**Creditcard Defaulters and Insurance Premium Batch Project 2**

**Synopsis:**

This project module is to track and identify the creditcard defaulters and define the insurance premium according to the list of defaulters by joining data from insurance system, creditcard systems, state code fixed width data and the customer master datasets.

**Data preparation in the source systems DB to Hadoop Data Ingestion:**

**Mysql tables:**

mysql> describe credits\_cst;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Id | Int(11) | YES |  | NULL |  |
| lmt | Int(11) | YES |  | NULL |  |
| sex | Int(11) | YES |  | NULL |  |
| edu | Int(11) | YES |  | NULL |  |
| Marital | Int(11) | YES |  | NULL |  |
| age | Int(11) | YES |  | NULL |  |
| pay | Int(11) | YES |  | NULL |  |
| billamt | Int(11) | YES |  | NULL |  |
| defaulter | Int(11) | YES |  | NULL |  |
| Issuerid1 | Int(11) | YES |  | NULL |  |
| Issuerid2 | Int(11) | YES |  | NULL |  |
| tz | Varchar(3) | YES |  | NULL |  |

(defaulter: 0 – nondefaulter, 1 – defaulter)

mysql> describe credits\_pst;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Id | Int(11) | YES |  | NULL |  |
| lmt | Int(11) | YES |  | NULL |  |
| sex | Int(11) | YES |  | NULL |  |
| edu | Int(11) | YES |  | NULL |  |
| Marital | Int(11) | YES |  | NULL |  |
| age | Int(11) | YES |  | NULL |  |
| pay | Int(11) | YES |  | NULL |  |
| billamt | Int(11) | YES |  | NULL |  |
| defaulter | Int(11) | YES |  | NULL |  |
| Issuerid1 | Int(11) | YES |  | NULL |  |
| Issuerid2 | Int(11) | YES |  | NULL |  |
| tz | Varchar(3) | YES |  | NULL |  |

mysql> describe custmaster;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| Id | Int(11) | YES |  | NULL |  |
| fname | varchar(100) | YES |  | NULL |  |
| lname | varchar(100) | YES |  | NULL |  |
| ageval | Int(11) | YES |  | NULL |  |
| profession | varchar(100) | YES |  | NULL |  |

**HIVE TABLES:**

**Database: insure**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Mysql table** | **HDFS Location** | **Comments** |
| insure.credits\_pst | credits\_pst | /user/hduser/credits\_pst/ | id=bigint,billamt=float |
| insure.credits\_cst | credits\_cst | /user/hduser/credits\_cst/ | id=bigint,billamt=float |

Sqoop import --hive-import --hive-table insure.credits\_pst --hive-overwrite --map-column-hive id=bigint,billamt=float

Sqoop import --hive-import --hive-table insure.credits\_cst --hive-overwrite --map-column-hive id=bigint,billamt=float

Alter table credits\_cst SET TBLPROPERTIES('EXTERNAL' = 'TRUE')

**ETL & ELT using Hive**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Mysql table** | **HDFS Location** | **Comments** |
| insure.cstpstreorder |  | /user/hduser/cstpstreorder/ | Union of cst and pst tables where billamount > 0 |

sqoop import --connect jdbc:mysql://localhost/custdbproj --username root -P \

--query 'select \* from credits\_pst where billamt>0 and $CONDITIONS union all select \* from credits\_cst where billamt>0 and $CONDITIONS' \

--delete-target-dir --target-dir /user/hduser/cstpstreorderext -m 1\

--hive-import --hive-overwrite --hive-table insure.cstpstreorderext **--map-column-hive id=bigint,billamt=float**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Mysql table** | **HDFS Location** | **Comments** |
| insure.cstpstpenality  (stored as orc) |  | /user/hduser/cstpstpenalty/ | Created from cstpstreorder with calculated values for newlmt and newbillamount |

insure.cstpstpenality

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| id | bigint |  |
| Issuerid1 | int |  |
| Issuerid2 | int |  |
| Lmt | int |  |
| newlmt | int | **case defaulter when 1 then lmt-(lmt\*0.04) else lmt end as newlmt** |
| sex | int |  |
| edu | int |  |
| marital | int |  |
| pay | int |  |
| billamt | float |  |
| newbillamt | float | **case defaulter when 1 then billamt+(billamt\*.02) else billamt end as newbillamt** |
| defaulter | int |  |

**Data Provisioning using Hive, Sqoop and DistCp**

**Export and overwrite the above data into /user/hduser/defaultersout/:**

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/defaultersout/ | insure.cstpstpenality  (stored as orc) | Defaulter =1 |
| /user/hduser/nondefaultersout/ | insure.cstpstpenality  (stored as orc) | Defaulter =0 |

insert overwrite directory '/user/hduser/defaultersout/'

row format delimited fields terminated by ', '

select \* from insure.cstpstpenality where defaulter=1;

**Data Provisioning to the Consumers using DistCP**

**Copy the data non defaulters data from one cluster (Prod) to another cluster (Non Prod) for analytics purpose.**

hadoop **distcp -overwrite** hdfs://localhost:54310/user/hduser/nondefaultersout/ hdfs://localhost:54310/tmp/promocustomers

**Data Ingestion Pipeline**

|  |  |  |
| --- | --- | --- |
| **Cloud url** | **Script** | **Location** |
| bash sfm\_insuredata.sh https://s3.amazonaws.com/in.inceptez.bucket1/insurance\_project/insuranceinfo.csv | sfm\_insuredata.sh | /user/hduser/insurance\_clouddata |

**ETL & ELT using Hive**

Insure.insurance

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| Issuerid1 | int |  |
| Issuerid2 | int |  |
| businessyear | int |  |
| statecode | String |  |
| Sourcename | string |  |
| networkname | String |  |
| networkurl | String |  |
| rownumber | int |  |
| marketcoverage | string |  |
| dentalplan | string |  |

**provide your solution here to add the property to ignore the header record:**

**TBLPROPERTIES(“skip.header.line.count” =“1”);**

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/insurance\_clouddata | insure.insurance |  |

(OR)

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/insurance\_clouddata | insure.insurance | Partitioned by (datadt date,hr int) |

|  |  |  |
| --- | --- | --- |
| **Cloud url** | **Script** | **Location** |
| bash /home/hduser/creditcard\_insurance/hivepart.sh /user/hduser/insurance\_clouddata insure.insurance creditcard\_insurance | Hivepart.sh | /user/hduser/insurance\_clouddata |

**Delete the invalid data with null issuerid1 and issuerid2 using insert select query**

insert overwrite table insurance partition(datadt,hr) select \* from insurance where issuerid1 is not null and issuerid2 is not null;

**Create one more fixed width hive table to load the fixed width states\_fixedwidth data using Regex Serde**

**Insure.state\_master:**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| statecd | string | Loaded from/home/hduser/creditcard\_insurance/states\_fixedwidth |
| Statedesc | string |  |

**ROW FORMAT SERDE 'org.apache.hadoop.hive.contrib.serde2.RegexSerDe'**

**WITH SERDEPROPERTIES ("input.regex" = "(.{2})(.{20})" )**

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/states | Insure.state\_master |  |

**Create a managed table on top of the hive output defaulter’s dataset created above.**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| id | int |  |
| Issuerid1 | int |  |
| Issuerid2 | int |  |
| Lmt | int |  |
| newlmt | double | **case defaulter when 1 then lmt-(lmt\*0.04) else lmt end as newlmt** |
| sex | int |  |
| edu | int |  |
| marital | int |  |
| pay | int |  |
| billamt | int |  |
| newbillamt | float | **case defaulter when 1 then billamt+(billamt\*.02) else billamt end as newbillamt** |
| defaulter | int |  |

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/defaultersout | Insure.defaulters | Defaulter = 1 |

**Create a final managed table (later convert to external table) in orc with snappy compression and load the above 2 tables joined by applying different functions as given below .**

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| Issuerid | int | Insurance |
| businessyear | int |  |
| statecode | string |  |
| statedesc | string | State\_master |
| Sourcename | strings | insurance |
| networkname | string |  |
| networkurl | string |  |
| rownumber | int |  |
| marketcoverage | String |  |
| dentalplan | string |  |
| id | int | defaulters |
| Lmt | Int |  |
| newlmt | int |  |
| Reduced\_lmt | int | d.newlmt-d.lmt as reduced\_lmt |
| sex | Varchar(6) | case when d.sex=1 then 'male' else 'female'  end as sex |
| grade | Varchar(20) | case when d.edu=1 then 'lower grade'  when d.edu=2 then 'lower middle grade'  when d.edu=3 then 'middle grade'  when d.edu=4 then 'higher grade'  when d.edu=5 then 'doctrate grade'  end as grade |
| marital | Int |  |
| pay | Int |  |
| billamt | int |  |
| newbillamt | float |  |
| penality | float |  |
| defaulter | Int |  |

from insurance i inner join defaulters d on (i.IssuerId1=d.IssuerId1 and i.IssuerId2=d.IssuerId2)

inner join state\_master s on (i.statecode=s.statecd);

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/insurancetg | Insure.insurancestg | Joining insurance, state\_master,defaulters |

Insure.insuranceorc(same as insurancetg)

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/insuranceorc  (stored as orc) | Insure.insurancestg | Same as insurancetg |

**stored as orc**

**LOCATION '/user/hduser/insuranceorc'**

**TBLPROPERTIES ("immutable"="true","orc.compress"="SNAPPY");**

Insert into insuranceorc select \* from insurancestg where issuerid is not null;

**write common table expression queries in hive**

with T1 as ( select max(penality) as penalitymale from insuranceorc where sex='male'),

T2 as ( select max(penality) as penalityfemale from insuranceorc where sex='female')

select penalitymale,penalityfemale from T1 inner join T2 ON 1=1;

**Data Governance - Redaction and Masking using Hive and Python**

**Export the above view data into hdfs location /user/hduser/defaulterinfo with the pipe ‘|’ delimiter the following columns issuerid,businessyear,statedesc,sourcename,maskednetworkurl,sex,grade,marital,newbillamt,defaulter**

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/defaulterinfo | Insure.middlegradeview |  |

**Data Provisioning to the Consumers into legacy Databases using Sqoop including Validation**

**Data export using Sqoop into DB**

**Mysql tables:**

mysql> describe middlegrade;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| issuerid | Int(11) | YES |  | NULL |  |
| bussinessyear | Int(11) | YES |  | NULL |  |
| maskedstatedec | Varchar(200) | YES |  | NULL |  |
| maskedsourcename | Varchar(100) | YES |  | NULL |  |
| defaulter | Varchar(20) | YES |  | NULL |  |
| maskednetworkurl | Varchar(200) | YES |  | NULL |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Mysql table** | **HDFS Location** | **Comments** |
|  | middlegrade | /user/hduser/defaulterinfo/ |  |

sqoop export --connect jdbc:mysql://127.0.0.1/custdb -username root -password root -table middlegrade -export-dir /user/hduser/defaulterinfo/ --fields-terminated-by '|' **-validate**

**Create a Complex type table to understand how to group the like issuers in a single row using array, struct and map.**

insure.insuranceorc\_collection

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Comment** |
| issuerid | int |  |
| personalinfo | struct<marital:int,sex:varchar(6),grade:varchar(20)> |  |
| networkname | array<string> |  |
| Networkurl | array<string> |  |

alter table insuranceorc\_collection SET TBLPROPERTIES('EXTERNAL'='TRUE');

|  |  |  |
| --- | --- | --- |
| **HDFS Location** | **Hive table** | **Comments** |
| /user/hduser/insurance\_collection | Insure.insuranceorc\_collection |  |

**Select only the issuerid,grade,second networkname,second networkurl where we have more than 1 networkname and networkurl accessed by the customers.**

Select issuerid, personalinfo.grade, networkname[1] as networkname, networkurl[1] as networkurl from insuranceorc\_collection where size(networkname) =2 and size(networkurl) =2;

**Data movement & migration from DB to HBase using Sqoop**

**Join insurance and credit card data and load into hbase table created with 2 column families credit and insurance.**

**1. Import using sqoop from db into hbase custmaster data.**

sqoop import --connect jdbc:mysql://inceptez/custdb --username root --password root --table custmaster **--hbase-create-table --hbase-table** custmaster **--column-family customer** **--hbase-row-key** id - m 1 -validate &> /tmp/sqoop.log

mysql> describe custmaster;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| id | Int(11) | YES |  | NULL |  |
| fname | Varchar(100) | YES |  | NULL |  |
| lname | Varchar(100) | YES |  | NULL |  |
| ageval | Int(11) | YES |  | NULL |  |
| profession | Varchar(100) | YES |  | NULL |  |

|  |  |  |
| --- | --- | --- |
| **HBase table** | **Mysql table** | **columnfamily** |
| Custmaster | Custmaster | customer |

**Create a hbase handler table in hive using hbase storage handler referring to insurancehive table that will be automatically created in hbase with insurance and credit card column families when we create the below hive table.**

CREATE TABLE insurancehive (idkey int, issuerid int,id int,businessyear int,statedesc string,networkurl string,pay int,defaulter string,billamt int,newbillamt float,penality float)

**STORED** **BY 'org.apache.hadoop.hive.hbase.HBaseStorageHandler' WITH SERDEPROPERTIES ("hbase.columns.mapping" = ":key,insurance:issuerid,insurance:id,insurance:businessyear,insurance:statedesc,insurance:networkurl, creditcard:pay,creditcard:defaulter,creditcard:billamt,creditcard:newbillamt,creditcard:penality") TBLPROPERTIES ("hbase.table.name" = "insurancehive", "hbase.mapred.output.outputtable"="insurancehive");**

|  |  |  |
| --- | --- | --- |
| **HBase table** | **Hive table** | **columnfamily** |
| Insurancehive | Insure.insurancehive | Insurance,creditcard |

**Create a phoenix table view on the above hbase table and analyze profession based total payment and average payment.**

sqlline.py localhost

!set maxwidth 1000