Exercise-4 Perform the following tasks in Pig Environment

Report Template

Name : SWETHA K
Roll Number : 235229143

1. Load the Data

- Load the transactions.csv, branches.csv, and customers.csv datasets into Pig relations.

```
transactions = LOAD 'transactions.csv' USING PigStorage(',') AS
(transaction_id:int, account_id:int, branch_id:int,
transaction_date:chararray, transaction_type:chararray,
transaction_amount:float);

branches = LOAD 'branches.csv' USING PigStorage(',') AS
(branch_id:int, branch_name:chararray,
branch_city:chararray);

customers = LOAD 'customers.csv' USING PigStorage(',') AS
(customer_id:int, account_id:int, customer_name:chararray,
customer_city:chararray);
```

2. Store the Data

- Store the loaded data from transactions.csv, branches.csv, and customers.csv into different output files.

```
STORE transactions INTO 'output_transactions' USING

PigStorage(','); STORE branches INTO 'output_branches' USING

PigStorage(','); STORE customers INTO 'output_customers' USING

PigStorage(',');
```

3. Filter Transactions

- Filter transactions where the transaction_amount is greater than

1000.

high_value_transactions = FILTER transactions BY transaction_amount > 1000;

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.13.0 0.12.0-cdh5.13.0        cloudera        2024-08-21 21:14:04     2024-08-21 21:14:
34      FILTER

Success!
```

4. Split the Data

- Split the transactions data into two separate datasets: one for deposits and one for withdrawals.

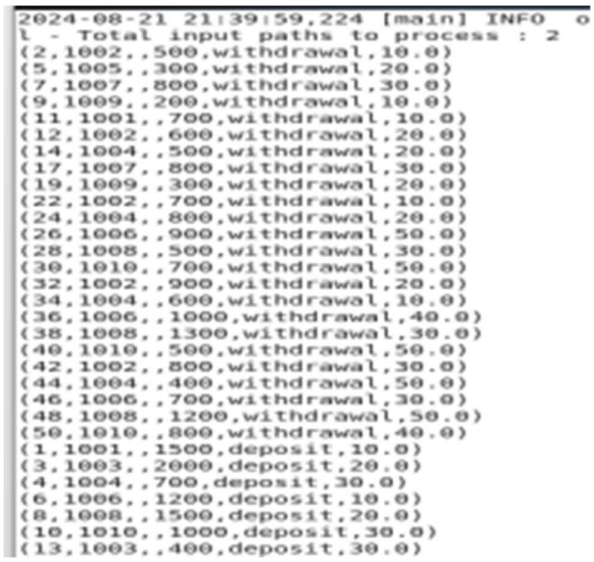
SPLIT transactions INTO deposits IF transaction_type == 'deposit',withdrawals IF transaction_type == 'withdrawal';

```
(,,,transaction_amount,transaction_type,)
(1,1001,,1500,deposit,10.0)
(2,1002,,500,withdrawal,10.0)
(3,1003,,2000,deposit,20.0)
(4,1004,,700,deposit,30.0)
(5,1005,,300,withdrawal,20.0)
(6,1006,,1200,deposit,10.0)
(7,1007,,800,withdrawal,30.0)
(8,1008,,1500,deposit,20.0)
(9,1009,,200,withdrawal,10.0)
(10,1010,,1000,deposit,30.0)
(11,1001,,700,withdrawal,10.0)
(12,1002,,600,withdrawal,20.0)
(13,1003,,400,deposit,30.0)
(14,1004,,500,withdrawal,20.0)
(15,1005,,900,deposit,40.0)
(16,1006,,1500,deposit,50.0)
(17,1007,,800,withdrawal,30.0)
(18,1008,,2000,deposit,10.0)
(19,1009,,300,withdrawal,20.0)
(20,1010,,1000,deposit,50.0)
(21,1001,,1200,deposit,40.0)
(22,1002,,700,withdrawal,10.0)
(23,1003,,1300,deposit,30.0)
(24,1004,,800,withdrawal,20.0)
(25,1005,,600,deposit,10.0)
(26,1006,,900,withdrawal,50.0)
(27,1007,,2000,deposit,40.0)
(28,1008,,500,withdrawal,30.0)
(29,1009,,1500,deposit,20.0)
(30,1010,,700,withdrawal,50.0)
(31,1001,,300,deposit,40.0)
```

5. Combine and Merge Data

- Combine the datasets for deposits and withdrawals into a single dataset.

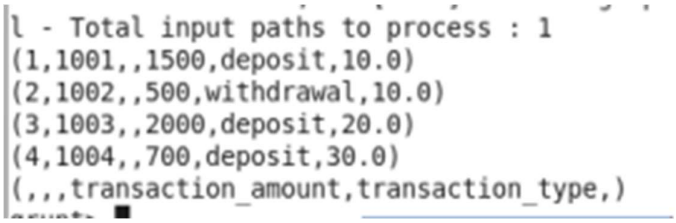
combined_transactions = UNION deposits,withdrawals;



6. Limit the Results

- Limit the output of transactions to the first 5 records.

limited_transactions = LIMIT transactions 5;



7. Group Transactions

- Group transactions by account_id and compute the total transaction amount per account.

grouped_transactions = GROUP transactions BY account_id;

```
l - Total input paths to process : 1
(1001,{(11,1001,,700,withdrawal,10.0),(1,1001,,1500,deposit,10.0),(21,1001,,1200,deposit,40.0),(3
1,1001,,300,deposit,40.0),(41,1001,,2000,deposit,20.0)})
(1002,{(12,1002,,600,withdrawal,20.0),(2,1002,,500,withdrawal,10.0),(32,1002,,900,withdrawal,20.0
),(22,1002,,700,withdrawal,10.0),(42,1002,,800,withdrawal,30.0)})
(1003,{(3,1003,,2000,deposit,20.0),(23,1003,,1300,deposit,30.0),(43,1003,,600,deposit,10.0),(13,1
003,,400,deposit,30.0),(33,1003,,1100,deposit,30.0)})
(1004,{(34,1004,,600,withdrawal,10.0),(14,1004,,500,withdrawal,20.0),(44,1004,,400,withdrawal,50.
0),(24,1004,,800,withdrawal,20.0),(4,1004,,700,deposit,30.0)})
(1005,{(35,1005,,1500,deposit,50.0),(25,1005,,600,deposit,10.0),(15,1005,,900,deposit,40.0),(45,1
005,,900,deposit,40.0),(5,1005,,300,withdrawal,20.0)})
(1006,{(6,1006,,1200,deposit,10.0),(36,1006,,1000,withdrawal,40.0),(26,1006,,900,withdrawal,50.0)
,(16,1006,,1500,deposit,50.0),(46,1006,,700,withdrawal,30.0)})
(1007,{(27,1007,,2000,deposit,40.0),(37,1007,,1200,deposit,20.0),(47,1007,,1500,deposit,20.0),(17
,1007,,800,withdrawal,30.0),(7,1007,,800,withdrawal,30.0)})
(1008,{(38,1008,,1300,withdrawal,30.0),(18,1008,,2000,deposit,10.0),(48,1008,,1200,withdrawal,50.
0),(28,1008,,500,withdrawal,30.0),(8,1008,,1500,deposit,20.0)})
(1009,{(19,1009,,300,withdrawal,20.0),(9,1009,,200,withdrawal,10.0),(29,1009,,1500,deposit,20.0),
(39,1009,,800,deposit,10.0),(49,1009,,1100,deposit,30.0)})
(1010,{(50,1010,,800,withdrawal,40.0),(40,1010,,500,withdrawal,50.0),(30,1010,,700,withdrawal,50.
0),(20,1010,,1000,deposit,50.0),(10,1010,,1000,deposit,30.0)})
(,{{,,,transaction_amount,transaction_type,)})
```

total_per_account = FOREACH grouped_transactions GENERATE group AS account_id, SUM(transactions.transaction_amount) AS total_amount;

```
l - Total input paths to process : 1
(1001,120.0)
(1002,90.0)
(1003,120.0)
(1004,130.0)
(1005,160.0)
(1006,180.0)
(1007,140.0)
(1008,140.0)
(1009,90.0)
(1010,220.0)
(,)
```

8. Sort Transactions

- Sort the transactions by transaction_amount in descending order.

sorted_transactions = ORDER transactions BY transaction_amount DESC;

```
l - Total input paths to process : 1
(40,1010,,500,withdrawal,50.0)
(16,1006,,1500,deposit,50.0)
(20,1010,,1000,deposit,50.0)
(26,1006,,900,withdrawal,50.0)
(30,1010,,700,withdrawal,50.0)
(44,1004,,400,withdrawal,50.0)
(35,1005,,1500,deposit,50.0)
(48,1008,,1200,withdrawal,50.0)
(31,1001,,300,deposit,40.0)
(36,1006,,1000,withdrawal,40.0)
(27,1007,,2000,deposit,40.0)
(50,1010,,800,withdrawal,40.0)
```

9. Join Transactions with Branch Information

- Join the transactions dataset with branches dataset on branch_id to include branch details in the transaction records.

transactions_with_branch = JOIN transactions BY branch_id, branches BY branch_id;

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.13.0 0.12.0-cdh5.13.0        cloudera        2024-08-21 22:08:03     2024-08-21 22:08:
59      HASH_JOIN

Success!
```

10. Join Transactions with Customer Information

- Join the transactions dataset with customers dataset on account_id to include customer details in the transaction records.

transactions_with_customers = JOIN transactions BY account_id, customers BY account_id;

```
HadoopVersion   PigVersion      UserId  StartedAt       FinishedAt      Features
2.6.0-cdh5.13.0 0.12.0-cdh5.13.0        cloudera        2024-08-21 22:00:52     2024-08-21 22:01:
42      HASH_JOIN

Success!
```

11. Union Two Datasets

- Assume you have a second transactions_day2.csv dataset with similar schema. Union this dataset with transactions.csv to create a unified dataset of transactions.

transactions_day2 = LOAD 'transactions_day2.csv' USING PigStorage(',') AS (transaction_id:int, account_id:int, branch_id:int, transaction_date:chararray, transaction_type:chararray, transaction_amount:float);

STORE transactions INTO 'output_transactions_day2' USING PigStorage(',');

unified_transactions = UNION transactions,

transactions_day2;

```
l - Total input paths to process : 2
(,,,transaction_amount,transaction_type,)
(51,1011,,1300,deposit,60.0)
(52,1012,,800,withdrawal,70.0)
(53,1013,,1200,deposit,80.0)
(54,1014,,1000,withdrawal,90.0)
```

```
(,,,transaction_amount,transaction_type,)
(1,1001,,1500,deposit,10.0)
(2,1002,,500,withdrawal,10.0)
(3,1003,,2000,deposit,20.0)
(4,1004,,700,deposit,30.0)
(5,1005,,300,withdrawal,20.0)
(6,1006,,1200,deposit,10.0)
```