Credit Card Management System

Case study for Data engineering

Jobayar Alom 2018-11-PL-DE03

**Proposed Steps:**

1: Create modules/ Business logic on java using eclipse. User can only use console for input and output because Servlets are not under the scope of the course

2: Use RDBMS MySQL for Database. Initially, Data will be stored in MySQL. We will use Sqoop for pullout (import) the data from RDBMS into HADOOP or HIVE. Then will use Oozie for the schedule the Jobs

**Core Java OOP**

Here is Java program using Eclipse allow users to display information based on user input:

* **Transaction Details Module**

Transaction Details - This Module will be used by bank admin to enter the bank details received upon a request approval from admin. The system should then update the Customer details in the system along for the corresponding Customer

* Display the transactions made by customers living in a given zip code for a given month and year. Order by day in descending order.
* Display the number and total values of transactions for a given type.
* Display the number and total values of transactions for branches in a given state
* **Customer Details Module**

This Module will be used by the Bank admin to Access Customer Details and change the existing Customer details into the system. The bank admin should have the details of the Customers to be entered into the system.

* Check the existing account details of a customer
* Update the customer’s account.
* Generate monthly statement.
* Search for transactions in between a date period.
* **Java Program Description**

Java Project is called “**CDW\_SAPP**”, and contains 4 Packages into Source Folder:

**Dao:** Established the connection between Java and MySQL using JDBC

**Model**: the format of how the data is processed from the tables

**Resources**: queries.java: it contains all the SQL statements

**Runne**r: which is for the main user menu to run the project.

* **RDBMS / MySQL**

**Inside the folder there is a [CDW\_SAPP.sql] file to create database with all information.**

**Hadoop/HDFS/Data warehousing**

**SQOOP\_DataExtraction\_Transportation**

I transform all the data based on requirements found in the Mapping Document prior to loading the data into Hadoop

**1. Branch data into CDW\_SAPP\_D\_BRANCH\_Table\_Oozie**

**2. Credit Card Data into CDW\_SAPP\_F\_CREDIT\_CARD\_Oozie**

**3. Time data into CDW\_SAPP\_Time\_Id\_Oozie**

**4. Customer Data into CDW\_SAPP\_D\_CUSTOMER\_Oozie**

**HIVE\_Data\_Loading**

This folder contain a file name: create\_hive\_tables. It has all the command to create hive table in HDFS directory.

**1. CDW\_SAPP\_D\_BRANCH\_oozie to display Branch Data.**

**2. CDW\_SAPP\_F\_CREDIT\_CARD\_Oozie to display Credit Card Data.**

**3. CDW\_SAPP\_TIMEID\_Oozie to Display (Created new field) TimeId.**

**4. CDW\_SAPP\_D\_CUSTOMER\_Oozie to display Customer Data.**

[NB: All those 4 table mention above has 4 partition table based on specific categories.]

* **Oozie Process Automation Module**

(Sqoop and Hive)

In the Oozie\_Automation folder you have coordinatorjob.properties to run coordinator.xml file.

Then this coordinator.xml file will execute workflow.xml based on assign time, date and frequency.

Workflow.xml:

Workflow is an xml file contains action. It execute those actions in order you assign.

Workflow.xml will do following task.

* Extract all the data from RDBMS
* Drop Hive Tables if exist.
* Create Hive Tables.
* Create Partition for hive table to execute query fast.
* **Oozie\_Process Optimization Module**

Create a new Oozie workflow similar to **Oozie Process Automation Module**. This time, Sqoop will only import new data from RDBMS.

In the Oozie\_Optimization folder you incrementcoordjob.properties to run incrementcoordinator.xml file.

Then this incrementcoordinator.xml file will execute incrementcoordinatorflow .xml based on assign time, date and frequency.

incrementcoordinatorflow:

incrementcoordinatorflow is an xml file contains action. It execute those actions in order you assign.

Incrementcoordinatorflow.xml will do following task.

* Extract [**ONLY NEW or UPDATED**] data from RDBMS
* Load Data to hdfs
* It does not effect on original workflow.
* **Visualization**

**There are two queries requested in this section**:

To list the top 20 branch zip code based on the total of the transaction values.

To list the total transaction values based on the four quarters in 2018, and transaction types.

In the folder “2.2.5\_Data\_Visualization” file contains the 2 HQL queries for the above requirements.