Sqoop Import Deltas

Another very important feature is to allow a user using Sqoop to only import the deltas (or newly added rows since the last import). This helps to reduce the amount of work the sqoop job has to do and reduce load on the network. To do this we will need to learn about another concept in Sqoop called the Sqoop Metastore.

The Sqoop Metastore keeps track of the latest KEY that was used in the Sqoop import process for each table. So if I had a job that would periodically pull data from the customers table and use the customerNumber as the PRIMARY KEY, then the sqoop metastore would keep track of the last MAX value of that key so that way the next time the job gets kicked off, it can get the data from that last MAX number to the current MAX number.

The Sqoop Metastore, by default

How you can specify it in the sqoop command is with the --meta-connect argument:

* --meta-connect {connect\_string}
  + Connection string to the Sqoop Metastore

We can list the jobs that are currently in the Metastore with the following command:

$ sqoop job \

--meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop \

--list

Now that we’ve started up the Sqoop Metastore, we can now create jobs in the Sqoop Metastore which when ran will get the latest data from the target tables. To do this we will use the “sqoop job” command, connect to the Sqoop Metastore and create new jobs similar to how we ran the Sqoop Import command in the past:

* --meta-connect {connect\_string}
  + Connection string to the Sqoop Metastore
* --create {job\_name}
  + Some unique identifier for the job that we’re creating
  + We usually use the database name and table name as the identifier
    - Example: classicmodels.customers
* -- import
  + Note the space between ‘--‘ and ‘import’ in the above command. This separation is needed to help express what command that job is executing.
* --incremental {mode}
  + Specifies how Sqoop determines which rows are new. Legal values for mode include “append” and “lastmodified”.
* --check-column {column\_name}
  + Specifies the column to be examined when determining which rows to import.
  + This doesn’t automatically use the primary key. It needs to be specified.
* --last-value {value}
  + Specifies the maximum value of the check column from the previous import.
* All other parameters used in a traditional Sqoop Import process
  + --connect {connection\_string}
  + --table {table\_name}
  + etc.

$ sqoop job \

--meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop \

--create {job\_name}\

-- import \

--connect jdbc:mysql://localhost/classicmodels \

--username {user} \

--password {password} \

--table logs \

-m 2 \

--hive-import \

--hive-database {db\_name} \

--hive-table {table\_name} \

--incremental append \

--check-column {column\_name} \

--last-value 0

Note: See that there is a space between ‘--‘ and ‘import’ in the above command? This separation is needed to help express what command that job is executing. In this case where it is a regular export job, you would use “-- export” and so on.

You’ll notice that a Sqoop job was not triggered but rather just finished quickly. That’s because we merely created the job, we didn’t trigger it.

Now if you were to list the contents of the Sqoop Metastore, you would see the job listed:

$ sqoop job \

--meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop \

--list

we can execute the job to import what data is in the table. To do this we need to use the “sqoop job” command and specify the --exec argument:

* --exec {job\_id}
  + Unique identifier of the job you want to execute

$ sqoop job \

--meta-connect jdbc:hsqldb:hsql://localhost:16000/sqoop \

--exec {job\_name}