1. Table creation and loading data into hive warehouse.

```
CREATE TABLE ecom_data (
    order_id STRING,
    customer_id STRING,
    quantity INT,
    price_MRP FLOAT,
    payment FLOAT,
    timestamp STRING,
    rating INT,
    product_category STRING,
    product_id STRING,
    payment_type STRING,
    order_status STRING,
    product_weight_g INT,
    product_length_cm INT,
    product_height_cm INT,
    product_width_cm INT,
    customer_city STRING,
    customer_state STRING,
    seller_id STRING,
    seller_city STRING,
    seller_state STRING,
    payment_installments INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
TBLPROPERTIES("skip.header.line.count"="1");
```

loading data into table;

```
load data inpath 'ecom_hive_project/ecom_data' into table ecom_data;
```

creating new empty table to remove timestamp issues and duplicate values

```
CREATE TABLE ecom(
   order_id STRING,
   customer_id STRING,
   quantity INT,
   price_MRP FLOAT,
   payment FLOAT,
   timestamp timestamp,
   rating INT,
   product_category STRING,
   product_id STRING,
   payment_type STRING,
   order_status STRING,
   product_weight_g INT,
   product_length_cm INT,
   product_height_cm INT,
   product_width_cm INT,
   customer_city STRING,
   customer_state STRING,
   seller_id STRING,
   seller_city STRING,
   seller_state STRING,
   payment_installments INT
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
TBLPROPERTIES("skip.header.line.count"="1");
```

Inserting value in newly created table

```
INSERT OVERWRITE TABLE ecom
SELECT DISTINCT order_id,
   customer_id,
   quantity,
   price_MRP,
   payment,
   CASE
      WHEN timestamp LIKE '%-%' THEN from_unixtime(unix_timestamp(timestamp, 'dd-MM-yyyy HH:mm'))
      WHEN timestamp LIKE '%/%' THEN from_unixtime(unix_timestamp(timestamp, 'dd/MM/yyyy hh:mm'))
      ELSE NULL
   END,
   rating,
   product_category,
   product_id,
   payment_type,
   order_status,
   product_weight_g,
   product_length_cm,
   product_height_cm,
   product_width_cm,
   customer_city,
   customer_state,
   seller_id,
   seller_city,
   seller_state,
   payment_installments
FROM (
   SELECT *,
      dense_rank() OVER (PARTITION BY order_id, product_id ORDER BY quantity DESC) AS `ranking`
   FROM ecom_data
```

) t

WHERE ranking = 1;

-> HIVE JOBS.....

1. Customer Segmentation

Categorizing customers based on their spendings

SELECT customer_id,SUM(payment) AS spending, CASE WHEN SUM(payment) < 25000 THEN 'Under Affluent' WHEN SUM(payment) BETWEEN 25000 AND 75000 THEN 'Semi-affluent' WHEN SUM(payment) > 75000 THEN 'Affluent' END AS Categories FROM ecom_data GROUP BY customer_id ORDER BY spending DESC;

-> creating external table

create external table customer_segmentation(Customer_id string, Spending float, Categories string) row format delimited fields terminated by ' ' location '/user/hive/warehouse/problem_1/result.txt';

->loading data into external table

insert overwrite table customer_segmentation SELECT customer_id,SUM(payment) AS spending, CASE WHEN SUM(payment) < 25000 THEN 'Under-Affluent' WHEN SUM(payment) BETWEEN 25000 AND 75000 THEN 'Semi-affluent' WHEN SUM(payment) > 75000 THEN 'Affluent' END AS Categories FROM ecom_data GROUP BY customer_id ORDER BY spending DESC;

-> create empty table in database

create table customer_segmentation(Customer_id varchar(255), Spending float, Categories varchar(30));

-> exporting data from hive to database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table customer_segmentation --export-dir /user/hive/warehouse/problem_1/result.txt/000000_0 --input-fields-terminated-by ' '

2. Monthly Trend Forecasting

The monthly trend of sales

with cte as (select year(timestamp) as year, month(timestamp) as month, product_category, sum(quantity) as count, rank() over(partition by year(timestamp), month(timestamp) order by sum(quantity) desc) as r, sum(payment) as spend from ecom group by year(timestamp), month(timestamp), product_category) select cte.year, cte.month, cte.product_category, cte.count, spend from cte where r=1 order by year desc, spend desc;

-> create external table

create external table monthly_trend(year int, month int, product string, total_no_product int, total_sales float) row format delimited fields terminated by ' ' location '/user/hive/warehouse/program_2/result.txt';

-> load data into external table

with cte as (select year(timestamp) as year, month(timestamp) as month, product_category, sum(quantity) as count, rank() over(partition by year(timestamp), month(timestamp) order by sum(quantity) desc) as r, sum(payment) as spend from ecom group by year(timestamp), month(timestamp), product_category) insert overwrite table monthly_trend select cte.year, cte.month, cte.product_category, cte.count, spend from cte where r=1 order by year desc, spend desc;

-> creating empty table in database

create table monthly_trend(year int, month int, product varchar(100), total_no_product int, total_sales float);

-> export data to empty table

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table monthly_trend --export-dir /user/hive/warehouse/program_2/result.txt/000000_0 --input-fields-terminated-by ' '

3. Hourly Sales Analysis

Which hour has more no. of sales?

select hour(timestamp) as hour, sum(quantity) as no_of_sale from ecom group by hour(timestamp) order by no_of_sale desc limit 1;

-> create external table

create external table hourly_sales(hour int, no_of_sales int) row format delimited fields terminated by ' ' location '/user/hive/warehouse/problem_3/result.txt';

-> loading data into external table

insert overwrite table hourly_sales select hour(timestamp) as hour, sum(quantity) as no_of_sale from ecom group by hour(timestamp) order by no_of_sale desc limit 1;

-> create empty table in database

create table hourly_sales(hour int, no_of_sales int);

-> export data in empty table

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table hourly_sales --export-dir /user/hive/warehouse/problem_3/result.txt/000000_0 --input-fields-terminated-by ' '

4. Product Based Analysis

Which category product has sold more?

select product_category, sum(quantity) as count from ecom group by product_category order by count desc limit 1;

-> create external table

create external table product_1(product string, count int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_4_1/result.txt'

-> load data in external table

insert overwrite table product_1 select product_category, sum(quantity) as count from ecom group by product_category order by count desc limit 1;

->create empty table

create table product_sold(product varchar(100), count int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table product_sold --export-dir /user/hive/wareshouse/problem_4_1/result.txt/000000_0 --input-fields-terminated-by ' '

Which category product has more rating?

select product_category, avg(rating) as avg_rating from ecom group by product_category order by avg_rating desc limit 1;

-> create external table

create external table product_2(product string, avg_rating float) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_4_2/result.txt'

-> load data in external table

insert overwrite table product_2 select product_category, avg(rating) as avg_rating from ecom group by product_category order by avg_rating desc limit 1;

->create empty table

create table product_rating(product varchar(100), rating float);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table product_rating --export-dir /user/hive/wareshouse/problem_4_2/result.txt/000000_0 --input-fields-terminated-by ' '

Which product has sold more?

select product_category, sum(quantity) as no_of_sold from ecom group by product_category order by no_of_sold desc limit 1;

-> create external table

create external table product_3(product string, no_of_sold int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_4_3/result.txt';

-> load data in external table

insert overwrite table product_3 select product_category, sum(quantity) as no_of_sold from ecom group by product_category order by no_of_sold desc limit 1;

->create empty table

create table highest_product_sold(product varchar(100), no_of_sold int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table highest_product_sold --export-dir /user/hive/wareshouse/problem_4_3/result.txt/000000_0 --input-fields-terminated-by ' '

Top 10 highest & least product rating?

Highest rating

select product_category, avg(rating) as avg_rating from ecom group by product_category order by avg_rating desc limit 10;

-> create external table

create external table product_4i(product string, avg_rating float) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_4_4i/result.txt';

-> load data in external table

insert overwrite table product_4i select product_category, avg(rating) as avg_rating from ecom group by product_category order by avg_rating desc limit 10;

->create empty table

create table top_product_rating(product varchar(100), rating float);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table top_product_rating --export-dir /user/hive/wareshouse/problem_4_4i/result.txt/000000_0 --input-fields-terminated-by ' '

----------least rating----------

select product_category, avg(rating) as avg_rating from ecom group by product_category order by avg_rating asc limit 10;

-> create external table

create external table product_4ii(product string, avg_rating float) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_4_4ii/result.txt';

-> load data in external table

insert overwrite table product_4ii select product_category, avg(rating) as avg_rating from ecom group by product_category order by avg_rating asc limit 10;

->create empty table

create table least_product_rating(product varchar(100), rating float);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table least_product_rating --export-dir /user/hive/wareshouse/problem_4_4ii/result.txt/000000_0 --input-fields-terminated-by ' '

Order Count for each rating

select rating, count(rating)as count from ecom group by rating order by count desc;

-> create external table

create external table product_5(product string, avg_rating int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_4_5/result.txt';

-> load data in external table

insert overwrite table product_5 select rating, count(rating)as count from ecom group by rating order by count desc;

->create empty table

create table count_of_rating(rating int, count_of_rating int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table count_of_rating --export-dir /user/hive/wareshouse/problem_4_5/result.txt/000000_0 --input-fields-terminated-by ' '

5. Payment Preference

What are the most commonly used payment types?

select payment_type, count(payment_type) as no_of_times from ecom group by payment_type order by no_of_times desc;

-> create external table

create external table payment_1(payment_type string, no_of_times int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_5/result.txt';

-> load data in external table

insert overwrite table payment_1 select payment_type, count(payment_type) as no_of_times from ecom group by payment_type order by no_of_times desc;

->create empty table

create table count_of_payment_mode(payment_type varchar(30), no_of_times int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table count_of_payment_mode --export-dir /user/hive/wareshouse/problem_5/result.txt/000000_0 --input-fields-terminated-by ' '

Count of Orders With each No. of Payment Installments

select payment_installments, count(*) as count_orders from ecom where payment_installments is not null group by payment_installments;

-> create external table

create external table payment_2(payment_installments int, count_orders int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_5ii/result.txt';

-> load data in external table

insert overwrite table payment_2 select payment_installments, count(*) as count_orders from ecom where payment_installments is not null group by payment_installments;

->create empty table

create table Count_of_Orders(payment_installments int, count_of_orders int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table Count_of_Orders --export-dir /user/hive/wareshouse/problem_5ii/result.txt/000000_0 --input-fields-terminated-by ' '

6. Potential Customer's Location

Where do most customers come from?

select customer_state, count(customer_state) as no_of_customer from ecom group by customer_state order by no_of_customer desc;

-> create external table

create external table location(customer_state string, no_of_customer int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_6/result.txt';

-> load data in external table

insert overwrite table location select customer_state, count(customer_state) as no_of_customer from ecom group by customer_state order by no_of_customer desc;

->create empty table

create table Customer_location(customer_state varchar(30), no_of_customer int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table Customer_location --export-dir /user/hive/wareshouse/problem_6/result.txt/000000_0 --input-fields-terminated-by ' '

7. Seller Rating

Which seller sold more?

select seller_id, count(quantity) as sold from ecom group by seller_id order by sold desc limit 1;

-> create external table

create external table seller_sold(seller_id string, sold int) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_7_1/result.txt';

-> load data in external table

insert overwrite table seller_sold select seller_id, count(quantity) as sold from ecom group by seller_id order by sold desc limit 1;

->create empty table

create table seller_sold(seller_id varchar(200), sold int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table seller_sold --export-dir /user/hive/wareshouse/problem_7_1/result.txt/000000_0 --input-fields-terminated-by ' '

Which seller got more rating?

SELECT seller_id, rating FROM (SELECT seller_id, AVG(rating) AS rating, RANK() OVER (ORDER BY AVG(rating) DESC) AS ranking FROM ecom GROUP BY seller_id) temp WHERE ranking = 1 ORDER BY rating DESC;

-> create external table

create external table seller_rating(seller_id string, rating float) row format delimited fields terminated by ' ' location '/user/hive/wareshouse/problem_7_2/result.txt';

-> load data in external table

insert overwrite table seller_rating SELECT seller_id, rating FROM (SELECT seller_id, AVG(rating) AS rating, RANK() OVER (ORDER BY AVG(rating) DESC) AS ranking FROM ecom GROUP BY seller_id) temp WHERE ranking = 1 ORDER BY rating DESC;

->create empty table

create table seller_rating_higher(seller_id varchar(200), rating float);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table seller_rating_higher --export-dir /user/hive/wareshouse/problem_7_2/result.txt/000000_0 --input-fields-terminated-by ' '

8. Logistics based Optimization Insights

Which city buys heavy weight products and low weight products?

Higher Weight

select customer_city, sum(product_weight_g) as weight from ecom group by customer_city order by weight desc limit 10; -- high weight product

-> create external table

create external table higher_weight(customer_city string, weight int) row format delimited fields terminated by ',' location '/user/hive/wareshouse/problem_8/result.txt';

-> load data in external table

insert overwrite table higher_weight select customer_city, sum(product_weight_g) as weight from ecom group by customer_city order by weight desc limit 10;

->create empty table

create table higher_weight(customer_city varchar(20), weight int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table higher_weight --export-dir /user/hive/wareshouse/problem_8/result.txt/000000_0 --input-fields-terminated-by ','

----------lower weight----------

select customer_city, sum(product_weight_g) as weight from ecom group by customer_city order by weight asc limit 10; -- low weight product

-> create external table

create external table lower_weight(customer_city string, weight int) row format delimited fields terminated by ',' location '/user/hive/wareshouse/problem_8_1/result.txt';

-> load data in external table

insert overwrite table lower_weight select customer_city, sum(product_weight_g) as weight from ecom group by customer_city order by weight asc limit 10;

->create empty table

create table low_weight(customer_city varchar(20), weight int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table low_weight --export-dir /user/hive/wareshouse/problem_8_1/result.txt/000000_0 --input-fields-terminated-by ','

How much products sold within seller state?

select sum(quantity) as count from ecom where seller_state = customer_state;

-> create external table

create external table product_sold(quantity int) row format delimited fields terminated by ',' location '/user/hive/wareshouse/problem_8_2/result.txt';

-> load data in external table

insert overwrite table product_sold select sum(quantity) as count from ecom where seller_state = customer_state;

->create empty table

create table product_sold(No_of_product_sold_within_seller_state int);

->export data to the database

sqoop export --connect jdbc:mysql://localhost:3306/ecom --username root --password cloudera --table product_sold_within_seller_state --export-dir /user/hive/wareshouse/problem_8_2/result.txt/000000_0 --input-fields-terminated-by ','