1. **Passing arguments in hive.**
2. cnt=2

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;

# Hive Surrogate key

# hive slowly changing dimensions

# milestone load