**By Rebecca Amarh-Adjei Data Engineering DE03 NY September 2018**

**Case Study: Credit Card Management System**

**2.1.1 Core Java**

Functional requirements for the 2.1.1 Transaction Details Module and 2.1.2 Customer Details Module

**Core Java File List**

* casestudy\_de\_start.zip
* casestudy\_de\_start/src contains all java code listed below:
* **casestudy\_de\_start/src dao/**

/CustomerDao.java

/dbconnection\_abstract.java

/TransactionDao.java

* **casestudy\_de\_start/src/exceptions**

/MainException.java

/NoZeroException.java

* **casestudy\_de\_start/src/model**

/Customer.java

/Transaction.java

* **casestudy\_de\_start/src/resources**

/MyQuries.java

* **casestudy\_de\_start/src/runner**

/Customer\_runnable.java

/Main.java

/Transaction\_runnable.java

Run casestudy\_de\_start runners in eclipse or any appropriate IDE.

casestudy\_de\_start runners will launch a menu on the console

**2.1.1 RDBMS/mySQL Description**:

Relational Database for Credit Card Management System

**RDBMS/MySQL File List**

* CDW\_SAPP.sql
* In mysql create database CDW\_SAPP and use CDW\_SAPP before running CDW\_SAPP.sql script.

**2.2.1 Hadoop/hdfs/dataware housing**

Functional requirements for 2.2.1 Data Extraction and Transportation with sqoop. Utilized Sqoop to extract data from mysql

**Hadoop/hdfs/dataware housing file list**

* **2.2.1\_SQOOP\_DataExtractionAndTransportationModule folder**

\CDW\_Case\_Study\_Req\_2.2.1.txt

Contains commands for Sqoop import to extract data from mysql to hdfs for the various modules namely:

CDW\_SAPP\_D\_BRANCH (--target-dir /user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_D\_BRANCH)

CDW\_SAPP\_F\_CREDIT\_CARD (--target-dir/user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_F\_CREDIT\_CARD)

CDW\_SAPP\_D\_CUSTOMER (--target-dir /user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_D\_CUSTOMER)

CDW\_SAPP\_D\_TIME (--target-dir /user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_D\_TIME)

\CDW\_Case\_Study\_Req\_2.2.1\_Queries.txt

Contain queries to run on the mysql command line

**2.2.2 Hive and Partition**

Functional requirements for the 2.2.2 Oozie Loading Module (Data Loading with Hive). Utilize Hive to create tables in the Hadoop Filesystem and then load the data extracted via Sqoop into those tables.

Map to transform the data based on requirements found in the Mapping Document

**Hive and Partition File List**

* **2.2.2\_HIVE\_DataLoadingModule folder**

\CDW\_Case\_Study\_Req\_2.2.2 Hive - Partitioned Tables.txt

Contains HIVE query commands for following HIVE partition tables: CDW\_SAPP\_D\_BRANCH\_P; CDW\_SAPP\_D\_CUSTOMER\_P; CDW\_SAPP\_F\_CREDIT\_CARD\_P; CDW\_SAPP\_D\_TIME\_P

\CDW\_Case\_Study\_Req\_2.2.2-External\_HIVE\_Tbls.txt

Contains HIVE query commands for following External HIVE tables: CDW\_SAPP\_D\_BRANCH; CDW\_SAPP\_F\_CREDIT\_CARD; CDW\_SAPP\_D\_CUSTOMER; CDW\_SAPP\_D\_TIME

.

**2.2.3 Oozie (Sqoop and Hive)**

Functional Requirements for the 2.2.3 Process Automation Module (Automating the Process with OOzie). An Oozie workflow is created using the processes from 2.2.1 and 2.2.2. The workflow is then incorporated into an Oozie coordinator for scheduled job frequency executions.

**Oozie (Sqoop and Hive) File List**

* **2.2.3 OozieHIVE\_ProcessAutomationModule folder**

CDW\_CS 2.2.3 sqoop jobs.txt

branch.hive (LOCATION "/user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_D\_BRANCH/";)

ccs1.properties (oozie.wf.application.path=${nameNode}/user/maria\_dev/Credit\_Card\_System/workflow1.xml)

ccs2.properties (oozie.coord.application.path=${nameNode}/user/maria\_dev/Credit\_Card\_System/coordinator1.xml)

coordinator1.xml (<app-path>${nameNode}/user/maria\_dev/Credit\_Card\_System/workflow1.xml</app-path>)

creditcard.hive (LOCATION "/user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_F\_CREDIT\_CARD/";)

customer.hive (LOCATION "/user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_D\_CUSTOMER/";)

time.hive (LOCATION "/user/maria\_dev/Credit\_Card\_System/CDW\_SAPP\_D\_TIME/";)

\workflow1.xml (<app-path>${nameNode}/user/maria\_dev/Credit\_Card\_System/workflow1.xml</app-path>)

To run the workflow1.xml only, use job properties called ccs1

To run the coordinator1.xml use the job properties called ccs2

**2.2.4 Oozie (Sqoop and Hive optimized)**

Functional requirements for the 2.2.4 Process Optimization Module (Optimizing Process) Oozie workflow similar to 2.2.3 except only new data is imported

**Oozie (Sqoop and Hive optimized) File List**

* 2.2.4 OozieProcessOptimizationModule Folder

ccs3.properties (oozie.wf.application.path=${nameNode}/user/maria\_dev/Credit\_Card\_System/workflow2.xml)

ccs4.properties (oozie.coord.application.path=${nameNode}/user/maria\_dev/Credit\_Card\_System/coordinator2.xml)

CDW\_CS 2.2.4 sqoop Optimize the process.txt

coordinator2.xml (<app-path>${nameNode}/user/maria\_dev/Credit\_Card\_System/workflow2.xml</app-path>)

incrBranch\_P.hive

incrCreditCard\_P.hive

incrCustomer\_P.hive

incrTime\_P.hive

workflow2.hive (<command>job --meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop --exec incrCreditCardJob</command>)

To run workflow2.xml only the job properties file called ccs3.properties

To run the coordinator2.xml, use the job properties file called ccs4.

**- Steps - jobs**

1) Update SQL at ultra

2) delete existing jobs as they have been updated: [root@sandbox ~]# sqoop job --meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop --delete incrBankJob

3) Run Sqoop job

4) Check list of jobs = [root@sandbox ~]# sqoop job --meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop --list

5) Exec job eg: [root@sandbox ~]# sqoop job --meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop --exec incrBranchJob

6) Check Ambari to ensure job is there

7) Run the job:

8) // do cd Documents, then cd CreditCardSystem

Run Oozie Workflow2 [root@sandbox CreditCardSystem]# oozie job -oozie http://localhost:11000/oozie -config ccs3.properties -run

9) Run Oozie Coordinator2 [root@sandbox CreditCardSystem]# oozie job -oozie http://localhost:11000/oozie -config ccs4.properties -run

**2.2.5 Visualization**

Functional requirements for the 2.2.5 Data Visualization (Visualization of Dataset) Hive query and Hive visualization tool were used.

* **2.2.5 Data Visualization folder**

2.2.5 Visualization Hive Queries

2.2.5 DataVisualization.docx contains the hive queries with their corresponding hive visual snippets