2.2.1 Data Extraction and Transportation Module

2.2.1.1 Data Extraction and Transportation with Sqoop Credit Card System Req-2.2.1 Data Extraction and Transportation with Sqoop

Functional Requirements

Utilize Sqoop to extract the following data according to the specifications found in the mapping document: 1. Branch data into CDW\_SAPP\_BRANCH 2. Credit Card Data into CDW\_SAPP\_CREDITCARD 3. Time data into CDW \_SAPP\_TIME 4. Customer Data into CDW \_SAPP\_CUSTOMER

Notes:  Data Engineers will be required to transform the data based on requirements found in the Mapping Document prior to loading the data into Hadoop.

 TIMEID is a field that the Data Engineers should create based on the DAY, MONTH, and TIME fields located in the CUSTOMER table. Format should be YYYYMMDD. For instance, January 4th, 2017 would become 20170104

 Data Engineers should extract the above data to the /Credit\_Card\_System/ folder in the Hadoop Filesystem

First step:use sqoop import data

1. very fast command: sqoop import every table into HDFS, create only one file.

sqoop import-all-tables --connect jdbc:mysql://localhost/CDW\_SAPP --drive

r com.mysql.jdbc.Driver --warehouse-dir /user/maria\_dev/Credit\_card\_System -m 1

2. sqoop import table into HDFS and participate it into 4 files.

sqoop import-all-tables --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.

mysql.jdbc.Driver --direct -m1

3.direct import into hive database

sqoop import-all-tables --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.

mysql.jdbc.Driver -m1 --hive-database Credit\_card\_System --hive-import

second step: create table time

1. METHOD: In MYSQL OR in hive, use followd command to create time table and insert data

create table CDW\_SAPP\_TIME(ID INteger, day varchar(2), month varchar(2), year varchar(4));

insert into CDW\_SAPP\_TIME(id,day,month,year) select concat(YEAR,lpad(MONTH,2,'0'),lpad(DAY,2,'0')) AS ID, DAY, MONTH, YEAR FROM CDW\_SAPP\_CREDITCARD;

1. Method: use sqoop import column data: day, month, year to hive, create new table
2. sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --query 'SELECT concat(YEAR,lpad(MONTH,2,0),lpad(DAY,2,0)) AS ID, DAY, MONTH,quarter(concat(year,lpad(
3. MONTH,2,0),lpad(day,2,0))) as QUARTER, YEAR FROM CDW\_SAPP\_CREDITCARD WHERE $CONDITIONS' --target-dir /user/maria\_dev/CREDIT\_CARD\_SYSTEM --fields-terminated-by '\t' --split-by ID -m1 --hive-import --hive-table cr
4. edit\_card\_system.CDW\_SAPP\_TIME

3. Method: use sqoop query command import data TO HDFS

sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --query 'select concat(YEAR,lpad(MONTH,2,'0'),lpad(DAY,2,'0')) AS ID,DAY,MONTH,quarter(concat(YEAR,lpa

d(MONTH,2,0),lpad(DAY,2,0))) as QUARTER,YEAR from CDW\_SAPP\_CREDITCARD WHERE $CONDITIONS' --target-dir /user/maria\_dev/CREDIT\_CARD\_SYSTEM --fields-terminated-by '\t' --split-by ID -m1

2.2.2 Data Loading with Hive

Credit Card System

Req-2.2.2

Data Loading with Hive

Functional Requirements Utilize Hive to create tables in the Hadoop Filesystem and then load the data extracted via Sqoop into those tables.

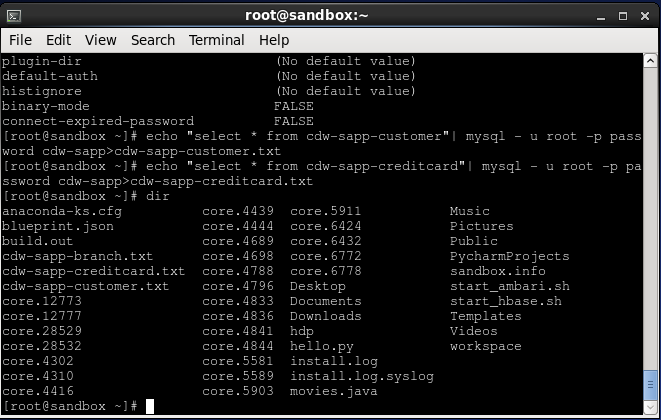
Data Engineers will be map to transform the data based on requirements found in the Mapping Document.

SELECT \* from cdw\_sapp\_customer INTO OUTFILE 'C:/tmp/cdw\_sapp\_customer.csv'

FIELDS ENCLOSED BY '"' TERMINATED BY ';' ESCAPED BY '"'

LINES TERMINATED BY '\r\n';

Method 1: in linux command line:



[maria\_dev@sandbox ~]$ hadoop fs -put cdw\_sapp\_branch.txt /user/maria\_dev/Credit\_card\_System/

[maria\_dev@sandbox ~]$ hadoop fs -put cdw\_sapp\_customer.txt /user/maria\_dev/Credit\_card\_System/

[maria\_dev@sandbox ~]$ hadoop fs -put cdw\_sapp\_creditcard.txt /user/maria\_dev/Credit\_card\_System/

[[maria\_dev@sandbox ~]$ hive --hiveconf hive.root.logger=warn,console

Logging initialized using configuration in file:/etc/hive/2.5.0.0-1245/0/hive-log4j.properties

hive> use Credit\_card\_System;

OK

Time taken: 3.044 seconds

hive> show databases;

OK

credit\_card\_system

default

foodmart

movielens

xademo

Time taken: 0.381 seconds, Fetched: 5 row(s)

hive> show tables;

OK

branch

temp\_branch

temp\_creditcard

temp\_customer

Time taken: 0.569 seconds, Fetched: 4 row(s)

hive> create table customer (firstname string, middlename string, lastname string, SSN string, credit\_card\_no string, APT\_no string, street\_name string, cust\_city string, cust\_state string, cust\_country string,

cust\_zip string, cust\_phone string, cust\_email string, last\_updated string) row format delimited fields terminated by ',' stored as textfile Tblproperties("skip.header.line.count"="3");

OK

Time taken: 0.744 seconds

hive> LOAD DATA INPATH '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_CUSTOMER.csv' OVERWRITE INTO TABLE customer;

Loading data to table credit\_card\_system.customer

18/08/22 15:49:57 [main]: ERROR hdfs.KeyProviderCache: Could not find uri with key [dfs.encryption.key.provider.uri] to create a keyProvider !!

Table credit\_card\_system.customer stats: [numFiles=1, numRows=0, totalSize=162470, rawDataSize=0]

OK

Time taken: 2.104 seconds

Second Method: use Hadoop hive

use CDW\_SAP

select \* from CDW\_SAPP\_CUSTOMER;

ALTER TABLE CDW\_SAPP\_CUSTOMER ADD last\_updated TIMESTAMP DEFAULT CURRENT\_TIMESTAMP;

sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --table CDW\_SAPP\_CUSTOMER -m1 --target-dir /user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_CUSTOMER/ --inc

remental lastmodified --check-column last\_updated --last-value 1 --append;

DROP TABLE CUSTOMER;

create table customer

( CUST\_F\_NAME VARCHAR(40), CUST\_M\_NAME VARCHAR(40), CUST\_L\_NAME VARCHAR(40),

CUST\_SSN INT,CUST\_CC\_NO STRING,CUST\_APT\_NO STRING, CUSTO\_STREET VARCHAR(38),

CUST\_CITY VARCHAR(30), CUST\_STATE VARCHAR(30), CUST\_COUNTRY VARCHAR(30),

CUST\_ZIP INT, CUST\_PHONE VARCHAR(8), CUST\_EMAIL VARCHAR(40), LAST\_UPDATED TIMESTAMP )

row format delimited fields terminated by ','

STORED AS TEXTFILE

Tblproperties("skip.header.line.count"="3");

LOAD DATA INPATH '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_CUSTOMER/part-m-00000'

OVERWRITE INTO TABLE customer;

drop table CDW\_SAPP\_D\_CUSTOMER;

create external table CDW\_SAPP\_D\_CUSTOMER

( CUST\_F\_NAME VARCHAR(40), CUST\_M\_NAME VARCHAR(40), CUST\_L\_NAME VARCHAR(40),

CUST\_SSN INT,CUST\_CC\_NO STRING, CUSTO\_STREET VARCHAR(38), CUST\_CITY VARCHAR(30),

CUST\_COUNTRY VARCHAR(30), CUST\_ZIP INT, CUST\_PHONE VARCHAR(8),

CUST\_EMAIL VARCHAR(40), LAST\_UPDATED TIMESTAMP )

PARTITIONED BY (CUST\_STATE VARCHAR(30))

row format delimited fields terminated by ','

LINES TERMINATED BY '\n'

location '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_CUSTOMER/';

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE CDW\_SAPP\_D\_CUSTOMER

PARTITION (cust\_state)

SELECT CONCAT(UCASE(SUBSTRING(cust\_f\_name, 1, 1)),LOWER(SUBSTRING(cust\_f\_name, 2))),

lcase(CUST\_M\_NAME),

CONCAT(UCASE(SUBSTRING(cust\_l\_name, 1, 1)),LOWER(SUBSTRING(cust\_l\_name, 2))),

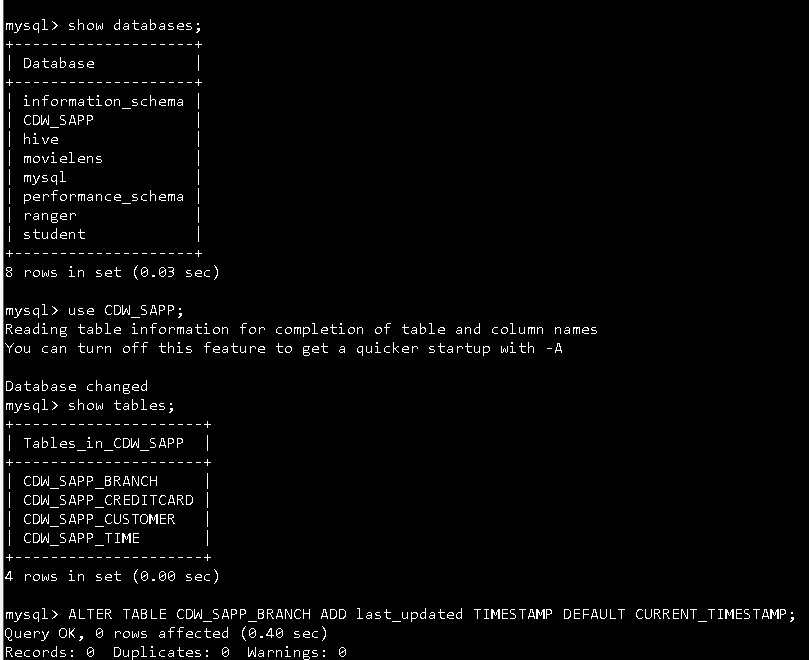
cust\_cc\_no, concat(cust\_apt\_no," ",custo\_street) as cust\_street, cust\_city,

cust\_country,cust\_zip, cust\_ssn

concat(SUBSTRING(cust\_phone, 1, 3),'-',SUBSTRING(cust\_phone, 4, 4)),

cust\_email, last\_updated, cust\_state

FROM CUSTOMER;



sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --table CDW\_SAPP\_BRANCH -m1 --target-dir /user/maria\_dev/Credit\_card\_System/BRANCH;

drop table cdw\_branch;

create table cdw\_branch

( BRANCH\_CODE INT,BRANCH\_NAME VARCHAR(25), BRANCH\_STREET VARCHAR(30),

BRANCH\_CITY VARCHAR(30), BRANCH\_STATE VARCHAR(30), BRANCH\_ZIP INT,

BRANCH\_PHONE VARCHAR(13), LAST\_UPDATED TIMESTAMP )

row format delimited fields terminated by ','

STORED AS TEXTFILE

Tblproperties("skip.header.line.count"="3");

LOAD DATA INPATH '/user/maria\_dev/Credit\_card\_System/BRANCH/part-m-00000'

OVERWRITE INTO TABLE cdw\_branch;

drop table CDW\_SAPP\_D\_BRANCH;

create external table CDW\_SAPP\_D\_BRANCH

(BRANCH\_CODE INT,BRANCH\_NAME VARCHAR(25), BRANCH\_STREET VARCHAR(30),

BRANCH\_CITY VARCHAR(30), BRANCH\_STATE VARCHAR(30), BRANCH\_ZIP INT,

BRANCH\_PHONE VARCHAR(13))

PARTITIONED BY (LAST\_UPDATED TIMESTAMP)

row format delimited fields terminated by ','

LINES TERMINATED BY '\n'

location '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_BRANCH/';

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE CDW\_SAPP\_D\_BRANCH

PARTITION(LAST\_UPDATED)

SELECT BRANCH\_CODE,BRANCH\_NAME, BRANCH\_STREET, BRANCH\_CITY, BRANCH\_STATE,

COALESCE(BRANCH\_ZIP,99999),concat('(',substring(branch\_phone,1,3),')',

substring(branch\_phone,4,3),'-',substring(branch\_phone,7,4)),

last\_updated

FROM cdw\_branch;

sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --query 'select concat(YE

AR,lpad(MONTH,2,'0'),lpad(DAY,2,'0')) AS ID,DAY,MONTH,quarter(concat(YEAR,lpad(MONTH,2,0),lpad(DAY,2,0))) as QUARTER,YEAR from CDW\_SAP

P\_CREDITCARD WHERE $CONDITIONS' --target-dir /user/maria\_dev/Credit\_card\_System/time --fields-terminated-by '\t' --split-by ID -m1

drop table TIME;

create table TIME

( TIMEID VARCHAR(8), DAY INT, MONTH INT,

QUARTER VARCHAR(8), YEAR INT)

row format delimited fields terminated by '\t'

STORED AS TEXTFILE

Tblproperties("skip.header.line.count"="3");

LOAD DATA INPATH '/user/maria\_dev/Credit\_card\_System/time'

OVERWRITE INTO TABLE TIME;

drop table CDW\_SAPP\_D\_TIME;

create external table CDW\_SAPP\_D\_TIME

( TIMEID VARCHAR(8), DAY INT, MONTH INT,

YEAR INT)

PARTITIONED BY (QUARTER VARCHAR(8))

row format delimited fields terminated by ','

LINES TERMINATED BY '\n'

location '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_D\_TIME/';

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE CDW\_SAPP\_D\_TIME

PARTITION (QUARTER)

SELECT TIMEID, DAY, MONTH,

YEAR, QUARTER

FROM TIME;

sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --query 'select TRANSACTI

ON\_ID,CREDIT\_CARD\_NO, concat(YEAR,lpad(MONTH,2,'0'),lpad(DAY,2,'0')) AS TIMEID,CUST\_SSN,BRANCH\_CODE,TRANSACTION\_TYPE,TRANSACTION\_VALUE

FROM CDW\_SAPP\_CREDITCARD WHERE $CONDITIONS' --target-dir /user/maria\_dev/Credit\_card\_System/CREDITCARD --fields-terminated-by '\t' --

split-by ID -m1;

drop table CREDITCARD;

create table creditcard

( TRANSACTION\_ID INT, CUST\_CC\_NO STRING, TIMEID VARCHAR(8),

CUST\_SSN INT,BRANCH\_CODE INT, TRANSACTION\_TYPE VARCHAR(30),

TRANSACTION\_VALUE DECIMAL(20,3))

row format delimited fields terminated by '\t'

STORED AS TEXTFILE

Tblproperties("skip.header.line.count"="3");

LOAD DATA INPATH '/user/maria\_dev/Credit\_card\_System/CREDITCARD'

OVERWRITE INTO TABLE creditcard;

drop table CDW\_SAPP\_F\_CREDIT\_CARD;

create external table CDW\_SAPP\_F\_CREDIT\_CARD

( TRANSACTION\_ID INT, CUST\_CC\_NO STRING,CUST\_SSN INT,

BRANCH\_CODE INT,TRANSACTION\_TYPE VARCHAR(30),

TRANSACTION\_VALUE DECIMAL(20,3), TIMEID VARCHAR(8))

PARTITIONED BY (YEAR VARCHAR(4))

row format delimited fields terminated by ','

LINES TERMINATED BY '\n'

location '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_F\_CREDIT\_CARD/';

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE CDW\_SAPP\_F\_CREDIT\_CARD

PARTITION (YEAR)

SELECT TRANSACTION\_ID, CUST\_CC\_NO, CUST\_SSN,

BRANCH\_CODE ,TRANSACTION\_TYPE ,

TRANSACTION\_VALUE , TIMEID,SUBSTRING(TIMEID,1,4) AS YEAR

FROM creditcard;

2.2.5 Data Visualization

2.2.5 Visualization of Dataset

Credit Card System

Req-2.2.5

Data exploration and visualization

Functional Requirements Use Hive Query and Hive Visualization tool. 1) Find the top 20 zip codes(hint: branch\_zip) by total transaction value 2) Find total transaction value for each transaction type by Quarter in 2018 Hint: Find quarter from 'creditcard' table using month or use 'time' table if you already added transaction\_id column there.

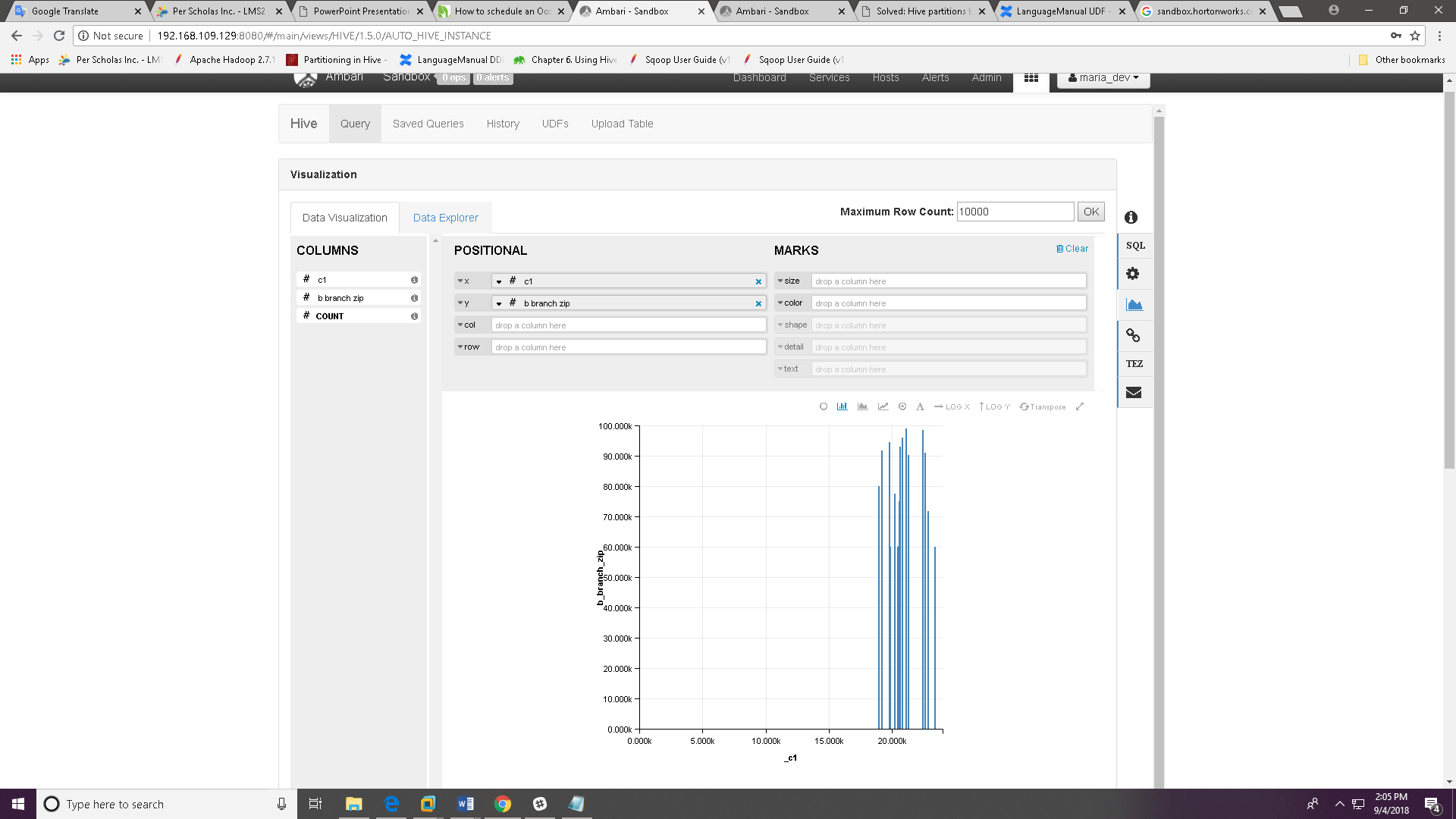
1) select b.branch\_zip, sum(transaction\_value) from cdw\_sapp\_f\_credit\_card c

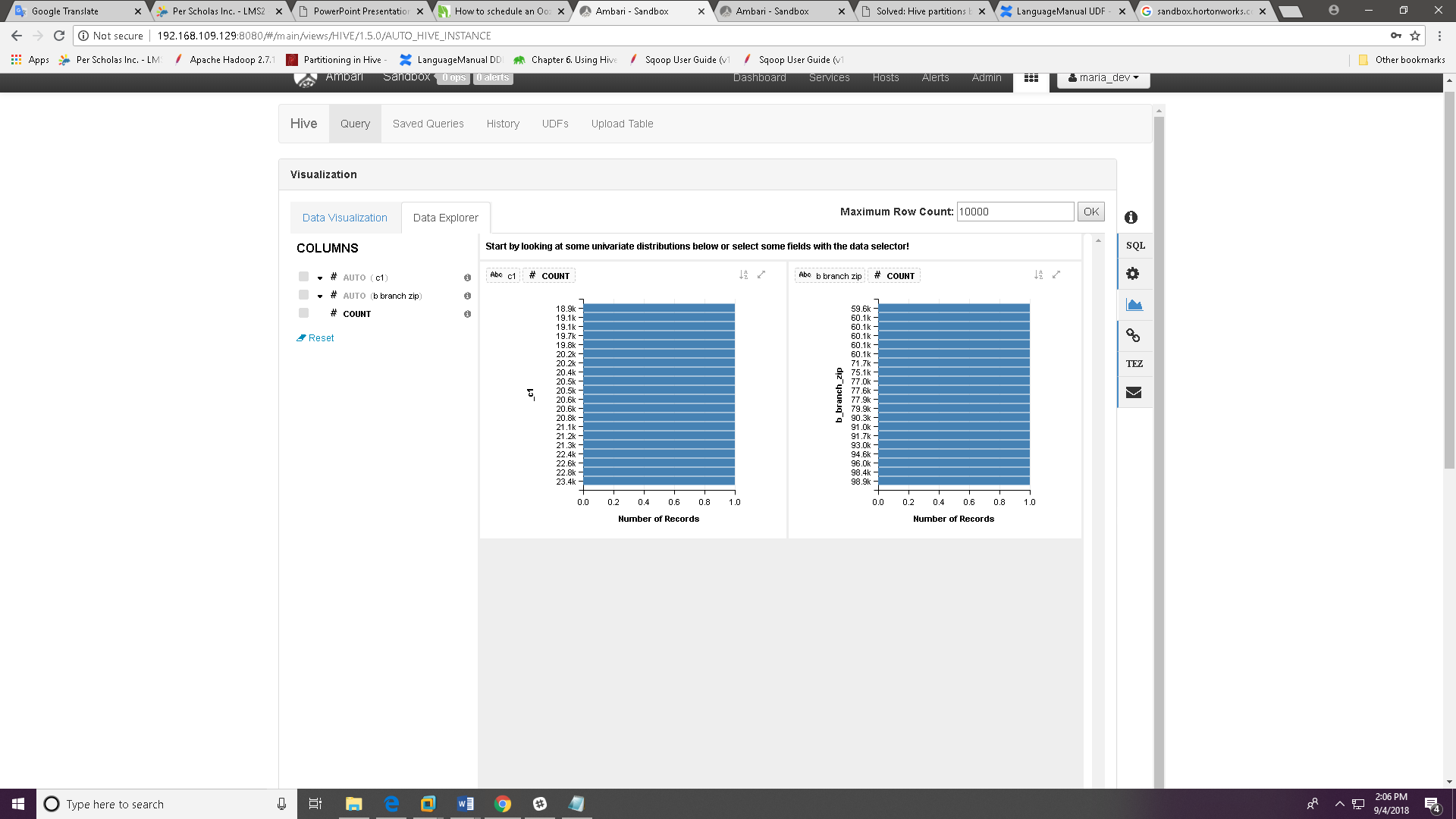
join cdw\_sapp\_d\_branch b on (c.branch\_code=b.branch\_code)

group by b.branch\_zip

order by b.branch\_zip desc limit 20;

|  |  |
| --- | --- |
| b.branch\_zip | \_c1 |
| 98908 | 21082.58 | 71730 | 22772.8 | 90278 | 21261.37 |
| 98444 | 22356.68 | 60148 | 20453.29 | 79930 | 18923.41 |
| 95993 | 20757.48 | 60103 | 20414.61 | 77904 | 21234.44 |
| 94580 | 19735.83 | 60091 | 23350.68 | 77566 | 20156.75 |
| 93035 | 20571.75 | 60089 | 19142.95 | 77016 | 20605.86 |
| 91740 | 19112.13 | 60060 | 19774.43 | 75088 | 20544.27 |
| 91010 | 22583.27 | 59601 | 20189.63 |





2)

sqoop import --connect jdbc:mysql://localhost/CDW\_SAPP --driver com.mysql.jdbc.Driver --query 'select concat(YEAR,lpad(MONTH,2,'0'),lpad(DAY,2,'0')) AS ID,DAY,MONTH,quarter(concat(YEAR,lpa

d(MONTH,2,0),lpad(DAY,2,0))) as QUARTER,YEAR,transaction\_id from CDW\_SAPP\_CREDITCARD WHERE $CONDITIONS' --target-dir /user/maria\_dev/Credit\_card\_System/time --fields-terminated-by '\t' --split-by ID -m1

drop table TIME;

create table TIME

( TIMEID VARCHAR(8), DAY INT, MONTH INT,

QUARTER VARCHAR(8), YEAR INT, TRANSACTION\_ID INT)

row format delimited fields terminated by '\t'

STORED AS TEXTFILE

Tblproperties("skip.header.line.count"="3");

LOAD DATA INPATH '/user/maria\_dev/Credit\_card\_System/time'

OVERWRITE INTO TABLE TIME;drop table CDW\_SAPP\_D\_TIME;

create external table CDW\_SAPP\_D\_TIME

( TIMEID VARCHAR(8), DAY INT, MONTH INT,

YEAR INT,TRANSACTION\_ID VARCHAR(4))

PARTITIONED BY (QUARTER VARCHAR(8))

row format delimited fields terminated by '\t'

LINES TERMINATED BY '\n'

location '/user/maria\_dev/Credit\_card\_System/CDW\_SAPP\_D\_TIME/';

SET hive.exec.dynamic.partition=true;

SET hive.exec.dynamic.partition.mode=nonstrict;

INSERT OVERWRITE TABLE CDW\_SAPP\_D\_TIME

PARTITION (QUARTER)

SELECT TIMEID, DAY, MONTH,

YEAR, TRANSACTION\_ID,QUARTER

FROM TIME;

select t.quarter, t.year, transaction\_type, sum(transaction\_value)

from cdw\_sapp\_f\_credit\_card c

join CDW\_SAPP\_D\_TIME t on (c.transaction\_id=t.transaction\_id)

group by t.quarter,t.year, c.transaction\_type

having t.year=2018

order by t.quarter,t.year, c.transaction\_type desc;

|  |  |  |  |
| --- | --- | --- | --- |
| t.quarter | t.year | transaction\_type | \_c3 |
| 1 | 2018 | Test | 85990 | 3 | 2018 | Test | 87521.19 |
| 1 | 2018 | Healthcare | 80736.7 | 3 | 2018 | Healthcare | 86299.67 |
| 1 | 2018 | Grocery | 76323.14 | 3 | 2018 | Grocery | 80955.42 |
| 1 | 2018 | Gas | 89927.84 | 3 | 2018 | Gas | 83453.32 |
| 1 | 2018 | Entertainment | 89341.24 | 3 | 2018 | Entertainment | 88121.49 |
| 1 | 2018 | Education | 80436.57 | 3 | 2018 | Education | 85844.52 |
| 1 | 2018 | Bills | 85816.21 | 3 | 2018 | Bills | 89673.01 |
| 2 | 2018 | Test | 86375.73 | 4 | 2018 | Test | 92064.83 |
| 2 | 2018 | Healthcare | 81910.16 | 4 | 2018 | Healthcare | 80670.91 |
| 2 | 2018 | Grocery | 80313.77 | 4 | 2018 | Grocery | 80987.24 |
| 2 | 2018 | Gas | 86317.78 | 4 | 2018 | Gas | 87481.44 |
| 2 | 2018 | Entertainment | 85485.3 | 4 | 2018 | Entertainment | 88950.8 |
| 2 | 2018 | Education | 84131.83 | 4 | 2018 | Education | 82839.13 |
| 2 | 2018 | Bills | 89326.39 | 4 | 2018 | Bills | 88255.17 |

