HIVE PROJECT (HEALTHCARE)

COMMON IMPORT OF ALL TABLES

DATABASE IMPORT USING SQOOP

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

PROBLEM STATEMENT 1

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.

HIVE QUERY

```
SELECT gender, SUM(total_treat), SUM(total_claim), CONCAT(SUM(total_treat), ":", SUM(total_claim))
as ratio
FROM(
   SELECT gender, total_treat, total_claim
   FROM(
      SELECT gender, patientID, SUM(treat_count) as total_treat, SUM(claim_count) as total_claim
      FROM(
         SELECT pe.gender, pa.patientID, COUNT(t.treatmentID) as treat_count, COUNT(t.claimID) as
claim_count
         FROM person pe
         INNER JOIN patient pa ON pe.personID = pa.patientID
         INNER JOIN treatment t ON t.patientID = pa.patientID
         GROUP BY pe.gender, pa.patientID, t.claimID
      ) a
      GROUP BY gender, patientID
   ) b
) c
GROUP BY gender;
```

The above query joins 3 tables person, patient and treatment. The subquery uses count () to calculate no. of treatments and claims for each patient grouped by gender and patientID. Finally the result is the total treatments to claims made ratio based on the gender.

HIVE EXTERNAL TABLE

```
create external table ps1_q3(gender string,
sum_treatment int,
sum_claim int,
ratio string)
```

```
row format delimited
fields terminated by ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps1_q3
SELECT gender, SUM(total_treat), SUM(total_claim), CONCAT(SUM(total_treat), ":", SUM(total_claim))
as ratio
FROM(
   SELECT gender, total_treat, total_claim
   FROM(
      SELECT gender, patientID, SUM(treat_count) as total_treat, SUM(claim_count) as total_claim
      FROM(
         SELECT pe.gender, pa.patientID, COUNT(t.treatmentID) as treat_count, COUNT(t.claimID) as
claim_count
         FROM person pe
         INNER JOIN patient pa ON pe.personID = pa.patientID
         INNER JOIN treatment t ON t.patientID = pa.patientID
         GROUP BY pe.gender, pa.patientID, t.claimID
      ) a
      GROUP BY gender, patientID
   ) b
) c
GROUP BY gender;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps1_q3 \
--export-dir /user/hive/warehouse/ps1_q3/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 2

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.

HIVE QUERY

```
SELECT c.pharmacyID, SUM(c.quantity) AS total_quantity, SUM(c.total_mrp) AS total_cost,
SUM(c.total_discounted_price) AS total_discount_cost
FROM (
   SELECT k.pharmacyID, k.medicineID, k.quantity, m.maxPrice, k.discount,
        (m.maxPrice * k.quantity) AS total_mrp,
        ((m.maxPrice * k.quantity) - ((m.maxPrice * k.quantity) * k.discount / 100)) AS
total_discounted_price
   FROM keep k
   INNER JOIN medicine m ON k.medicineID = m.medicineID
) c
```

```
GROUP BY c.pharmacyID;
```

The above query joins the keep and medicine tables. For each row, the MRP and total discounted value is calculated and grouping is done based on pharmacyID and finally calculates the total quantity, total cost and total discounted cost for each group.

HIVE EXTERNAL TABLE

```
create external table ps1_q4(pharmacyID int,
total_quantity int,
total_cost float,
total_discount_cost float)
row format delimited
fields terminated by ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps1_q4
SELECT k.pharmacyID, k.medicineID, k.quantity, m.maxPrice, k.discount,
        (m.maxPrice * k.quantity) AS total_mrp,
        ((m.maxPrice * k.quantity) - ((m.maxPrice * k.quantity) * k.discount / 100)) AS
total_discounted_price
   FROM keep k
   INNER JOIN medicine m ON k.medicineID = m.medicineID
) c
GROUP BY c.pharmacyID;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps1_q4 \
--export-dir /user/hive/warehouse/ps1_q4/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 3

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

HIVE QUERY

```
SELECT p.pharmacyID, c.prescriptionID, MIN(c.quantity), MAX(c.quantity), ROUND(AVG(c.quantity))
FROM prescription p
JOIN contain c ON p.prescriptionID = c.prescriptionID
GROUP BY p.pharmacyID, c.prescriptionID;
```

The above query joins contain and prescription tables using prescriptionID column. For every row it calculates the minimum, maximum and average quantity of medicines in every prescription prescribed by the pharmacy. For this the grouping is done based on pharmacyID and precriptionID columns.

HIVE EXTERNAL TABLE

```
create external table ps1_q5(pharmacyID int,
prescriptionID int,
min_quantity float,
max_quantity float,
round_avg_quantity float)
row format delimited
fields terminated by ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE  ps1_q5
SELECT p.pharmacyID, c.prescriptionID, MIN(c.quantity), MAX(c.quantity), ROUND(AVG(c.quantity))
FROM prescription p
JOIN contain c ON p.prescriptionID = c.prescriptionID
GROUP BY p.pharmacyID, c.prescriptionID;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps1_q5 \
--export-dir /user/hive/warehouse/ps1_q5/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 4

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 QUERY

```
SELECT
   CASE
     WHEN datediff(current_date, dob)/365 BETWEEN 0 AND 14 THEN 'Children'
     WHEN datediff(current_date, dob)/365 BETWEEN 15 AND 24 THEN 'Youth'
     WHEN datediff(current_date, dob)/365 BETWEEN 25 AND 64 THEN 'Adult'
     ELSE 'Seniors'
   END AS category,
   count(treatmentID) AS tot_treat
FROM treatment t
 INNER JOIN patient p
 on p.patientID = t.patientID
GROUP BY
 CASE
```

```
    WHEN datediff(current_date, dob)/365 BETWEEN 0 AND 14 THEN 'Children'
    WHEN datediff(current_date, dob)/365 BETWEEN 15 AND 24 THEN 'Youth'
    WHEN datediff(current_date, dob)/365 BETWEEN 25 AND 64 THEN 'Adult'
    ELSE 'Seniors'
    END;
```

The above query joins treatment and patient tables and uses CASE to categorise the data based on age and gives the total count of treatments for every category.

HIVE EXTERNAL TABLE

```
create external table ps1_q1(category string,
total_treatments int)
row format delimited
fields terminated by ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps1_q1
SELECT
   CASE
     WHEN datediff(current_date, dob)/365 BETWEEN 0 AND 14 THEN 'Children'
     WHEN datediff(current_date, dob)/365 BETWEEN 15 AND 24 THEN 'Youth'
     WHEN datediff(current_date, dob)/365 BETWEEN 25 AND 64 THEN 'Adult'
     ELSE 'Seniors'
   END AS category,
   count(treatmentID) AS tot_treat
FROM treatment t
 INNER JOIN patient p
 on p.patientID = t.patientID
GROUP BY
 CASE
     WHEN datediff(current_date, dob)/365 BETWEEN 0 AND 14 THEN 'Children'
     WHEN datediff(current_date, dob)/365 BETWEEN 15 AND 24 THEN 'Youth'
     WHEN datediff(current_date, dob)/365 BETWEEN 25 AND 64 THEN 'Adult'
     ELSE 'Seniors'
     END;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps1_q1 \
--export-dir /user/hive/warehouse/ps1_q1/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 5

Jimmy from healthcare department wants to know the state wise patient count.
Help him in generating this report.

HIVE QUERY

```
SELECT a.state, COUNT(DISTINCT p.personID) AS patient_count
FROM address_part a
JOIN person p ON a.addressid = p.addressID
JOIN patient pt ON p.personID = pt.patientID
GROUP BY a.state;
```

The above query joins the address, patient and person tables. The result is a distinct count of patients in every state which achieved by inner joins and grouping the state column. The query has been executed on a partitioned address table. It has been been partitioned on the state column.

PARTITION TABLE

```
CREATE TABLE IF NOT EXISTS address_part (addressid int,address1 string,city string,zip int)
   COMMENT 'Address_partition'
   PARTITIONED BY (state string)
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ','
   LINES TERMINATED BY '\n';
```

HIVE EXTERNAL TABLE

```
create external table ps1_q2(state string,
count_patients int)
row format delimited
fields terminated by ','
lines terminated by '\n';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps1_q2
SELECT a.state, COUNT(DISTINCT p.personID) AS patient_count
FROM address_part a
JOIN person p ON a.addressid = p.addressID
JOIN patient pt ON p.personID = pt.patientID
GROUP BY a.state;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps1_q2 \
--export-dir /user/hive/warehouse/ps1_q2/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 6

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.

HIVE QUERY

```
SELECT state, COUNT(treatmentID) / COUNT(claimID) AS ratio
FROM address_part a
JOIN person p ON a.addressid = p.addressID
JOIN treatment t ON p.personID = t.patientID
WHERE t.date BETWEEN '2021-04-01' AND '2022-03-31'
GROUP BY state;
```

The above query joins 3 tables, address, person and patient and results in state wise ratio of treatment to claims between 31st March 2021 and 1st April 2021 and grouped by state column. The query has been applied on a partitioned address table, partitioned on the state column.

PARTITION TABLE

```
CREATE TABLE IF NOT EXISTS address_part (addressid int,address1 string,city string,zip int)
   COMMENT 'Address_partition'
   PARTITIONED BY (state string)
   ROW FORMAT DELIMITED
   FIELDS TERMINATED BY ','
   LINES TERMINATED BY '\n';
```

HIVE EXTERNAL TABLE

```
CREATE EXTERNAL TABLE IF NOT EXISTS ps2_q5 (state string,ratio int)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps2_q5
SELECT state, COUNT(treatmentID) / COUNT(claimID) AS ratio
FROM address_part a
JOIN person p ON a.addressid = p.addressID
JOIN treatment t ON p.personID = t.patientID
WHERE t.date BETWEEN '2021-04-01' AND '2022-03-31'
GROUP BY state;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps2_q5 \
--export-dir /user/hive/warehouse/ps2_q5/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 7

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.

HIVE QUERY

```
select p.pharmacyName,sum(c.quantity) as total_quant
from
    address a inner join pharmacy p on a.addressID = p.addressID
    inner join prescription pr on p.pharmacyID = pr.pharmacyID
    inner join contain c on c.prescriptionID = pr.prescriptionID
    inner join medicine m on m.medicineID = c.medicineID
    inner join treatment_part_clus t on pr.treatmentID=t.treatmentID
where m.taxCriteria='I'
and cast(t.date as date) between '2021-01-31' and '2021-12-31'
and a.state='AZ'
group by p.pharmacyName;
```

The above query joins 5 tables, address,pharmacy,prescription,medicine and treatment based on the columns common between them. The query results the total quantity of medicines prescribed by the pharmacies of Arizona which fall under tax criteria I in the year 2021. The data is achieved by grouping the results based on pharmacyName column.
The treatment table here is partitioned on diseaseID column and clustered into 2 buckets on patientID column

PARTITION CLUSTERED TABLE

```
CREATE TABLE treatment_part_clus(
   treatmentID BIGINT,
   patientID BIGINT,
   date STRING,
   claimID BIGINT
)
COMMENT 'Treatment to claim ratio'
PARTITIONED BY (diseaseID STRING) CLUSTERED BY(patientID) INTO 2 BUCKETS
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ',';
```

HIVE EXTERNAL TABLE

```
create external table ps3_q5(pharmacyName string, total_quant int)
row format delimited
fields terminated by ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps3_q5
select p.pharmacyName,sum(c.quantity) as total_quant
from
    address a inner join pharmacy p on a.addressID = p.addressID
    inner join prescription pr on p.pharmacyID = pr.pharmacyID
    inner join contain c on c.prescriptionID = pr.prescriptionID
    inner join medicine m on m.medicineID = c.medicineID
    inner join treatment_part_clus t on pr.treatmentID=t.treatmentID
where m.taxCriteria='I'
and cast(t.date as date) between '2021-01-31' and '2021-12-31'
```

```
and a.state='AZ'
group by p.pharmacyName;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps3_q5 \
--export-dir /user/hive/warehouse/ps3_q5/000000_0 \
--input-fields-terminated-by ','
```

PROBLEM STATEMENT 8

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.

HIVE QUERY

```
SELECT city, diseaseName, cnt_disease
FROM (
 SELECT *, dense_rank() OVER (PARTITION BY city ORDER BY cnt_disease DESC) rnk
 FROM (
  SELECT DISTINCT a.city, d.diseaseName, count(*) OVER (PARTITION BY d.diseaseName, a.city)
cnt_disease
  FROM treatment t
  JOIN disease d ON t.diseaseID = d.diseaseID
  JOIN patient p ON p.patientID = t.patientID
  JOIN person pr ON pr.personID = p.patientID
  JOIN address a ON a.addressID = pr.addressID
  WHERE a.state = 'AL'
 ) a
) b
WHERE rnk = 1;
```

The above query uses subqueries and joins. Joins are made on tables disease, patient, person, address in the innermost subquery which finds the city, diseaseName and uses count () as window function to get the count of disease in the state of Alabama. The outer query then uses dense rank() on the result of the inner query and selects only those records who have the highest rank.

HIVE EXTERNAL TABLE

```
create external table ps2_q21(city string, diseaseName string, count_dis int)
row format delimited
fields terminated by ',';
```

INSERT INTO EXTERNAL TABLE

```
INSERT OVERWRITE TABLE ps2_q21
SELECT city, diseaseName, cnt_disease
FROM (
```

```
 SELECT *, dense_rank() OVER (PARTITION BY city ORDER BY cnt_disease DESC) rnk
 FROM (
  SELECT DISTINCT a.city, d.diseaseName, count(*) OVER (PARTITION BY d.diseaseName, a.city)
cnt_disease
  FROM treatment t
  JOIN disease d ON t.diseaseID = d.diseaseID
  JOIN patient p ON p.patientID = t.patientID
  JOIN person pr ON pr.personID = p.patientID
  JOIN address a ON a.addressID = pr.addressID
  WHERE a.state = 'AL'
 ) a
) b
WHERE rnk = 1;
```

DATA EXPORT TO CLIENT DATABASE USING SQOOP

```
sqoop export \
--connect jdbc:mysql://localhost:3306/healthcare \
--username root \
--password cloudera \
--table ps2_q21 \
--export-dir /user/hive/warehouse/ps2_q21/000000_0 \
--input-fields-terminated-by ','
```
