1. **Passing arguments in hive.**
2. cnt=2java.lang.OutOfMemoryError: GC overhead limit exceeded

hive -e "select \* from employees.dept\_emp limit $cnt"

1. cnt=3

hive -hiveconf num=$cnt -e ' set num; select \* from employees.dept\_emp limit ${hiveconf:num}'

1. **Hive incremental load**

**STEP 1: INGEST**

sqoop import --connect jdbc:teradata://{host name or ip address}/Database=retail --connection-manager org.apache.sqoop.teradata.TeradataConnManager --username dbc --password dbc --table SOURCE\_TBL --target-dir /user/hive/incremental\_table -m 1

sqoop import --connect jdbc:teradata://{host name or ip address}/Database=retail --connection-manager org.apache.sqoop.teradata.TeradataConnManager --username dbc --password dbc --table SOURCE\_TBL --target-dir /user/hive/incremental\_table -m 1--check-column modified\_date --incremental lastmodified --last-value {last\_import\_date}

sqoop import --connect jdbc:teradata://{host name or ip address}/Database=retail --connection-manager org.apache.sqoop.teradata.TeradataConnManager --username dbc --password dbc --target-dir /user/hive/incremental\_table -m 1 --query 'select \* from SOURCE\_TBL where modified\_date > {last\_import\_date} AND $CONDITIONS’

**STEP 2: RECONCILE**

CREATE VIEW reconcile\_view AS SELECT t1.\* FROM

(SELECT \* FROM base\_table UNION ALL SELECT \* FROM incremental\_table) t1

JOIN

(SELECT id, max(modified\_date) max\_modified FROM (SELECT \* FROM base\_table UNION ALL SELECT \* FROM incremental\_table) t2

GROUP BY id) s

ON t1.id = s.id AND t1.modified\_date = s.max\_modified;

**STEP 3: COMPACT**

DROP TABLE reporting\_table;

CREATE TABLE reporting\_table AS SELECT \* FROM reconcile\_view;

**STEP 4: PURGE**

DROP TABLE base\_table;

CREATE TABLE base\_table AS SELECT \* FROM reporting\_table;

1. **java.lang.OutOfMemoryError: GC overhead limit exceeded**

SET mapred.child.java.opts=-Xmx4G -XX:+UseConcMarkSweepGC -XX:-UseGCOverheadLimit;

1. **Hive configuration tuning:**

export HADOOP\_CLIENT\_OPTS=" -Xmx2048m"

set hive.exec.dynamic.partition=true;

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

set hive.exec.max.dynamic.partitions=2048;

set hive.exec.max.dynamic.partitions.pernode=256;

set mapreduce.map.memory.mb=2048;

set yarn.scheduler.minimum-allocation-mb=2048;

set hive.exec.max.created.files=250000;

set hive.vectorized.execution.enabled=true;

set hive.merge.smallfiles.avgsize=283115520;

set hive.merge.size.per.task=209715200;

export HADOOP\_NAMENODE\_OPTS="-Dcom.sun.management.jmxremote"

export HADOOP\_NAMENODE\_OPTS="-XX:+UseConcMarkSweepGC"

export HADOOP\_NAMENODE\_OPTS="-XX:ParallelGCThreads=8"

export HADOOP\_NAMENODE\_OPTS="-XX:+UseCMSInitiatingOccupancyOnly"

export HADOOP\_NAMENODE\_OPTS="-XX:CMSInitiatingOccupancyFraction=70 -Xms1G -Xmx1G"

export HADOOP\_NAMENODE\_OPTS="-XX:NewSize=128M"

export HADOOP\_NAMENODE\_OPTS="-XX:MaxNewSize=128M "

export HADOOP\_NAMENODE\_OPTS="-XX:PermSize=128M "

export HADOOP\_NAMENODE\_OPTS="-XX:MaxPermSize=256M -verbose:gc -Xloggc:/Users/chris/hadoop-deploy-trunk/hadoop-3.0.0-SNAPSHOT/logs/gc.log-`date +'%Y%m%d%H%M'` "

export HADOOP\_NAMENODE\_OPTS="-XX:+PrintGCDetails "

export HADOOP\_NAMENODE\_OPTS="-XX:+PrintGCTimeStamps "

export HADOOP\_NAMENODE\_OPTS="-XX:+PrintGCDateStamps "

export HADOOP\_NAMENODE\_OPTS="-XX:ErrorFile=/Users/chris/hadoop-deploy-trunk/hadoop-3.0.0-SNAPSHOT/logs/hs\_err\_pid%p.log "

export HADOOP\_NAMENODE\_OPTS="-XX:+HeapDumpOnOutOfMemoryError $HADOOP\_NAMENODE\_OPTS""

# Hive Surrogate key

# hive slowly changing dimensions

# milestone load

# different newline delimiter

# custom serde