**Qualif Big Data Processing AY23-2 | Case 01**

**Notes:**

Pindahin file dari PC kita ke Desktop Cloudera dengan cara drag n drop.

Ada 2 file:

1. Case01

2. Case02

Pindahin file dari Cloudera ke Hue dengan cara:

1. cd ke Desktop dahulu

2. hadoop fs -copyFromLocal Case01

3. hadoop fs -copyFromLocal Case02

**1. Load data from CSV to Hive**

CREATE DATABASE bluejackstore;

USE bluejackstore;

CREATE EXTERNAL TABLE location (

location\_id INT,

country VARCHAR(100),

region VARCHAR(100),

city VARCHAR(100)

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LOAD DATA INPATH '/user/cloudera/Case01/location.csv'

INTO TABLE location

SELECT \* FROM location

-- Tabel Product Category

CREATE EXTERNAL TABLE product\_category (

product\_category\_id INT,

product\_category\_name VARCHAR(100)

)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY ','

LOAD DATA INPATH '/user/cloudera/Case01/product\_category.csv'

INTO TABLE product\_category

SELECT \* FROM product\_category

**2. Load data from MySQL to Hive**

- mysql -u root -p [password = cloudera]

- CREATE DATABASE bluejackstore

- Buka terminal di directory yang sama dengan .sql

- source insert.sql

- sudo sqoop import-all-tables --connect jdbc:mysql://quickstart:3306/bluejackstore --username=root -P --hive-import --hive-database=bluejackstore [password = clp

- balik ke hive dan refresh tablenya

**3. Query Analysis**

- Pindah ke impala saja dulu dan refresh databasenya

- Saat ingin refresh pilih yang ke-3 [Invalidate all metadata and rebuild index]

**A. Show the momst profitable product category in 2019.**

SELECT

pc.product\_category\_name AS ProductCategoryName,

SUM((p.product\_price - (p.product\_price \* p.discount)) \* quantity) AS Revenue

FROM sales s

JOIN sales\_detail sd ON s.sales\_id = sd.sales\_id

JOIN products p ON sd.product\_id = p.product\_id

JOIN product\_category pc ON p.product\_category\_id = pc.product\_category\_id

WHERE sd.is\_cancelled IS NULL AND YEAR(s.sales\_date) = 2019

GROUP BY pc.product\_category\_name

ORDER BY Revenue ASC

LIMIT 1

**B. Show the location where the most transactions occurred in the most recent year that transactions occurred.**

SELECT

l.country AS Country,

l.region AS Region,

l.city AS City,

COUNT(s.sales\_id) AS TotalSales

FROM `location` l

JOIN sales s ON l.location\_id = s.location\_id

WHERE YEAR(s.sales\_date) = (

SELECT

MAX(YEAR(s.sales\_date))

FROM sales s

)

GROUP BY Country, Region, City

ORDER BY TotalSales 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 AS ProductName,

SUM((p.product\_price - (p.product\_price \* p.discount)) \* sd.quantity) AS Revenue

FROM sales\_detail sd

JOIN products p ON sd.product\_id = p.product\_id

WHERE sd.is\_cancelled IS NULL

GROUP BY p.product\_name

HAVING Revenue > 15000000

ORDER BY Revenue DESC

LIMIT 1

**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 TotalTransaction

FROM customer c

JOIN sales s ON c.customer\_id = s.customer\_id,

(

SELECT

AVG(SQ.TotalTransaction) AS AvgTransaction

FROM

(

SELECT

COUNT(s.sales\_id) AS TotalTransaction

FROM Sales s

JOIN Customer c ON s.customer\_id = c.customer\_id

GROUP BY c.customer\_id

) AS SQ

) AS RS

GROUP BY c.customer\_name, c.customer\_id, rs.AvgTransaction

HAVING TotalTransaction > RS.AvgTransaction

**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,

RS.CustomerSpending,

CASE

WHEN RS.CustomerSpending >= 10000000 AND RS.CustomerSpending <= 24999999 THEN "1.000.000"

WHEN RS.CustomerSpending >= 25000000 AND RS.CustomerSpending <= 49999999 THEN "5.000.000"

WHEN RS.CustomerSpending > 50000000 THEN "10.000.000"

END AS Voucher

FROM customer c,

(

SELECT

c.customer\_id,

SUM((p.product\_price - p.product\_price \* p.discount) \* sd.quantity) AS CustomerSpending

FROM sales s

JOIN sales\_detail sd ON s.sales\_id = sd.sales\_id

JOIN customer c on s.customer\_id = c.customer\_id

JOIN products p ON sd.product\_id = p.product\_id

WHERE sd.is\_cancelled IS NULL AND MONTH(s.sales\_date) = 12 AND YEAR(s.sales\_date) = 2019

GROUP BY c.customer\_id

) AS RS

WHERE c.customer\_id = RS.customer\_id