Sqoop incremental and Meta store Hands on Activity

INCREMENTAL UPDATES AND INSERT

Hadoop and Hive are quickly evolving to outgrow previous limitations for integration and data access.
On the near-term development roadmap, we expect to see Hive supporting full CRUD operations (Insert, Select, Update, Delete). As we wait for these advancements, there is still a need to work with the current options—OVERWRITE or APPEND— for Hive table integration.

The OVERWRITE option requires moving the complete record set from source to Hadoop. While this approach may work for smaller data sets, it may be prohibitive at scale.

The APPEND option can limit data movement only to new or updated records. As true Inserts and Updates are not yet available in Hive, we need to consider a process of preventing duplicate records as Updates are appended to the cumulative record set.

Table . Incremental import arguments:

Argument	Description
--check-column (col)	Specifies the column to be examined when determining which rows to import.
--incremental (mode)	Specifies how Sqoop determines which rows are new. Legal values for mode include append and lastmodified.
--last-value (value)	Specifies the maximum value of the check column from the previous import.

Sqoop supports two types of incremental imports: append and lastmodified. You can use the --incremental argument to specify the type of incremental import to perform.

You should specify append mode when importing a table where new rows are continually being added with increasing row id values. You specify the column containing the row's id with --check-column. Sqoop imports rows where the check column has a value greater than the one specified with --last-value.

An alternate table update strategy supported by Sqoop is called lastmodified mode. You should use this when rows of the source table may be updated, and each such update will set the value of a last-modified column to the current timestamp. Rows where the check column holds a timestamp more recent than the timestamp specified with `--last-value` are imported.

At the end of an incremental import, the value which should be specified as `--last-value` for a subsequent import is printed to the screen. When running a subsequent import, you should specify `--last-value` in this way to ensure you import only the new or updated data. This is handled automatically by creating an incremental import as a saved job, which is the preferred mechanism for performing a recurring incremental import. See the section on saved jobs later in this document for more information.

A. create external table by any name at any location as mention below

```
CREATE EXTERNAL TABLE movies_externaltable (
  id varchar(10),
  branch_name varchar(25),
  release_date DATE
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
LINES TERMINATED BY '\n'
LOCATION "/user/maria_dev/foldername";
```

Using incremental append

//Below command will import data after id "200" row

```
sqoop import
--connect jdbc:mysql://localhost/movielens --driver com.mysql.jdbc.Driver --table movies -m 1
--target-dir /user/maria_dev/foldername
--incremental append
--check-column id
--last-value '200'
--fields-terminated-by '\t'
```

Using the --query

Note: we have to define target directory every time and we have to delete target directory if exist for below command every time
visit https://hortonworks.com/blog/four-step-strategy-incremental-updates-hive/

```

sqoop import
--connect jdbc:mysql://localhost/movielens --driver com.mysql.jdbc.Driver  -m 1
--query 'select * from  movies WHERE id > '1686' and $CONDITIONS'
--target-dir /user/maria_dev/foldername
--incremental append
--check-column release_date
--last-value '1990-01-01'
--fields-terminated-by '\t'
--incremental append
```

How to update HDFS directory

Using --incremental lastmodified (Modifiy and update)

```
sqoop import
--connect jdbc:mysql://localhost/movielens --driver com.mysql.jdbc.Driver --table movies -m 1
--append
--incremental lastmodified
--target-dir /user/maria_dev/foldername
--check-column release_date
--last-value '1990-01-01'
--outdir java_files
--fields-terminated-by '\t'
```

Sqoop job

This is very difficult to remember last value to avoid this we will create sqoop jobs. because sqoop job cab remember the last value.

- Last Value from the previous import acts as the argument for --last-value
- Sqoop Job checks for changes in data between the last value timestamp (Lower bound value) and Current timestamp (Upper bound value) and imports the modified or newly added rows.

Sqoop job commands

```
# To list existing Sqoop Jobs
sqoop job --list
# To show details of Sqoop Job
sqoop job --show incrementalImportJob
# To Execute Sqoop Job
sqoop job --exec incrementalImportJob
# To drop a Sqoop Job
sqoop job --delete incrementalImportJob
```

Below command will create sqoop job by the name of 'd3'. you can give any name to the job

```
sqoop job
--create d3
-- import
-m1
--connect jdbc:mysql://localhost/movielens
--driver com.mysql.jdbc.Driver
--query 'select * from movies WHERE $CONDITIONS'
--incremental append
--check-column id
--last-value '0'
--target-dir /user/maria_dev/foldername
--outdir java_files
--fields-terminated-by '\t'
```

Below command will execute/run Run sqoop Job

```
sqoop job --exec dailyImport
```

Create job for update external tables in hive

Note:You have to use 'date' field when you will use "--incremental lastmodified

Problem with lastmodified: it will not delete old data in hdfs .It will append new data(row) with older data(row) or get new data only. The Solution is use --merge-key instead of lastmodified

```
sqoop job
--create updateData
-- import
-m 1
--connect jdbc:mysql://localhost/movielens
--driver com.mysql.jdbc.Driver
--append
--query 'select * from movies WHERE $CONDITIONS'
--incremental lastmodified
--check-column changing_date
--last-value '1990-01-01'
--target-dir /user/maria_dev/foldername
--fields-terminated-by '\t'
```

// Run sqoop Job

```
sqoop job --exec updateData
```

Using Merge Key

```
sqoop job
--create updateData
-- import
-m 1
--connect jdbc:mysql://localhost/movielens
--driver com.mysql.jdbc.Driver
--query 'select * from movies WHERE $CONDITIONS'
--merge-key id
--incremental lastmodified
--check-column changing_date
--last-value '1990-01-01'
--target-dir /user/maria_dev/foldername
--fields-terminated-by '\t'
```

Sqoop metastore

Sqoop metastore is used to store Sqoop job information in a central place. This helps collaboration between Sqoop users and developers for example User A can create a job to load some specific data, then any other user can access from any node in the cluster the same job and just run it again. This is very convenient when using Sqoop in OOzie workflows.

Sqoop Metastore Syntax

```
$ sqoop metastore (generic-args) (metastore-args)
$ sqoop-metastore (generic-args) (metastore-args)
```

Procedure

At a high level, the following steps were followed:

A. Open any New terminal/command prompt then type below script

```
sqoop-metastore
```

B. Goto back to the previous/ old terminal then type below commands. Below commands will Create job in sqoop

```
sqoop job
--meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop
--create updateData
-- import
--connect jdbc:mysql://localhost/movielens
--driver com.mysql.jdbc.Driver
--query
'SELECT * FROM movies WHERE $CONDITIONS'
--fields-terminated-by '\t'
--incremental append
--check-column id
--last-value '1000'
--target-dir /user/maria_dev/updateData
-m1
```

C. Below command will show how many jobs are created in sqoop meta store

```
sqoop job  --meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop --list
```

D. Below command will Execute sqoop meta-store job. Type below command on previous terminal

```
sqoop job  --meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop --exec updateData
```

Reference:
https://sqoop.apache.org/docs/1.4.2/SqoopUserGuide.html#_saved_jobs_and_incremental_imports

http://www.hadooptechs.com/sqoop/sqoop-incremental-import-mysql-to-hive