**HIVE ECOMMERCE DATA ANALYSIS PROJECT REPORT**

**Save data into HDFS from LFS:**

hadoop fs -mkdir e-commerce

hadoop fs -put ecom\_data\_csv.csv e-commerce

**Create internal table schema in hive to load csv into table:**

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");

**Load data into table:**

> load data inpath 'e-commerce/ecom\_data\_csv.csv' into table ecom\_data;

**Cleaning and transforming data in table:**

–removing extra rows as per quantity column

– removing duplicate values

–formatting timestamp as per default

**Table for cleaned data:**

CREATE TABLE ecom\_data1 ( 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 cleaned data into new table:**

INSERT overwrite TABLE ecom\_data1

SELECT DISTINCT

order\_id,

customer\_id,

quantity,

price\_mrp,

payment,

from\_unixtime ( unix\_timestamp (TIMESTAMP, 'dd-MM-yyyy HH:mm'), 'yyyy-MM-dd HH:mm:ss' ),

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 customer\_id, order\_id, product\_id

ORDER BY

quantity desc ) AS r

FROM

ecom\_data

)

a

WHERE

r = 1;

**Creating partitions based on order status:**

CREATE TABLE ecom\_partitions ( 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, 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 ) partitioned BY (order\_status string) ROW format delimited fields terminated BY ',' tblproperties ("skip.header.line.count" = "1");

**Commands to enable partitions in hive:**

hive> set hive.exec.dynamic.partition=true;

hive> set hive.exec.dynamic.partition.mode=nonstrict;

INSERT overwrite TABLE ecom\_partitions PARTITION (order\_status)

SELECT DISTINCT

order\_id,

customer\_id,

quantity,

price\_mrp,

payment,

from\_unixtime ( unix\_timestamp (TIMESTAMP, 'dd-MM-yyyy HH:mm'), 'yyyy-MM-dd HH:mm:ss' ),

rating,

product\_category,

product\_id,

payment\_type,

product\_weight\_g,

product\_length\_cm,

product\_height\_cm,

product\_width\_cm,

customer\_city,

customer\_state,

seller\_id,

seller\_city,

seller\_state,

payment\_installments,

order\_status

FROM

(

SELECT

\*,

DENSE\_RANK() OVER ( PARTITION BY customer\_id, order\_id, product\_id

ORDER BY

quantity desc ) AS r

FROM

ecom\_data

)

a

WHERE

r = 1;

—-------------------------------------------------------------------------------------------------------------------------

**Hive jobs:**

**1. Customer Segmentation**

**Query: Categorizing customers based on their spendings**

1. CREATE EXTERNAL TABLE customer\_segmentation ( order\_id string, customer\_id string, payment FLOAT, payment\_type string, payment\_installments INT, customer\_spending\_category string ) ROW format delimited fields terminated BY ',' location '/user/hive/warehouse/e\_com/customer\_segmentation.txt';

insert overwrite table customer\_segmentation

select

order\_id,

customer\_id,

payment,

payment\_type,

payment\_installments,

case

when

payment between average\_payment\*0.8 and average\_payment\*1.2

then

"Mid - range Purchasers"

when

payment > average\_payment\*1.2

then

"Exclusive Purchasers"

when

payment < average\_payment\*0.8

then

"Affordable Purchasers"

end

as customer\_spending\_category

from

(

select

order\_id,

customer\_id,

payment,

product\_category,

payment\_type,

payment\_installments,

avg(payment) over() as average\_payment

from

ecom\_data1

)

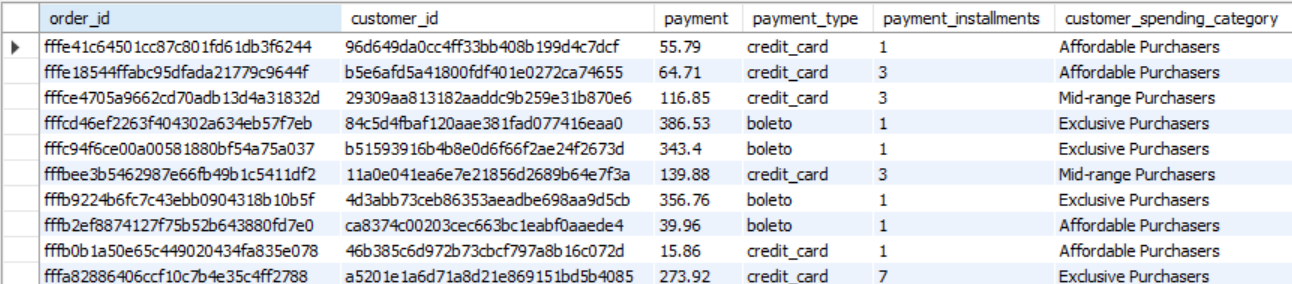
A;

**Exporting to client Database:**

create table customer\_segmentation(order\_id varchar(200), customer\_id varchar(200), payment float, payment\_type varchar(40), payment\_installments int, customer\_spending\_category varchar(40));

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table customer\_segmentation --export-dir /user/hive/warehouse/e\_com/customer\_segmentation.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



2. create external table customer\_spending\_segmentation(customer\_spending\_category string, payment\_type string, customers\_count int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/customer\_spending\_segmentation.txt';

insert overwrite table customer\_spending\_segmentation

select

customer\_spending\_category,

payment\_type,

count(customer\_id) as customers\_count

from

customer\_segmentation

group by

customer\_spending\_category,

payment\_type

order by

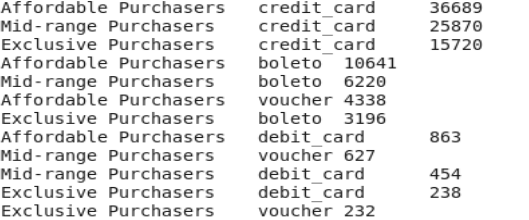
customers\_count desc;

**Exporting to client Database:**

create table customer\_spending\_segmentation(customer\_spending\_category varchar(40), payment\_type varchar(40), customers\_count int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table customer\_spending\_category --export-dir /user/hive/warehouse/e\_com/customer\_spending\_category.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



—-------------------------------------------------------------------------------------------------------------------------

**2. Monthly Trend Forecasting**

**Monthly trend of sales**

create external table monthly\_sales(year int, month int, sales float)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/monthly\_sales.txt';

insert overwrite table monthly\_sales

select

year(timestamp),

month(timestamp),

sum(payment)

from

ecom\_data1

group by

year(timestamp),

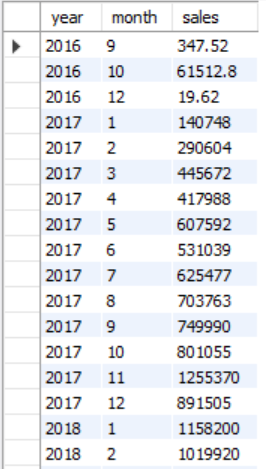
month(timestamp);

**Exporting to client Database:**

create table monthly\_sales(year int, month int, sales float);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table monthly\_sales --export-dir /user/hive/warehouse/e\_com/monthly\_sales.txt/000000\_0 --input-fields-terminated-by ','

**output:**



—---------------------------------------------------------------------------------------------------------------------------

**3. Hourly Sales Analysis Which hour has more no. of sales?**

create external table hourly\_sales(hour int, no\_of\_sales int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/hourly\_sales.txt';

insert overwrite table hourly\_sales

select

hour(timestamp),

sum(quantity) as no\_of\_sales

from

ecom\_data1

group by

hour(timestamp)

order by

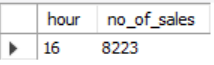
no\_of\_sales desc limit 1;

**Exporting to client Database:**

create table hourly\_sales(hour int, no\_of\_sales int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table hourly\_sales --export-dir /user/hive/warehouse/e\_com/hourly\_sales.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



—---------------------------------------------------------------------------------------------------------------------------

**4. Product Based Analysis**

**Which category product has sold more?**

create external table product\_category\_sales(product\_category string, sales int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/product\_category\_sales.txt';

insert overwrite table product\_category\_sales

select

product\_category,

sum(quantity) as sales

from

ecom\_data\_buckets

where

order\_status != 'canceled'

or order\_status != 'unavailable'

group by

product\_category

order by

sales desc limit 1;

**Exporting to client Database:**

create table product\_category\_sales(product\_category varchar(60), sales int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table product\_category\_sales --export-dir /user/hive/warehouse/e\_com/product\_category\_sales.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Which category product has more rating?**

create external table highest\_rating\_product(product\_category string, rating float)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/hightest\_rating\_prodcut.txt';

insert overwrite table highest\_rating\_product

select

product\_category,

avg(rating) as rating

from

ecom\_data\_buckets

where

order\_status = 'delivered'

or order\_status != 'shipped'

group by

product\_category

order by

sales desc limit 1;

**Exporting to client Database:**

create table highest\_rating\_product(product\_category varchar(60), rating float);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table highest\_rating\_product --export-dir /user/hive/warehouse/e\_com/hightest\_rating\_prodcut.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Which product has sold more?**

create external table highest\_sold\_product(product\_id string, sales\_quantity int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/hightest\_sold\_product.txt';

insert overwrite table highest\_sold\_product

select

product\_id,

sum(quantity) as sales\_quantity

from

ecom\_data\_buckets

where

order\_status = 'delivered'

or order\_status = 'shipped'

group by

product\_id

order by

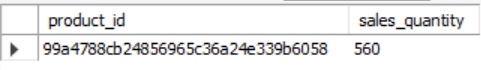
sales\_quantity desc limit 1;

**Exporting to client Database:**

create table highest\_sold\_product(product\_id varchar(200), sales\_quantity int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table highest\_sold\_product --export-dir /user/hive/warehouse/e\_com/hightest\_sold\_prodcut.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Top 10 highest & least product rating?**

create external table top\_10\_rated\_products(product\_id string, rating float)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/top\_10\_rated\_products.txt';

insert overwrite table top\_10\_rated\_products

select

product\_id,

avg(rating) as rating

from

ecom\_data\_buckets

where

order\_status = 'delivered'

or order\_status = 'shipped'

group by

product\_id

order by

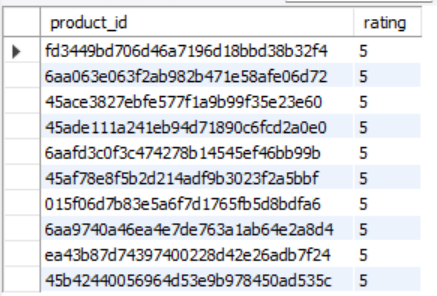
rating desc limit 10;

**Exporting to client Database:**

create table top\_10\_rated\_products(product\_id varchar(200), rating float);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table top\_10\_rated\_products --export-dir /user/hive/warehouse/e\_com/top\_10\_rated\_products.txt/000000\_0 --input-fields-terminated-by ','

**OUTPUT:**



create external table least\_10\_rated\_products(product\_id string, rating float)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/top\_10\_rated\_products.txt';

insert overwrite table top\_10\_rated\_products

select

product\_id,

avg(rating) as rating

from

ecom\_data\_buckets

where

order\_status = 'delivered'

or order\_status = 'shipped'

group by

product\_id

order by

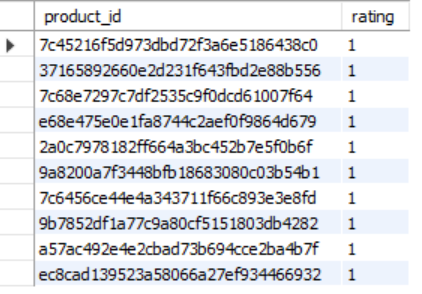
rating limit 10;

**Exporting to client Database:**

create table least\_10\_rated\_products(product\_id varchar(200), rating float);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table least\_10\_rated\_products --export-dir /user/hive/warehouse/e\_com/least\_10\_rated\_products.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Order Count for each rating**

create external table ratings\_order\_count(rating int, order\_count int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/ratings\_order\_count.txt';

insert overwrite table ratings\_order\_count

select

rating,

count(order\_id)

from

ecom\_partitions

where

order\_status = 'shipped'

or order\_status = 'delivered'

group by

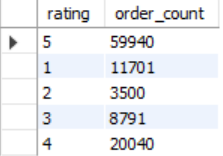
Rating;

**Exporting to client Database:**

create table ratings\_order\_count(rating int, order\_count int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table ratings\_order\_count --export-dir /user/hive/warehouse/e\_com/ratings\_order\_count.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**5. Payment Preference**

**What are the most commonly used payment types?**

create external table most\_common\_payment\_types(payment\_type string, no\_of\_payments int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/most\_common\_payment\_types.txt';

insert overwrite table most\_common\_payment\_types

select

payment\_type,

count(order\_id) as no\_of\_payments

from

ecom\_partitions

where

order\_status != 'unavailable'

or order\_status != 'canceled'

group by

payment\_type

order by

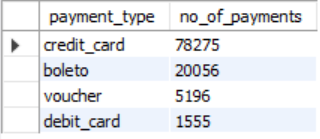
no\_of\_payments desc;

**Exporting to client Database:**

create table most\_common\_payment\_types(payment\_type varchar(40), no\_of\_payments int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table most\_common\_payment\_types --export-dir /user/hive/warehouse/e\_com/most\_common\_payment\_types.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Count of Orders With each No. of Payment Installments**

create external table payment\_installment\_counts(payment\_installments int, no\_of\_orders int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/payment\_installment\_counts.txt';

insert overwrite table payment\_installment

select

payment\_type,

count(order\_id) as no\_of\_payments

from

ecom\_partitions

where

order\_status != 'unavailable'

or order\_status != 'canceled'

group by

payment\_type

order by

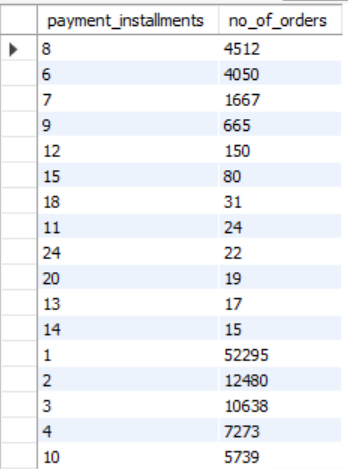
no\_of\_payments desc;

**Exporting to client Database:**

create table payment\_installment\_counts(payment\_installments int, no\_of\_orders int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table payment\_installment\_counts --export-dir /user/hive/warehouse/e\_com/payment\_installment\_counts.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**6. Potential Customer's Location**

**Where do most customers come from?**

create external table customer\_location\_trends(customer\_state string, no\_of\_customers int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/customer\_location\_trends.txt';

insert overwrite table customer\_location\_trends

select

customer\_state,

count(customer\_id) as no\_of\_customers

from

ecom\_data1

group by

customer\_state

order by

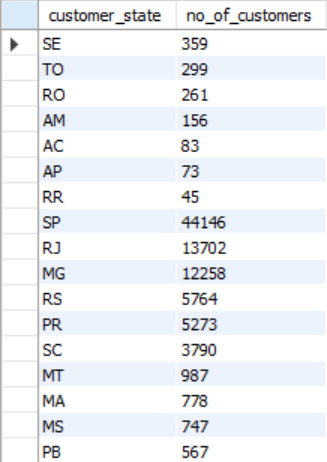
no\_of\_customers desc;

**Exporting to client Database:**

create table customer\_location\_trends(customer\_state varchar(10), no\_of\_customers int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table customer\_location\_trends --export-dir /user/hive/warehouse/e\_com/customer\_location\_trends.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**7. Seller Rating**

**Which seller sold more?**

create external table sales\_by\_seller(seller\_id string, sales\_quantity int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/sales\_by\_seller.txt';

insert overwrite table sales\_by\_seller

select

seller\_id,

sum(quantity) as sales\_quantity

from

ecom\_data1

group by

seller\_id

order by

sales\_quantity desc limit 1;

**Exporting to client Database:**

create table sales\_by\_seller(seller\_id varchar(200), sales\_quantity int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table sales\_by\_seller --export-dir /user/hive/warehouse/e\_com/sales\_by\_seller.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Which seller got more rating?**

create external table ratings\_by\_seller(seller\_id string, average\_rating float)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/ratings\_by\_seller.txt';

insert overwrite table ratings\_by\_seller

select

seller\_id,

avg(rating) as average\_rating

from

ecom\_partitions

where

order\_status = 'shipped'

or order\_status = 'delivered'

group by

seller\_id

order by

average\_rating desc limit 1;

**Exporting to client Database:**

create table ratings\_by\_seller(seller\_id varchar(200), average\_rating float);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table ratings\_by\_seller --export-dir /user/hive/warehouse/e\_com/ratings\_by\_seller.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**8. Logistics based Optimization Insights**

**Which city buys heavy weight products and low weight products?**

create external table heavy\_product\_weight\_cities(customer\_city string, sales\_quantity int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/heavy\_product\_weight\_cities.txt';

insert overwrite table heavy\_product\_weight\_cities

SELECT

e1.customer\_city,

sum(e1.quantity) as quantity

FROM

ecom\_data1 e1

JOIN

(

SELECT

AVG(product\_weight\_g) AS avg\_weight

FROM

ecom\_data1

)

e2

WHERE

e1.product\_weight\_g > e2.avg\_weight

group by

e1.customer\_city

order by

quantity desc limit 5;

**Exporting to client Database:**

create table heavy\_product\_weight\_cities(customer\_city varchar(30), sales\_quantity int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table heavy\_product\_weight\_cities --export-dir /user/hive/warehouse/e\_com/heavy\_product\_weight\_cities.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



create external table least\_product\_weight\_cities(customer\_city string, sales\_quantity int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/least\_product\_weight\_cities.txt';

insert overwrite table lease\_product\_weight\_cities

SELECT

e1.customer\_city,

sum(e1.quantity) as quantity

FROM

ecom\_data1 e1

JOIN

(

SELECT

AVG(product\_weight\_g) AS avg\_weight

FROM

ecom\_data1

)

e2

WHERE

e1.product\_weight\_g > e2.avg\_weight

group by

e1.customer\_city

order by

quantity limit 5;

**Exporting to client Database:**

create table least\_product\_weight\_cities(customer\_city varchar(30), sales\_quantity int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table least\_product\_weight\_cities --export-dir /user/hive/warehouse/e\_com/least\_product\_weight\_cities.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**How much products sold within seller state?**

create external table seller\_state\_product\_sales(seller\_state string, sales\_quantity int)row format delimited fields terminated by ',' location '/user/hive/warehouse/e\_com/seller\_state\_product\_sales.txt';\

insert overwrite table seller\_state\_product\_sales

select

seller\_state,

sum(quantity) as sales\_quantity

from

ecom\_data1

group by

seller\_state

order by

sales\_quantity desc limit 1;

**Exporting to client Database:**

create table seller\_state\_product\_sales(seller\_state varchar(10), sales\_quantity int);

sqoop export --connect jdbc:mysql://localhost:3306/e\_com --username root --password cloudera --table seller\_state\_product\_sales --export-dir /user/hive/warehouse/e\_com/seller\_state\_product\_sales.txt/000000\_0 --input-fields-terminated-by ','

**Output:**



**Visualization:**

