#Script to extract Table From Traditional DBs and load to Hadoop and create hive table on top of Dataset

#Create by Sohail\_Mohammed@grainger.com

#Date 10-01-2015

usage() {

cat <<EOF

Extract Usage:

-I - Ip address and jdbc connection of DB

-D - Driver name of DB

-u - Username of DB

-p - Password of DB

-d - Database

-t - Table Name

-s - Split Column

-m - Mode(full/delta) enter "full" for full table extract(recommended for Dimension tables) or "delta" for partial table extract(Recommended for fact tables)

EOF

}

# -F - From Date format 'yyyy-mm-dd hh:mm:ss' in single quotes (Enter value ONLY if chose -m as delta)

# -T - To Date format 'yyyy-mm-dd hh:mm:ss' in single quotes (Enter value ONLY if chose -m as delta)

#EOF

#}

while getopts "I:D:u:p:d:t:s:m:F:T:" OPTION

do

case $OPTION in

I) DBIP=$OPTARG;;

D) DBDriver=$OPTARG;;

u) DBUserName=$OPTARG;;

p) DBPassword=$OPTARG;;

d) DataBase=$OPTARG;;

t) Table=$OPTARG;;

s) Split=$OPTARG;;

m) Mode=$OPTARG;;

# F) FromDt=$OPTARG;;

# T) ToDt=$OPTARG;;

esac

done

Date=`date +%Y%m%d`

Hdata=/data/raw/analytics/SRG/sqoop

LOGFILE=/hadoop/grainger/SRG/sqoop/logs

ToDt=`date --date="-36 days" +%Y-%m-%d`

FromDt=`date --date="-36 days" +%Y-%m-%d`

Script=/hadoop/grainger/SRG/sqoop/scripts

Hive\_warehouse=/apps/hive/warehouse

Schema=srg\_sqoop

File=$(echo "$Table" | tr '[:upper:]' '[:lower:]')

echo "Entered variables for sqoop extracts are" > ${LOGFILE}/${Table}\_${Date}

echo "${DBIP} ${DBDriver} ${DBUserName} ${DBPassword} ${DataBase} ${Table} ${Split} ${Mode} ${FromDt} / ${ToDt}" >> $LOGFILE/${Table}\_${Date}

if [[ -z ${DBIP} || -z ${DBDriver} || -z ${DBUserName} || -z ${DBPassword} || -z ${DataBase} || -z ${Table} || -z ${Split} || -z ${Mode} ]]

then echo "Options -u -p -d -t -s -m can not be empty">> ${LOGFILE}/${Table}\_${Date}

usage

exit 1

fi

#if [[ "${Mode}" = delta ]];then

# if [[ -z ${FromDt} ]] || [[ -z ${ToDt} ]];then

# echo "${FromDt} / ${ToDt} Options -f -t can not be empty">> $LOGFILE/${Table}\_${Date}

# usage

# exit 1

# elif [[ "${FromDt}" =~ ^([-: ]) || "${ToDt}" =~ ^([-: ]) ]];then

# echo "Please enter -f -t in yyyy-mm-dd hh:mm:ss in this format only ">> $LOGFILE/${Table}\_${Date}

# usage

# exit 1

# fi

if [[ "${Mode}" != full && "${Mode}" != delta ]];then

echo "For Option -m either enter delta or full ">> ${LOGFILE}/${Table}\_${Date}

usage

exit 1

fi

Extract\_Full(){

sqoop import \

--connect ${DBIP}/Database=${DataBase} \

--driver ${DBDriver} \

--username ${DBUserName} \

--password ${DBPassword} \

--table ${Table} \

--split-by ${Split} \

--compress \

--fields-terminated-by \| \

--hive-import \

--hive-table ${Table} \

--target-dir ${Hdata}/${Table}/${Table}\_${Date}

}

Extract\_Delta(){

sqoop import \

--connect ${DBIP}/Database=${DataBase} \

--driver ${DBDriver} \

--username ${DBUserName} \

--password ${DBPassword} \

--table ${Table} \

--split-by ${Split} \

--where "BATCH\_TS >= '${FromDt} 00:00:01' and BATCH\_TS <= '${ToDt} 23:59:59'" \

--compress \

--fields-terminated-by \| \

--hive-import \

--hive-table ${Table} \

--target-dir ${Hdata}/${Table}/${Table}\_${Date}

}

Hive\_Table(){

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" >> $LOGFILE/${Table}\_${Date}

echo " ${Table} dataset moving as is to hadoop location ${Hdata}/${Table}\_${Date} and creating hive table, if not present, on top of it" >> $LOGFILE/${Table}\_${Date}

if hadoop fs -test -d ${Hive\_warehouse}/${File};then

hadoop fs -mv ${Hive\_warehouse}/${File}/\* ${Hdata}/${Table}/${Table}\_${Date}/ >> $LOGFILE/${Table}\_${Date}

echo "show create table default.${Table};" > ${Script}/${Table}\_create\_table.hive

chmod 755 ${Script}/${Table}\_create\_table.hive

hive -f ${Script}/${Table}\_create\_table.hive 1 > ${Script}/hive\_table.sh 2>>$LOGFILE/${Table}\_${Date}

echo "DROP TABLE default.${Table};

CREATE TEMPORARY EXTERNAL TABLE ${Schema}.${Table}\_Tmp(" > ${Script}/${Table}\_hive.hive

cat ${Script}/hive\_table.sh| grep \`|cut -d'(' -f2| sed s/\`//g >> ${Script}/${Table}\_hive.hive

echo "ROW FORMAT DELIMITED FIELDS TERMINATED BY '|' LINES TERMINATED BY '\n'

STORED AS TEXTFILE

LOCATION '${Hdata}/${Table}/${Table}\_${Date}';">> ${Script}/${Table}\_hive.hive

if [[ "${mode}" = full ]];then

echo "DROP TABLE ${Schema}.${Table};">> ${Script}/${Table}\_hive.hive

fi

echo "CREATE TABLE IF NOT EXISTS ${Schema}.${Table}(">> ${Script}/${Table}\_hive.hive

cat ${Script}/hive\_table.sh| grep \`|cut -d'(' -f2| sed s/\`//g >> ${Script}/${Table}\_hive.hive

echo "STORED AS ORC;

FROM ${Schema}.${Table}\_Tmp

INSERT INTO TABLE ${Schema}.${Table}

SELECT \*; ">>${Script}/${Table}\_hive.hive

chmod 755 ${Script}/${Table}\_hive.hive

hive -f ${Script}/${Table}\_hive.hive>>$LOGFILE/${Table}\_${Date} 2>&1

if [[ $? -ne 0 ]];then

echo "Error...failed HIVE script for table ${Table}" >> $LOGFILE/${Table}\_${Date}

exit 1

fi

rm -f ${Script}/${Table}\*.\* >>$LOGFILE/${Table}\_${Date}

else

echo "NUMBER RECORDS FETCHED IS 0, SO TABLE IS NOT CREATED IN HIVE">>$LOGFILE/${Table}\_${Date}

exit 1

fi

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" >> $LOGFILE/${Table}\_${Date}

echo "${Table} Dataset copied as is to hadoop location ${Hdata}/${Table}\_${Date} and hive table created is ${Schema}.${Table}" >> $LOGFILE/${Table}\_${Date}

}

## \*\*\*\* method ends

if [[ "${Mode}" = full ]];then

echo "extracting full table, data first stores in hive warehouse">> $LOGFILE/${Table}\_${Date}

hadoop fs -rmr ${Hive\_warehouse}/${File}/\* >>$LOGFILE/${Table}\_${Date} 2>&1

hadoop fs -rmr ${Hdata}/${Table}/\* >>$LOGFILE/${Table}\_${Date} 2>&1

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" >> $LOGFILE/${Table}\_${Date}

echo " sqoop script to extract ${Table} starts" >> $LOGFILE/${Table}\_${Date}

Extract\_Full >> $LOGFILE/${Table}\_${Date} 2>&1

if [[ $? -ne 0 ]];then

echo "Error...failed sqoop script for table ${Table}" >> $LOGFILE/${Table}\_${Date}

exit 1

else

Hive\_Table

fi

else

echo "extracting based on date range and deleting already existing today's extractions">> $LOGFILE/${Table}\_${Date}

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" >> $LOGFILE/${Table}\_${Date}

echo " sqoop script to extract ${Table} starts" >> $LOGFILE/${Table}\_${Date}

hadoop fs -rmr ${Hive\_warehouse}/${File}/\* >>$LOGFILE/${Table}\_${Date} 2>&1

hadoop fs -rmr ${Hdata}/${Table}/${Table}\_${Date} >>$LOGFILE/${Table}\_${Date} 2>&1

Extract\_Delta >> $LOGFILE/${Table}\_${Date} 2>&1

if [[ $? -ne 0 ]];then

echo "Error...failed sqoop script for table ${Table}" >> $LOGFILE/${Table}\_${Date}

exit 1

else

Hive\_Table

fi

fi

echo "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" >> $LOGFILE/${Table}\_${Date}

echo " sqoop script to extract ${Table} ends" >> $LOGFILE/${Table}\_${Date}