**Load Data:**

o Load the dataset into Pig, defining an appropriate schema for the data.

wget <https://raw.githubusercontent.com/reddax02/PIG/main/customer_purchases.csv>

hdfs dfs -put /home/maria\_dev/ customer\_purchases.csv /user/maria\_dev/

Pig

purchase = LOAD '/user/maria\_dev/customer\_purchases.csv' USING PigStorage(',') AS(customer\_id:int,purchase\_date:chararray,product\_id:int,product\_name:chararray,quantity:int,total\_amount:double);

2. **Calculate Total Purchase Amount per Customer:**

o Group the data by customer\_id.

o For each customer, calculate the total purchase amount (sum of

total\_amount).

grouped\_by\_customer = GROUP purchase BY customer\_id;

total\_purchases\_per\_customer = FOREACH grouped\_by\_customer GENERATE group AS customer\_id, SUM(purchase.total\_amount) AS total\_spent;

DUMP total\_purchases\_per\_customer;

3. **Find the Most Purchased Product:**

o Determine the product that has been purchased the most (highest total

quantity across all customers).

* Need to group by items and check which items was purchased more

grouped\_by\_product = GROUP purchase BY (product\_id, product\_name);

total\_quantity\_per\_product = FOREACH grouped\_by\_product GENERATE FLATTEN(group) AS (product\_id, product\_name), SUM(purchase.quantity) AS total\_quantity;

4. **Calculate Average Purchase Amount per Customer:**

o Calculate the average purchase amount per customer.

grouped\_by\_customer = GROUP purchase BY customer\_id;

customer\_totals = FOREACH grouped\_by\_customer GENERATE group AS customer\_id, SUM(purchase.total\_amount) AS total\_spent, COUNT(purchase) AS purchase\_count;

average\_purchase\_amount = FOREACH customer\_totals GENERATE customer\_id, total\_spent / purchase\_count AS avg\_purchase\_amount;

DUMP average\_purchase\_amount;

5. **Identify High-Value Customers:**

o Determine customers who have made purchases with a total amount

grouped\_by\_customer = GROUP purchase BY customer\_id;

total\_purchases\_per\_customer = FOREACH grouped\_by\_customer GENERATE group AS customer\_id, SUM(purchase.total\_amount) AS total\_spent;

threshold = 1000;

high\_value\_customers = FILTER total\_purchases\_per\_customer BY total\_spent > threshold;

STORE high\_value\_customers INTO '/user/maria\_dev/high\_value\_customers' USING PigStorage(',');

6. **Sort Customers by Purchase Frequency:**

o Sort the customers based on the number of purchases they have made.

grouped\_by\_customer = GROUP purchase BY customer\_id;

purchase\_frequency = FOREACH grouped\_by\_customer GENERATE group AS customer\_id, COUNT(purchase) AS purchase\_count;

ordered\_by\_frequency = ORDER purchase\_frequency BY purchase\_count DESC;

STORE ordered\_by\_frequency INTO '/user/maria\_dev/customers\_by\_purchase\_frequency' USING PigStorage(',');

7. **Generate a Report:**

o Create a report that includes the customer ID, total purchase amount,

most purchased product, average purchase amount, and whether the

customer is a high-value customer.

Record and upload an execution video demonstrating your code.

• Start the video by showing your name and the Lab topic.

• Navigate through your code, explaining key sections and highlighting any

customizations you made.

• Execute the code and showcase the output/results.

• Ensure the code and output are clearly visible in the video.

Note:

• The execution video is a mandatory part of the submission. Failure to provide the

execution video will result in the Lab not being graded.

• Late submissions will not be accepted or marked.

• Please ensure that you follow the instructions carefully and submit your Lab within the

given timeframe.