**Core Java**

* Files: DECaseStudy.zip
* Description: The zip file DECaseStudy.zip is the eclipse project that includes the JDBC library jar file in place. The main class is the DECaseStudy.java class and can take arguments of connection url, username, and password. The default values for user and password are “root” and “mysql” and should be changed for local MySQL configuration. When running, the first message will let you know if the connection with MySQL was successful, then prompt you to enter 1-3 for transaction details, customer details, and exiting the application.

**RDBMS/mySQL Description**:

* Files: CDW\_SAPP.sql
* Description: This MySQL script was provided and is included. This file was used as the basis and testing for the project.

**Hadoop/HDFS/Dataware Housing**

* Files: hadoop.sh
* Description: This shell script included in hadoop.sh will automate the entire Sqoop process. ***ALL SHELL FILES IN THIS PROJECT MUST HAVE “755” PERMISSIONS.*** The script will first create the credit card system directory if it exists, then it will delete all fields in the credit card system directory if it exists. The script then performs 4 Sqoop imports for the 4 customer tables, and finally move files and rename them in the root of the Credit\_Card\_System directory.

**Hive and Partition**

* Files: hive.hql
* Description: The hive script is included in the hive.hql file. This can be run directly using the command “hive –f hive.hql” on the command line. This script will first drop all 4 tables relating to the case study, temporary and partitioned from hive if it exists, and then create the temporary and partitioned tables. The script then loads data from the sqoop text files created from the sqoop job. Then the script will insert all data from the temporary table into the final partitioned tables, and the partitions will be created dynamically.

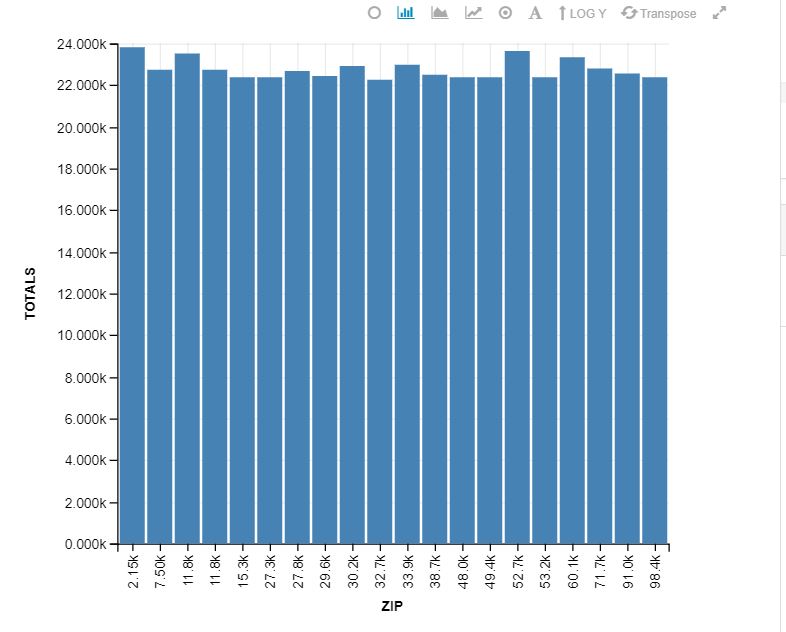
**Oozie (Sqoop and Hive)**

* Files: coordinator.xml, java-json.jar, job.properties, metastore.sh, setup.sh, hive.hql, workflow.xml
* Description: The workflow and coordinator definition files are included in the workflow.xml and coordinator.xml files. The job properties are included in the job.properties file configured for the coordinator. The java-json.jar file is a library extension required for the oozie jobs to function. The metastore.sh script will add all sqoop jobs into the metastore. The hive.hql is the hive script to load data into the temporary tables and then into the partitioned tables. The setup.sh will automate the entire process, create the case\_study folder in the maria\_dev user folder in HDFS, copy the workflow and coordinator files to HDFS, copy the hive.hql script to HDFS, create the lib folder and copy the java-json.jar file to HDFS, run the metastore script to add sqoop jobs to the metastore, and finally run the coordinator job in oozie. The workflow will first delete the Credit\_Card\_System directory if it exists and create a new directory, then runs 4 sqoop jobs from the metastore, then the hive script to insert the oozie dump into the partitioned tables.

**Oozie (Sqoop and Hive optimized)**

* Files: coordinator\_2.xml, insert.hql, job.properties, metastore.sh, setup.hql, setup.sh, workflow\_2.xml
* Description: The workflow and coordinator definition file are included in workflow\_2.xml and coordinator.xml files. The job properties are included in the job.properties file configured for the coordinator. The setup.hql file will first drop any temporary tables if there are any, and will create them for the incremental sqoop load. The insert.hql will append the new data from the temporary tables to the existing partitioned tables. The metastore.sh will create the sqoop jobs in the metastore. The setup.sh will automate the entire process; will copy the hive script files, workflow, and coordinator files to the case\_study folder, run the metastore script to place the sqoop jobs in the metastore, and finally run the oozie coordinator job.

**Visualization**

* Files: Capture1.JPG, Capture2.JPG, visualization.txt
* Description: The queries 1 and 2 are included in the visualization.txt file. The included .jpg files are the Hive charts that were created from the queries, they are also shown below. For some reason, the zip codes are taken as numbers and not strings, therefore are hard to individually identify.
* 
* 