2. Systems Requirements

By Rafael Perez Fabian

2.1. Functional Requirements - Operational Database

2.1.1. Transaction Details Module

2.1.1.1. Using Windows

a. Download the submitted file Final_Case_RPF_Java.zip from Canvas platform to your desired location

b. Open the Eclipse application. Once there, go to File -> Import -> Existing Projects into Workspace and press Next

c. Select archive file and import the downloaded file Final_Case_RPF_Java.zip

d. Press Finish

e. Run the Main class, and enjoy!!!!

2.1.1.2. Using Linux

a. Download the submitted file *Final_Case_RPF_Java.zip* from Canvas Platform to your desired location

b. Open a new **WinSCP** session using the values:

Host name: 192.168.109.130
Port number: 2222
User name: root
Password: password

c. In your Windows machine go to the location where you stored the *Final_Case_RPF_Java.zip* file, and move to the location **/root/Documents** in your VMWare Linux system

d. Open a **UltraVNC Viewer** session and log in VNC Server 192.168.109.130:1 (use password= password)

e. Open the **Eclipse** application. Once there, go to *File -> Import -> Existing Projects into Workspace* and press Next

f. Select archive file option and import the downloaded file *Final_Case_RPF_Java.zip* from */root/Documents*

g. Press Finish

h. Run the Main class, and enjoy!!!!

2.1.1.3. (RDBMS/MySQL)

2.1.1.3.1. Using Windows

a. Download the submitted file *CDW_SAPP_LAST_UPDATED2.sql* from Canvas platform to your desired location.

b. Open the MySQL Workbench application and log into the server by clicking Local Instance MySQL 5.7. (use password=password)

c. Once there, go to *File -> Open SQL Script* and select the downloaded file *CDW_SAPP_LAST_UPDATED2.sql*

d. Run the above SQL script, and enjoy!!!!

2.1.1.3.2. Using Windows

a. Download the submitted file *CDW_SAPP_LAST_UPDATED2.sql* from Canvas Platform to your desired location

b. Open a new **WinSCP** session following the authentication process described in the section *2.1.1.2* of this document

c. In your Windows machine, go to the location where you stored the *CDW_SAPP_LAST_UPDATED2.sql* file and move to the location */root/Documents* in your VMWare Linux system

d. Open a new **UltraVNC Viewer** session and log in VNC Server 192.168.109.130:1 (use password= password)

e. Open the **MySQL Workbench** application and log into the server by clicking *Local Instance 3306* (use password=password)

f. Once there, go to *File -> Open SQL Script* and select the file *CDW_SAPP_LAST_UPDATED2.sql* from */root/Documents*

g. Run the above SQL script, and enjoy!!!!

2.2. Functional Requirements - ETL of Data

Prerequisites

*** Follow these directions before executing the sections below!!!

a. Download the submitted file *Final_Case_RPF.zip* from Canvas Platform to your desired location

b. Open a new **WinSCP** session following the authentication process described in the section 2.1.1.2 of this document

c. In your Windows machine go to the location where you stored the *Final_Case_RPF* file, and move to the location **/root** in your VMWare Linux system

d. Open a new browser window and log into the sandbox server *Your_Server_IP_Address:4200 (i.e. 192.168.109.130:4200)* using the values below:

User: root
Password: password

You will be prompted at */root*
Note: Type *pwd* command to check that you're at */root*. In case you're in another location type the command *cd ~*

e. In case you submitted many times the zip file in Canvas Platform, rename the zip file typing:

mv Final_Case_RPF.zip Final_Case_RPF.zip*

f. Unzip the *Final_Case_RPF.zip* file:

unzip Final_Case_RPF.zip

g. Make executable all the necessary scripts by typing the command:

find ./Final_Case_RPF -name ".sh" -exec chmod 744 {} +*

2.2.1. Data Extraction and Transformation Module - Data Extraction and Transportation with Sqoop

a. Under **/root** directory, import the tables from MySQL *cdw_sapp* database to your HDFS system by executing the following script:

./Final_Case_RPF/2.2.1_DEATM/sqoop/sq_import_cdw_sapp_tables.sh

2.2.2. Data Loading Module - Data Loading with Hive

a. Under **/root** directory, execute the below scripts to:

i. Create Hive database *cdw_sapp* by executing the Hive script:

hive -f ./Final_Case_RPF/2.2.2_DLM/hive/cr_database.hql

ii. Create Hive external non-partitioned tables by executing the Hive script:

hive -f ./Final_Case_RPF/2.2.2_DLM/hive/cr_ext_tables.hql

iii. Create Hive internal partitioned tables by executing the Hive script:

hive -f ./Final_Case_RPF/2.2.2_DLM/hive/cr_int_tables.hql

iv. Load the data from external tables into the warehouse by executing the Hive script:

hive -f ./Final_Case_RPF/2.2.2_DLM/hive/ins_wh_tables.hql

2.2.3. Process Automation Module

2.2.3.1. Prerequisites

In order to prevent data redundancy, we should have to recreate the existent database, *cdw_sapp*, in our HDFS system. Since both modules, Process Automation Module, and Process Optimization Module will use the same database, the necessary scripts Hive/bash scripts for re-creating the database are located in a shared folder (***./Final_Case_RPF/shared***). Please follow the below instructions **before** executing the section ***2.2.3.2 Automating the Process with Oozie***

a. Under **/root** directory, execute the following scripts:

i. Drop/create the *cdw_sapp* database:

./Final_Case_RPF/shared/cr_database.sh

ii. Create the Hive external non-partitioned tables:

./Final_Case_RPF/shared/cr_ext_tables.sh

iii. Create the Hive internal partitioned tables:

./Final_Case_RPF/shared/cr_int_tables.sh

b. We also need to create the necessary HDFS file structure to execute our automatized process through Oozie. To do that, under **/root** execute the script below:

./Final_Case_RPF/2.2.3_PAM/shared/prepare_env.sh

This script will create the following file structure in our HDFS system:

/Credit_Card_System
/unoptimized
/hive
ins_branch.hql
ins_credit_card.hql
ins_customer.hql
ins_time.hql
/oozie
wf_coord_unoptimized.xml
wf_unoptimized.xml

2.2.3.2. Automating the Process with Oozie

a. Execute the Oozie coordinator workflow to automatize the data loading from MySQL *cdw_sapp* database to our WareHouse by using Sqoop import statements. Under **/root** directory type the following command:

oozie job -oozie http://localhost:11000/oozie -config ./Final_Case_RPF/2.2.3_PAM/job.properties -run

b. Monitor the Oozie workflow execution through the Oozie Web Console. Open a new browser window and log in by entering *Your_Server_IP_Address:11000/oozie (i.e. 192.168.109.130:11000/oozie)*

2.2.4. Process Optimization Module

2.2.4.1. Prerequisites

a. Follow the steps a.i (create *cdw_sapp* database), a.ii (create Hive external tables), a.iii (create Hive internal tables) from the section *2.2.3.1 Prerequisites* to recreate the *cdw_sapp* database in our HDFS system.

b. The optimized process uses Sqoop jobs stored in the meta-store Server. This is the way to Sqoop to have control over the last updated and new data imported from MySQL database (last-modified option). In order to execute these jobs follow the instructions below **before** executing the section **2.2.4.2 Optimizing the process.**

i. Open a new browser window and log into the sandbox server *Your_Server_IP_Address:4200 (i.e. 192.168.109.130:4200)* using the values below:

User: root
Password: password

ii. In your new session, start the Sqoop metastore server by typing the command:

sqoop-metastore

iii. To prevent Sqoop jobs malfunction first, we will delete, if they exist, Sqoop jobs previously created and executed in our system. In your first sandbox server session, under **/root** directory execute the script:

./Final_Case_RPF/2.2.4_POM/shared/del_jobs_opt.sh

iv. Create the Sqoop to import data from MySQL *cdw_sapp* database to HDFS system by using import last-modified. Under **/root** directory, execute the following script:

./Final_Case_RPF/2.2.4_POM/shared/cr_sqoop_jobs_opt.sh

c. Finally, create the necessary HDFS file structure to execute our automatized process through Oozie. To do that, under **/root** execute the script below:

./Final_Case_RPF/2.2.4_POM/shared/prepare_env.sh

This script will create the following file structure in our HDFS system:

/Credit_Card_System
/optimized
/hive
ins_branch.hql
ins_credit_card.hql
ins_customer.hql
ins_time.hql
/oozie
wf_coord_optimized.xml
wf_optimized.xml

Additionally, this script will copy the java-json.jar file necessary to execute Sqoop jobs through Oozie

2.2.4.2. Optimizing the process

a. Execute the Oozie coordinator workflow to automatize the data loading from MySQL *cdw_sapp* database to our WareHouse by using Sqoop import last-modified jobs. Under **/root** directory type the following command:

oozie job -oozie http://localhost:11000/oozie -config ./Final_Case_RPF/2.2.4_POM/job.properties -run

b. Monitor the Oozie workflow execution through the Oozie Web Console. Open a new browser window and log in by entering *Your_Server_IP_Address:11000/oozie (i.e. 192.168.109.130:11000/oozie)*

2.2.5. Data Visualization - Visualization of Dataset

a. Our data finally is available in our Warehouse, you're ready to start your Business Intelligence job. In a new browser go to your Ambari Sandbox by typing *Your_Server_IP_Address:8080 (i.e. 192.168.109.130:8080)* using the values below:

   Username: maria_dev

   Password: maria_dev

b. Select from the menu Hive View. Once there, go to *Database Explorer* and select the database *cdw_sapp*

c. Type the following queries to obtain:

   i. Top 20 zip codes by total transaction value:

```
SELECT
b.branch_zip, SUM(cc.transaction_value) AS tv
FROM cdw_sapp_f_credit_card cc JOIN cdw_sapp_d_branch b
ON cc.branch_code=b.branch_code
GROUP BY b.branch_zip
ORDER BY tv DESC
LIMIT 20;
```

   ii. The total transaction value for each transaction type by quarter in 2018:

```
SELECT cc.transaction_type, t.quarter,
SUM(cc.transaction_value)
FROM cdw_sapp_f_credit_card cc JOIN cdw_sapp_d_time t
ON cc.transaction_id=t.transaction_id
WHERE t.year=2018
GROUP BY cc.transaction_type, t.quarter;
```

d. Although you can use the Hive Visualization tool to make graphs about the data in the *cdw_sapp* for this project I decided to connect Tableau with the Hive Sandbox system. Tableaus is a more flexible, user-friendly and probably, the most popular data visualization tool for Business Intelligence. Attached in this project you can find the obtained graphs mapping the result of the queries above.