;
```

Create external table

```
create external table prb1_1 (patientID int,age string,dates date,age_category string) row format
delimited fields terminated by ',' ;

INSERT OVERWRITE TABLE prb1_1 SELECT c.patientID, c.age, t.date AS treatment_date,
  CASE
    WHEN c.age BETWEEN 0 AND 14 THEN 'Children'
    WHEN c.age BETWEEN 15 AND 24 THEN 'Youth'
    WHEN c.age BETWEEN 25 AND 64 THEN 'Adults'
    ELSE 'Seniors'
  END AS age_category
FROM (
  SELECT patientID, ssn, dob, FLOOR(DATEDIFF(FROM_UNIXTIME(UNIX_TIMESTAMP()),
dob)/365.25) AS Age
  FROM patient
) as c
JOIN treatment t ON c.patientID = t.patientID
WHERE YEAR(t.date) = 2022;
```

```
hive> select * from prb1_1 limit 6;
OK
703275  56      2022-12-18      Adults
643936  57      2022-10-04      Adults
580404  7       2022-03-04      Children
958814  57      2022-02-03      Adults
605148  68      2022-09-19      Seniors
126581  61      2022-01-31      Adults
Time taken: 0.074 seconds, Fetched: 6 row(s)
hive>
```

Export external table to client DB

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_1 --export-dir /user/hive/warehouse/prb1_1 --input-fields-terminated-by ',';

```
mysql> select * from prb1_1 limit 5;
+-----------+------+------------+--------------+
| patientID | age  | dates      | age_category |
+-----------+------+------------+--------------+
|    703275 | 56   | 2022-12-18 | Adults       |
|    643936 | 57   | 2022-10-04 | Adults       |
|    580404 | 7    | 2022-03-04 | Children     |
|    958814 | 57   | 2022-02-03 | Adults       |
|    605148 | 68   | 2022-09-19 | Seniors      |
+-----------+------+------------+--------------+
5 rows in set (0.00 sec)
```

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.

```
SELECT diseaseid,
      SUM(CASE WHEN Gender = 'male' THEN Count END) AS male_count,
      SUM(CASE WHEN Gender = 'female' THEN Count END) AS female_count,
      (SUM(CASE WHEN Gender = 'male' THEN Count END) / SUM(CASE WHEN Gender =
'female' THEN Count END)) AS ratio
FROM (
  SELECT d.diseaseID AS diseaseid,
        pe.gender AS Gender,
        COUNT(*) AS Count
  FROM disease d
  JOIN treatment t ON d.diseaseID = t.diseaseID
  JOIN patient p ON p.patientID = t.patientID
  JOIN person pe ON pe.personID = p.patientID
  GROUP BY d.diseaseID, pe.gender
) a
GROUP BY diseaseid
ORDER BY diseaseid ASC;
```

Create external table

```
CREATE EXTERNAL TABLE prb1_2 (deseaseId INT,male_count INT,female_count INT, ratio
FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_2 SELECT diseaseid,
      SUM(CASE WHEN Gender = 'male' THEN Count END) AS male_count,
      SUM(CASE WHEN Gender = 'female' THEN Count END) AS female_count,
      (SUM(CASE WHEN Gender = 'male' THEN Count END) / SUM(CASE WHEN Gender =
'female' THEN Count END)) AS ratio
FROM (
  SELECT d.diseaseID AS diseaseid,
        pe.gender AS Gender,
        COUNT(*) AS Count
  FROM disease d
  JOIN treatment t ON d.diseaseID = t.diseaseID
  JOIN patient p ON p.patientID = t.patientID
  JOIN person pe ON pe.personID = p.patientID
  GROUP BY d.diseaseID, pe.gender
) a
GROUP BY diseaseid
ORDER BY diseaseid ASC;
```

Export external table to client DB

```
create table prb1_2 (deseaseId int,male_count int,female_count int,ratio float(3,2));

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password
cloudera --table prb1_2 --export-dir /user/hive/warehouse/prb1_2 --input-fields-terminated-by ',';
```

```
mysql> select * from prb1_2 limit 5;
+-----------+------------+--------------+-------+
| deseaseId | male_count | female_count | ratio |
+-----------+------------+--------------+-------+
|        32 |        145 |           94 |  1.54 |
|        33 |        157 |           93 |  1.69 |
|        34 |        156 |          113 |  1.38 |
|        35 |        170 |           96 |  1.77 |
|        36 |        190 |          117 |  1.62 |
+-----------+------------+--------------+-------+
5 rows in set (0.00 sec)
```

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.

```
SELECT p.gender,
  COUNT(t.treatmentId) AS `Total Treatment`,
  COUNT(c.claimId) AS `Total Claims`,
  COUNT(t.treatmentId) - COUNT(c.claimId) AS `difference`,
  COUNT(c.claimId) / COUNT(t.treatmentId) AS `ratio`
FROM treatment t
LEFT JOIN claim c ON t.claimID=c.claimID
INNER JOIN person p ON p.personId=t.patientId
GROUP BY p.gender;
```

Create external table

```
CREATE EXTERNAL TABLE prb1_3 (gender STRING, Total_Treatment INT, Total_Claims INT,
difference INT, ratio DECIMAL(5,4)) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;
```

```
INSERT OVERWRITE TABLE prb1_3 SELECT p.gender,
  COUNT(t.treatmentId) AS `Total Treatment`,
  COUNT(c.claimId) AS `Total Claims`,
  COUNT(t.treatmentId) - COUNT(c.claimId) AS `difference`,
  COUNT(c.claimId) / COUNT(t.treatmentId) AS `ratio`
FROM treatment t
LEFT JOIN claim c ON t.claimID=c.claimID
INNER JOIN person p ON p.personId=t.patientId
GROUP BY p.gender;
```

```
hive> select * from prb1_3;
OK
female  4206    2676    1530    0.6362
male    6679    4287    2392    0.6419
Time taken: 0.047 seconds, Fetched: 2 row(s)
```

Export external table to client DB

```
create table prb1_3 (gender varchar(30), Total_Treatment INT, Total_Claims INT, difference INT, ratio DECIMAL(5,4));

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_3  --export-dir /user/hive/warehouse/prb1_3 --input-fields-terminated-by ',';
```

```
mysql> select * from prb1_3;
+--------+-----------------+--------------+------------+--------+
| gender | Total_Treatment | Total_Claims | difference | ratio  |
+--------+-----------------+--------------+------------+--------+
| female |            4206 |         2676 |       1530 | 0.6362 |
| male   |            6679 |         4287 |       2392 | 0.6419 |
+--------+-----------------+--------------+------------+--------+
2 rows in set (0.00 sec)
```

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

```
SELECT p.pharmacyID, p.pharmacyName, COUNT(m.medicineid) AS units,
  ROUND(SUM(m.maxPrice),3) max_retail,
  ROUND(SUM(m.maxPrice - (k.discount * 0.01)),3) AS after_discount
FROM pharmacy p
JOIN keep k
  ON p.pharmacyID = k.pharmacyID
JOIN medicine m
  ON k.medicineID = m.medicineID
GROUP BY p.pharmacyID, p.pharmacyName;
```

Create external table

```
CREATE EXTERNAL TABLE prb1_4 (pharmacyID INT, pharmacyName STRING, units INT, max_retail FLOAT, after_discount FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_4 SELECT p.pharmacyID, p.pharmacyName, COUNT(m.medicineid) AS units,
  ROUND(SUM(m.maxPrice),3) max_retail,
  ROUND(SUM(m.maxPrice - (k.discount * 0.01)),3) AS after_discount
FROM pharmacy p
JOIN keep k
  ON p.pharmacyID = k.pharmacyID
```

JOIN medicine m
  ON k.medicineID = m.medicineID
GROUP BY p.pharmacyID, p.pharmacyName;

```
hive> select * from prb1_4 limit 5;
OK
1008    MobiMeds        47      16448.69        16442.29
1145    Spot Rx 455     170213.62       170145.12
1149    Modern Health   104     49718.78        49703.98
1194    Foundation Care 326     145124.4        145078.5
1204    Family Drug Mart        281     236986.31       236945.4
Time taken: 0.078 seconds, Fetched: 5 row(s)
```

Export external table to client DB

create table prb1_4 (pharmacyID INT, pharmacyName VARCHAR(30), units INT, max_retail FLOAT, after_discount FLOAT) ;

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_4 --export-dir /user/hive/warehouse/prb1_4 --input-fields-terminated-by ',';

```
mysql> select * from prb1_4 limit 10;
+------------+------------------------------+-------+------------+----------------+
| pharmacyID | pharmacyName                 | units | max_retail | after_discount |
+------------+------------------------------+-------+------------+----------------+
|       8109 | Premier Long Term Care Pharm |   404 |     155790 |         155727 |
|       8142 | RX Universal                 |   159 |    99044.8 |        99021.2 |
|       8173 | Innovia Drug Stores          |   393 |     367147 |         367085 |
|       8184 | Apotheco                     |   454 |     207948 |         207877 |
|       8265 | Pharmacy Alliance            |   223 |     115386 |         115351 |
|       8315 | RefillWise                   |   402 |     215854 |         215793 |
|       8320 | Adams Drugs                  |   375 |     235377 |         235319 |
|       8349 | Grand Medicine               |    14 |    3833.71 |        3831.31 |
|       8404 | Atlas Drugs                  |    41 |    11520.2 |        11514.8 |
|       8442 | Wellman's Pharmacy           |   252 |     146250 |         146213 |
+------------+------------------------------+-------+------------+----------------+
10 rows in set (0.00 sec)
```

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 the average number of medicines prescribed in their prescriptions.

```
SELECT p.pharmacyName,
    MAX(k.quantity) AS max_meds,
    MIN(k.quantity) AS min_meds,
    ROUND(AVG(k.quantity), 2) AS avg_meds
FROM pharmacy p
JOIN keep k ON p.pharmacyID = k.pharmacyID
GROUP BY p.pharmacyName;
```

Create external table

```
CREATE EXTERNAL TABLE prb1_5 (pharmacyName STRING, max_meds INT, min_meds INT, avg_meds FLOAT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_5 SELECT p.pharmacyName,
    MAX(k.quantity) AS max_meds,
    MIN(k.quantity) AS min_meds,
    ROUND(AVG(k.quantity), 2) AS avg_meds
FROM pharmacy p
JOIN keep k ON p.pharmacyID = k.pharmacyID
GROUP BY p.pharmacyName;
```

```
Time taken: 79.800 seconds
hive> select * from prb1_5 limit 5;
OK
Absolute Care   9996    1       5298.34
Acculife Drug Stores    9984    14      4972.57
Adams Drugs     9993    3       5143.2
Alliance Community      9997    0       5093.96
Ally Scripts    9878    169     5053.17
Time taken: 0.153 seconds, Fetched: 5 row(s)
hive>
```

Export external table to client DB

```
create table prb1_5  (pharmacyName VARCHAR(30), max_meds INT, min_meds INT, avg_meds FLOAT) ;

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_5  --export-dir /user/hive/warehouse/prb1_5 --input-fields-terminated-by ',';
```

```
mysql> select * from prb1_5 limit 10;
+---------------------------+----------+----------+----------+
| pharmacyName              | max_meds | min_meds | avg_meds |
+---------------------------+----------+----------+----------+
| Express Scripts           |     9940 |        1 |  4818.62 |
| Family Drug Mart          |     9987 |       53 |  4794.96 |
| Family Fare               |     9977 |        5 |  4977.29 |
| First Care Rx             |     9998 |       36 |  5114.15 |
| First Hill Pharmacy       |     9998 |      171 |  4919.98 |
| First Pharmacy            |     9949 |      183 |  5383.55 |
| Fisher Pharmacy & Gifts   |     9989 |        3 |  5081.22 |
| Foundation Care           |     9997 |       15 |  5064.79 |
| Friends & Family Pharmacy |     9898 |       94 |  5141.08 |
| Frost Medical Pharmacy    |     9999 |       24 |  5185.65 |
+---------------------------+----------+----------+----------+
10 rows in set (0.01 sec)
```

6) A company needs to set up 3 new pharmacies, they have come up with an idea that the pharmacy can be set up in cities where the pharmacy-to-prescription ratio is the lowest and the number of prescriptions should exceed 100. Assist the company to identify those cities where the pharmacy can be set up.

```
SELECT a.city,
    ROUND(COUNT(DISTINCT pha.pharmacyID) / COUNT(DISTINCT pre.prescriptionID),4)
AS pharmacy_prescription_ratio
FROM pharmacy pha
JOIN address a ON a.addressID = pha.addressID
JOIN prescription pre ON pha.pharmacyID = pre.pharmacyID
GROUP BY a.city
HAVING COUNT(pre.prescriptionID) > 100
ORDER BY pharmacy_prescription_ratio;
```

Create external table

```
CREATE EXTERNAL TABLE prb1_6 (city STRING, pharmacy_prescription_ratio DECIMAL
(6,4)) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_6 SELECT a.city,
    ROUND(COUNT(DISTINCT pha.pharmacyID) / COUNT(DISTINCT pre.prescriptionID),4)
AS pharmacy_prescription_ratio
FROM pharmacy pha
JOIN address a ON a.addressID = pha.addressID
JOIN prescription pre ON pha.pharmacyID = pre.pharmacyID
GROUP BY a.city
```

HAVING COUNT(pre.prescriptionID) > 100
ORDER BY pharmacy_prescription_ratio;

```
hive> select * from prb1_6 limit 10;
OK
Worcester       0.0137
Panama City Beach       0.014
Glen Burnie     0.0143
Goodlettsville  0.0147
Anchorage       0.0152
Crownsville     0.0153
Pooler  0.0153
Nashville       0.0153
Montgomery      0.0154
Washington      0.0155
Time taken: 0.109 seconds, Fetched: 10
```

Export external table to client DB

create table prb1_6 (city VARCHAR(30), pharmacy_prescription_ratio DECIMAL (6,4)) ;

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_6 --export-dir /user/hive/warehouse/prb1_6 --input-fields-terminated-by ',';

```
mysql> select * from prb1_6 limit 10;
+-------------------+-----------------------------+
| city              | pharmacy_prescription_ratio |
+-------------------+-----------------------------+
| Union City        |                      0.0162 |
| Worcester         |                      0.0137 |
| Panama City Beach |                      0.0140 |
| Glen Burnie       |                      0.0143 |
| Goodlettsville    |                      0.0147 |
| Anchorage         |                      0.0152 |
| Crownsville       |                      0.0153 |
| Panama City       |                      0.0165 |
| Oklahoma City     |                      0.0166 |
| Savannah          |                      0.0166 |
+-------------------+-----------------------------+
10 rows in set (0.00 sec)
```

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.

Creating partitions on address table based on states

```
CREATE TABLE address_part (
  addressid INT,
  address1 STRING,
  city STRING,
  zip INT)
PARTITIONED BY (state STRING) CLUSTERED BY (city) INTO 10 BUCKETS;

INSERT INTO TABLE address_part PARTITION (state)
SELECT addressid,address1,city,zip,state FROM address;
```

Checking performance difference (used EXPLAIN) with partitioned and non partitioned table

Partitioned:

```
                    table:
                        input
                        outpu
                        serde

  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink

Time taken: 1.599 seconds,
```

Non Partitioned:

```
  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink

Time taken: 4.29 seconds,
```

Create external table

CREATE EXTERNAL TABLE prb1_ 7 (city STRING, diseaseName STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_7 SELECT city, diseaseName
FROM (
  SELECT city, diseaseName, counts, DENSE_RANK() OVER (PARTITION BY city ORDER BY counts DESC) AS ranks
  FROM (
  SELECT a.city, d.diseaseName, COUNT(t.patientID) AS counts
  FROM treatment t
  JOIN person p ON t.patientID = p.personID
  JOIN disease d ON t.diseaseID = d.diseaseID
  JOIN address_part a ON p.addressID = a.addressID
  WHERE a.state = 'AL'
  GROUP BY a.city, d.diseaseName
) a
) b
WHERE ranks = 1;

```
hive> select * from prb1_7;
OK
Indian Springs Village  Diabetes mellitus type 2
Indian Springs Village  Alzheimer's disease
Indian Springs Village  Multiple sclerosis
Indian Springs Village  Parkinson's disease
Indian Springs Village  Schizophrenia
Indian Springs Village  Bipolar disorder
Montevallo      Schizophrenia
Montgomery      Guillain–Barré syndrome
Montgomery      Cancer
Time taken: 0.14 seconds, Fetched: 9 row(s)
```

Export external table to client DB

create table prb1_7 (city VARCHAR(30), diseaseName VARCHAR(50)) ;

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_7 --export-dir /user/hive/warehouse/prb1_7 --input-fields-terminated-by ',';

```
mysql> select * from prb1_7;
+------------------------+---------------------------+
| city                   | diseaseName               |
+------------------------+---------------------------+
| Indian Springs Village | Diabetes mellitus type 2  |
| Indian Springs Village | Multiple sclerosis        |
| Indian Springs Village | Parkinson's disease       |
| Indian Springs Village | Alzheimer's disease       |
| Montevallo             | Schizophrenia             |
| Montgomery             | Guillain–Barré syndrome   |
| Montgomery             | Cancer                    |
| Indian Springs Village | Schizophrenia             |
| Indian Springs Village | Bipolar disorder          |
+------------------------+---------------------------+
9 rows in set (0.00 sec)
```

8) The Healthcare department wants to know which disease is most likely to infect multiple people in the same household. For each disease find the number of households that has more than one patient with the same disease.
Note: 2 people are considered to be in the same household if they have the same address.

Creating BUCKETS on treatment table based on diseaseid

```
CREATE TABLE treatment_part (
  treatmentid INT,
  date DATE,
  patientid INT,
  diseaseid INT,
  claimid BIGINT
  )
  CLUSTERED BY (diseaseid) INTO 5 BUCKETS;

INSERT INTO TABLE treatment_part SELECT treatmentid,date,patientid,diseaseid,claimid
FROM treatment;
```

Checking performance difference (used EXPLAIN) with bucketed and non-bucketed table

Non bucketed:

```
  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink

Time taken: 0.311 seconds,
```

Bucketed:

```
  Stage: Stage-0
    Fetch Operator
      limit: -1
      Processor Tree:
        ListSink

Time taken: 0.248 seconds,
```

Create external table

```
CREATE EXTERNAL TABLE prb1_8 (diseaseName STRING, no_of_household INT) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_8 SELECT diseaseName, COUNT(address1) AS
no_of_household
FROM (select d.diseaseName,a.address1,COUNT(t.patientID)
from disease d join treatment_part t on d.diseaseID=t.diseaseID
join person p on t.patientID=p.personID
join address a on p.addressID=a.addressID
group by d.diseasename,a.address1
having count(t.patientID)>1
order by d.diseasename desc,a.address1) a
group by diseaseName;
```

```
hive> select * from prb1_8 limit 10;
OK
Alzheimer's disease     45
Amyotrophic lateral sclerosis   46
Anorexia nervosa        52
Anxiety disorder        39
Asthma  40
Atherosclerosis 52
Attention deficit hyperactivity disorder        44
Autism  37
Autoimmune diseases     46
Bipolar disorder        46
Time taken: 0.044 seconds, Fetched: 10 row(s)
```

Export external table to client DB

create table prb1_8 (diseaseName VARCHAR(50), no_of_household INT);

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_8 --export-dir /user/hive/warehouse/prb1_8 --input-fields-terminated-by ',';

```
mysql> select * from prb1_8 limit 10;
+---------------------------+-----------------+
| diseaseName               | no_of_household |
+---------------------------+-----------------+
| Panic disorder            |              44 |
| Parkinson's disease       |              38 |
| Psoriasis                 |              37 |
| Rheumatoid arthritis      |              41 |
| Sarcoidosis               |              56 |
| Schizophrenia             |              58 |
| Stroke                    |              54 |
| Thromboangiitis obliterans |              47 |
| Tourette syndrome         |              55 |
| Vasculitis                |              53 |
+---------------------------+-----------------+
10 rows in set (0.00 sec)
```

9) 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.

Create external table

CREATE EXTERNAL TABLE prb1_9 (state STRING, treatment_claim_ratio DECIMAL(3,2)) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_9 select a.state, round(count(treatmentID)/sum(if(claimID is null,0,1)),2) as treatment_claim_ratio
from treatment_part t join person p on t.patientID=p.personID
join address_part a on a.addressID=p.addressID
where t.date between '2021-04-01' and '2022-03-31'
group by a.state;

```
hive> select * from prb1_9;
OK
AK      1.46
AL      1.64
AR      1.53
AZ      1.65
CA      1.47
CO      1.6
CT      1.45
DC      1.52
FL      1.68
GA      1.54
KY      1.47
MA      1.48
MD      1.52
OK      1.68
TN      1.69
VT      1.47
Time taken: 0.131 seconds, Fe
```

Export external table to client DB

create table prb1_9 (state VARCHAR(2), treatment_claim_ratio DECIMAL(3,2));

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_9 --export-dir /user/hive/warehouse/prb1_9 --input-fields-terminated-by ',';

```
mysql> select * from prb1_9;
+-------+----------------------+
| state | treatment_claim_ratio |
+-------+----------------------+
| AK    |                 1.46 |
| AL    |                 1.64 |
| AR    |                 1.53 |
| AZ    |                 1.65 |
| MD    |                 1.52 |
| OK    |                 1.68 |
| TN    |                 1.69 |
| VT    |                 1.47 |
| CA    |                 1.47 |
| CO    |                 1.60 |
| CT    |                 1.45 |
| DC    |                 1.52 |
| FL    |                 1.68 |
| GA    |                 1.54 |
| KY    |                 1.47 |
| MA    |                 1.48 |
+-------+----------------------+
16 rows in set (0.01 sec)
```

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

Create external table

```
CREATE EXTERNAL TABLE prb1_10 (pharmacyID INT, Total_Quantity BIGINT) ROW
FORMAT DELIMITED FIELDS TERMINATED BY ',' ;

INSERT OVERWRITE TABLE prb1_10 SELECT p.pharmacyID,sum(k.quantity) as
Total_Quantity FROM pharmacy p
JOIN keep k ON p.pharmacyID = k.pharmacyID
JOIN address_part a on a.`addressID`=p.`addressID`
JOIN medicine m on m.medicineID=k.medicineID
JOIN prescription pr on pr.pharmacyID = p.pharmacyID
join treatment_part t on pr.treatmentID = t.treatmentID
where a.state='AZ' and year(t.date)=2021 and taxCriteria ='I'
GROUP BY p.pharmacyID;
```

```
hive> select * from prb1_10;
OK
1478    17573685
1624    24677247
1628    8163570
2218    11489748
2301    10172277
3104    10468116
3536    25515630
3799    2750373
4938    20963730
5450    2792202
5480    12929860
8442    11520917
8829    6890212
8897    16609140
8933    10970658
9659    15568260
9681    9995295
Time taken: 0.044 seconds, Fe
hive>
```

Export external table to client DB

create table prb1_10 (pharmacyID INT, Total_Quantity BIGINT);

sqoop export --connect jdbc:mysql://localhost/healthcare_sol --username root --password cloudera --table prb1_10 --export-dir /user/hive/warehouse/prb1_10 --input-fields-terminated-by ',';

```
mysql> select * from prb1_10;
+------------+----------------+
| pharmacyID | Total_Quantity |
+------------+----------------+
|       8897 |       16609140 |
|       8933 |       10970658 |
|       9659 |       15568260 |
|       9681 |        9995295 |
|       1478 |       17573685 |
|       1624 |       24677247 |
|       1628 |        8163570 |
|       2218 |       11489748 |
|       2301 |       10172277 |
|       3104 |       10468116 |
|       3536 |       25515630 |
|       3799 |        2750373 |
|       4938 |       20963730 |
|       5450 |        2792202 |
|       5480 |       12929860 |
|       8442 |       11520917 |
|       8829 |        6890212 |
+------------+----------------+
17 rows in set (0.00 sec)
```