**Production strategy**

**Sqoop jobs with Append mode on sysrevid**

1. Sqoop:
   * Sqoop metastore start &
   * Sqoop job delete if any
     + Sqoop job --meta-connect "jdbc:hsqldb:hsql://hn0-yetisp:16000/sqoop" –delete <jobname>
   * Sqoop job create (new)
     + Sqoop job --meta-connect "jdbc:hsqldb:hsql://hn0-yetisp:16000/sqoop" –create <jobname>
   * Sqoop job execution
     + Sqoop job --meta-connect "jdbc:hsqldb:hsql://hn0-yetisp:16000/sqoop" –exec <jobname> -- <optional arguments>
2. Hive:

* Create external table in HIVE
  + This can be used as History Table.

Drop table if exists erp\_customerpricelst\_stg;

CREATE EXTERNAL TABLE IF NOT EXISTS erp\_customerpricelst\_stg

(

Company STRING

, CustNum INT

, ShipToNum STRING

, SeqNum INT

, ListCode STRING

, SysRevID BIGINT

, SysRowID STRING

, DwLastUpdated TIMESTAMP)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\001'

LINES TERMINATED BY '\n'

STORED AS TEXTFILE

LOCATION 'wasb://yetispark04@yetidatalake01.blob.core.windows.net/hive/warehouse/yetistaging50/erp\_customerpricelst\_stg';

* Create target table in ORC format.
  + This is the final staging table after removing duplicates and reconciling most recent records.

Drop table if exists erp\_customerpricelst\_stg\_orc;

Create table if not exists erp\_customerpricelst\_stg\_orc stored as ORC as

SELECT \* FROM

(SELECT \*, RANK() over (partition by company, CustNum, ShipToNum, SeqNum

order by SysRevID desc) as rank

FROM erp\_customerpricelst\_stg) ranked\_customerpricelst WHERE ranked\_customerpricelst.rank=1;

Drop table if exists erp\_customer\_stg\_orc;

Create table if not exists erp\_customer\_stg\_orc stored as ORC as

SELECT distinct \* FROM

(SELECT \*, RANK() over (partition by company, CustNum order by SysRevID desc) as rank

FROM erp\_customer\_stg) ranked\_customer WHERE ranked\_customer.rank=1;

**Questionaire:**

* What if staging Blob deleted accidentally?
  + We will lose all historical data
  + We need to delete, create and execute sqoop jobs once again. (it will take approx.. 5-6 hrs)
* What if you want to switch across the clusters?
  + We may lose historical data
  + backup data from spark04 cluster to Blob storage on Yetidatalake01
  + restore it on new cluster
  + We need to delete, create and execute sqoop jobs once again. (it will take approx.. 5-6 hrs)
* What if default hive external history table gets deleted?(order\_hist)
  + We can create external Hive external table from blob storage.
  + Oozie workflow will take care of it when you run next incremental load.
  + OR you can run it manually.
* What if default staging table gets deleted?(orderdtl)
  + We can create hive staging table from History table.
  + Oozie workflow will take care of it when you run next incremental load.
  + OR you can run it manually.
* What if your partition files increase?
  + We can run a job to merge those small files every month.
* What if you delete some of part files in the blob?
  + This may not happen in real-time.
  + We can work around on this by finding max and min sysrevids in the HDFS file.
  + re-run sqoop job to pull those records from SQL04.
* What if you want to switch blob folder on the same cluster?
  + You need to move all the data to new folder.
  + Modify hive and sqoop scripts with new folder name.
  + Re-run oozie workflows.
* Is there any dependency among Yetidm tables?
  + If yes, we need run them in a group.
  + Hive staging tables’ data is static and so you can run Yetidm scripts on them whenever you want.

Production issues:

* Sqoop job fails:
  + Sqoop metastore failure issues.
  + Sqoop job OOM error
* Hive and Yetidm script fails:
  + OOM error
  + Errors in hive table schema.

Work around:

* We can run Sqoop jobs every 5 mins.
* Hive , Yetidm and Atscale reporting jobs takes approx. (40+40+20) minutes.

Advantages:

* Data loss will be minimal.

Disadvantages:

* Work arounds will be difficult if you want to move data across cluster.
* Sqoop meta-store should always be up and running.

**Sqoop with Max\_sysrevid**

Advantages:

* Workaround will be easy when you move data between clusters or with in the same cluster.

Disadvantages:

* Data loss will more.
* Incremental uploads will take approximately 2 hrs (8+18+40+40+20)

Diff b/w Rank , rownumber and Dense rank:

Rank() function is faster than Joins and slower than Max function.

**Using insert overwrite command in hive incrementals**

* Create external Hive staging tables in a textfile format. (Initial Load)
* First Time Incremental Load:
  + Convert external hive staging table into managed table with ORC format in Hive Warehouse. (run /home/yeti/07152016/incremental/default/.hql files)
* Automate cronjobs for daily Incremental loads:
  + Run /home/yeti/07152016/incremental/default/archive/08222016/.hql files.
  + These scripts have Insert Overwrite table command.