1. **Copying CSV from Local to Cloudera**

hadoop fs -copyFromLocal Desktop/QualifData/location.csv /user/cloudera

hadoop fs -copyFromLocal Desktop/QualifData/product\_category.csv /user/cloudera

1. **Creating External Table on Location**

CREATE EXTERNAL TABLE location(

id INT,

country VARCHAR(100),

region VARCHAR(100),

city VARCHAR(100)

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

1. **Load the data CSV on the table we make on number 2**

LOAD DATA INPATH '/user/cloudera/location.csv' INTO TABLE bluejack\_store.location

1. **Same as number two create a external table for product\_category**

CREATE EXTERNAL TABLE product\_category(

id INT,

product\_category\_name VARCHAR(100)

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

1. **Load the data CSV on the table we make on number 4**

LOAD DATA INPATH '/user/cloudera/product\_category.csv' INTO TABLE bluejack\_store.product\_category

1. **Open mysql on terminal and import insert.sql to local database mysql**

Source Data/insert.sql

1. **Import the sql to the cloudera (HIVE / IMPALA) Database**

sudo sqoop import-all-tables --connect jdbc:mysql://quickstart:3306/bluejack\_store --username=root -P --hive-import --hive-database=bluejack\_store

* 1. **Show the most profitable product category in 2019.**

select distinct pc.product\_category\_name

from product\_category pc,

(

select p.product\_category\_id, SUM((p.product\_price \* (1 - p.discount)) \* if(sd.is\_cancelled is NULL, sd.quantity, 0)) as total

from products p

join sales\_detail sd on p.product\_id = sd.product\_id

join sales s on s.sales\_id = sd.sales\_id

where YEAR(s.sales\_date) = 2019

group by p.product\_category\_id

order by total desc

limit 1

) as x

where pc.id = x.product\_category\_id

**b. Show location which transaction happens the most last year.**

select

l.country as Country,

l.region as Region,

l.city as City,

count(s.sales\_id) as Total Sales

from

`location` l

join

sales s on l.id = s.location\_id

where

year(s.sales\_date) in (

select max(year(s2.sales\_date))

from sales s2

)

group by country, region, city

order by total desc

limit 1

**c. Show the most profitable product by calculating its revenue. When there is a discount cut the product price by discount. Show data with profit more than 15.000.000.**

select p.product\_name

from products p,

(

select p.product\_id, SUM((p.product\_price \* (1 - p.discount)) \* if(sd.is\_cancelled is NULL, sd.quantity, 0)) as total

from products p

join sales\_detail sd on p.product\_id = sd.product\_id

group by p.product\_id

having total > 15000000

order by total desc

) as x

where p.product\_id = x.product\_id

**d. Show customer who handled transactions more than the average of the total transactions by each customer.**

select c.customer\_name, count(s.sales\_id) as customerTransaction from customer c join sales s on c.customer\_id = s.customer\_id,

(select avg(x.customerTransaction) as avgTotalTransactionEachCustomer from

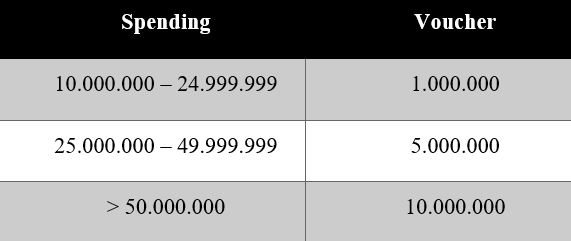
(select count(s.sales\_id) as customerTransaction from customer c join sales s on c.customer\_id = s.customer\_id group by c.customer\_id) as x

)a

group by c.customer\_name, c.customer\_id, a.avgTotalTransactionEachCustomer

having customerTransaction > a.avgTotalTransactionEachCustomer

e**. Show customers spending for last Christmas (December 2019). All customer who spends more than 10.000.000 will get a voucher for the next transaction with terms and condition as below**



select

c.customer\_name,

total,

case

when x.total >= 10000000 and x.total <= 24999999 then "1.000.000"

when x.total >= 25000000 and x.total <= 49999999 then "5.000.000"

when x.total >= 50000000 then "10.000.000"

else "0"

end as Voucher

from

customer c, (

select s.customer\_id,

SUM((p.product\_price \* (1 - p.discount)) \* if(sd.is\_cancelled is NULL, sd.quantity, 0)) as total

from products p

join sales\_detail sd on p.product\_id = sd.product\_id

join sales s on s.sales\_id = sd.sales\_id

where YEAR(s.sales\_date) = 2019

and month(s.sales\_date) = 12

group by s.customer\_id

) as x

where c.customer\_id = x.customer\_id