**Project 2 Pig, Streaming & RHadoop**

**Xiang Huang, ZhongZheng Shu**

**1.** **Query 1**

This problem asks us to write a query in Pig that reports the number of transactions each customer and the total sum of these transactions. We first load in the Transactions.txt into a variable, project on fields of Customer ID and TransactionsTotal. Then we group on the Customer ID and asks pig to aggregate on it, yielding the required results.

**2. Query 2**

Query 2 requires us to write a Pig query to report Customer ID, Name, Salary, the Numer of Transactions, sum of TransTotal, and the minimum number of items in transactions, for each customers. After loading the transactions dataset, we first select the three needed columns, CustID, TransTotal, TransNumItems, and then group them by CustID and get the number of transactions, the sum of TransTotal, and the minimum numer of items for each customer. As for customers dataset, we do the similar operations to get data we need, and then join the two datasets by customer ID using Replicated Join method. Finally we remove one duplicate customer ID column in the join result, and output it as the final result.

**3. Query 3**

This problem asks us to write a pig query that reports the number of customers for every country and the maximum and minimum of TransTotal fields. We load in both datasets, projected on the required rows and store them into two variables, cust\_clean and trans\_clean respectively. Then we group on the customerID of trans\_clean, and aggregate it to get the maximum and minimum of TransTotal fields per customer. This will help in reducing the traffic and cost in the following join. Then we join the cust\_clean with the trimmed trans\_clean, and aggregate it to get the number of customers per country,and the maximum and minimum TransTotal fields per country.

**4.Query 4**

After loading two datasets and projected on them to get the required fields, we first group transactions dataset by customer ID, and then not only aggregate to get the number of transactions for each customer, but also add a new assistant column to the grouping result. All the values of this column are One. In order to get the customers who have the smallest number of transactions, we first group the result above by the new assistant column, the values of which are all One. That means we make the whole dataset as a single group so that we can calculate the minimum number of transactions. However, we lose the customer ID. So we join the first grouping result with the send grouping result by the number of transactions which is/are the minimum to get the correspinding customer ID. As for the customers dataset, after projecting on the colunms needed, we join it with the result we get from transactions dataset with by customer ID, and then generate the name of a customer and his/her minimum number of transactions.

**5.Query 5**

This problem asks us to write a streaming map-reduce job in Python. We implemented the repartition join to tackle this proble In the mapper function, we used a helper function to tell if the streaming file belongs to customers.txt or transactions.txt. Then we output the Customer ID, Flag ‘1’ and string ‘1’ if this record belongs to transactions, and filters out those records that do not have countrycode equal to five and output the Customer ID, Flag ‘0’ and name. The reason why we choose to use Flag ‘1’ for transactions is that we want the record from the customers to show first on the reducer. The output values are concatenated by ‘.’, and we specified that only the first two is the key and the partitioner will only use the joining key for partitioning. Then we implemented the reducer function, which keeps track of count of transactions for each customer ID and outputs the desired CustomerID, name and count result. The configuration code for running this Streaming job is included in the Hadoop-Streaming-Config.txt.

6. Query 6

To comlete this question, we need to write a RHadoop script to finish the grouping job on customers dataset by CountryCode, and take advantage of R to plot Customers Count by CountryCode and Sorted Costomers Count by CountryCode. In terms of Map function, we get all country codes iteratively by each line, and output each code as key, and One as value. In the process of reducing, we just calculate the sum of counts for each key (CountryCode), and the output the result. The two plots are as below.



