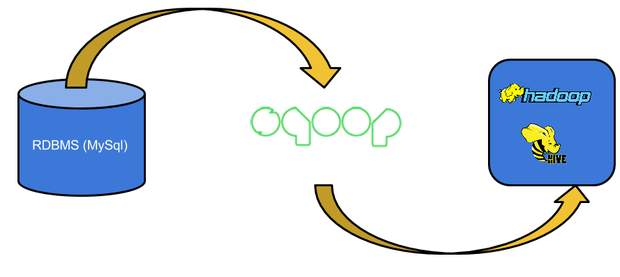
**Importing data from MySql to Hive using Sqoop**



**Prerequisite**: Hadoop Environment with Sqoop and Hive installed and working. To speed up the work, I am using Cloudera Quickstart VM  (requires 4GB of RAM) – although you can also work with Hortonworks Data Platform (requires 8GB of RAM) – since my laptop has only 8GB of RAM – I prefer to work with Cloudera VM image

If you are working with Cloudera/HDP VM and its all fired up in Virtualbox – it becomes easier to work with many of hadoop ecoystem packages comes pre-installed (mysql, oozie, hadoop,hive,zookeeper, storm, kafka, spark etc…)

**Create table in mysql**

In cloudera VM, open command prompt and just makesure mysql is installed. For me it’s

shell> mysql --version

mysql  Ver 14.14 Distrib 5.1.66, for redhat-linux-gnu (x86\_64) using readline 5.

You should always work in your own database, so create database in mysql using

mysql> create database sqoop;

then,

mysql> use sqoop;

mysql> create table customer(id varchar(3), name varchar(20), age varchar(3), salary integer(10));

Query OK, 0 rows affected (0.09 sec)

mysql> desc customer;

+--------+-------------+------+-----+---------+-------+

| Field  | Type        | Null | Key | Default | Extra |

+--------+-------------+------+-----+---------+-------+

| id     | varchar(3)  | YES  |     | NULL    |       |

| name   | varchar(20) | YES  |     | NULL    |       |

| age    | varchar(3)  | YES  |     | NULL    |       |

| salary | int(10)     | YES  |     | NULL    |       |

+--------+-------------+------+-----+---------+-------+

mysql> select \* from customer;

+------+--------+------+--------+

| id   | name   | age  | salary |

+------+--------+------+--------+

| 1    | John   | 30   |  80000 |

| 2    | Kevin  | 33   |  84000 |

| 3    | Mark   | 28   |  90000 |

| 4    | Jenna  | 34   |  93000 |

| 5    | Robert | 32   | 100000 |

| 6    | Zoya   | 40   |  60000 |

| 7    | Sam    | 37   |  75000 |

| 8    | George | 31   |  67000 |

| 9    | Peter  | 23   |  70000 |

| 19   | Alex   | 26   |  74000 |

+------+--------+------+-----

**Let’s start Sqooping**

As you can see **customer** table**does not have any primary key**. I have added few records in customet table. By default, Sqoop will identify the primary key column (if present) in a table and use it as the splitting column. The low and high values for the splitting column are retrieved from the database, and the map tasks operate on evenly-sized components of the total range.

If the actual values for the primary key are not uniformly distributed across its range, then this can result in unbalanced tasks. You should explicitly choose a different column with the --split-by argument. For example, --split-by id.

Since I want to import this table directly into hive I am adding –hive-import to my sqoop command.

sqoop import --connect jdbc:mysql://localhost:3306/sqoop

--username root

-P

--split-by id

--columns id,name

--table customer

--target-dir /user/cloudera/ingest/raw/customers

--fields-terminated-by ","

--hive-import

--create-hive-table

--hive-table sqoop\_workspace.customers

here’s what individual sqoop command option means

* **connect –** provide jdbc string
* **username** – database username
* **-P**  – will ask the password in console – alternatively you can use **–password** but this is not a good practice as its visible in your job execution logs and asking for trouble. One way to deal with this is store db password in a file in HDFS and provide at runtime.
* **table –**tell which table you want to import from mysql – here’s its customer
* **split-by –**specify whats your splitting column – I am specifying id here.
* **target-dir** – HDFS destination dir
* **fields-terminated-by** – I have specified comma (as by default it will import data into HDFS with comma separated values)
* **hive-import** – Import table into hive (Uses Hive’s default delimiters if none are set.)
* **create-hive-table** – If set job will fail if hive table already exist – it works in this case.
* **hive-table** – specify <db\_name>.<table\_name> here its sqoop\_workspace.customers where sqoop\_workspace is my database and customers is the table name.

As you can see below, sqoop is a map-reduce job. Notice that I am using -P for password option.  While this works, but can be easliy parameterized by using –password  and reading it from file.

sqoop import --connect jdbc:mysql://localhost:3306/sqoop --username root -P --split-by id --columns id,name --table customer  --target-dir /user/cloudera/ingest/raw/customers --fields-terminated-by "," --hive-import --create-hive-table --hive-table sqoop\_workspace.customers

Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.

Please set $ACCUMULO\_HOME to the root of your Accumulo installation.

16/03/01 12:59:44 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.5.0

**Enter password:**

16/03/01 12:59:54 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.

16/03/01 12:59:54 INFO tool.CodeGenTool: Beginning code generation

16/03/01 12:59:55 INFO manager.SqlManager: Executing SQL statement: SELECT t.\* FROM `customer` AS t LIMIT 1

16/03/01 12:59:56 INFO manager.SqlManager: Executing SQL statement: SELECT t.\* FROM `customer` AS t LIMIT 1

16/03/01 12:59:56 INFO orm.CompilationManager: HADOOP\_MAPRED\_HOME is /usr/lib/hadoop-mapreduce

Note: /tmp/sqoop-cloudera/compile/6471c43b5c867834458d3bf5a67eade2/customer.java uses or overrides a deprecated API.

Note: Recompile with -Xlint:deprecation for details.

16/03/01 13:00:01 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/6471c43b5c867834458d3bf5a67eade2/customer.jar

16/03/01 13:00:01 WARN manager.MySQLManager: It looks like you are importing from mysql.

16/03/01 13:00:01 WARN manager.MySQLManager: This transfer can be faster! Use the --direct

16/03/01 13:00:01 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.

16/03/01 13:00:01 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)

16/03/01 13:00:01 INFO mapreduce.ImportJobBase: Beginning import of customer

16/03/01 13:00:01 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address

16/03/01 13:00:02 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar

16/03/01 13:00:04 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps

16/03/01 13:00:05 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032

16/03/01 13:00:11 INFO db.DBInputFormat: Using read commited transaction isolation

16/03/01 13:00:11 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`id`), MAX(`id`) FROM `customer`

16/03/01 13:00:11 WARN db.TextSplitter: Generating splits for a textual index column.

16/03/01 13:00:11 WARN db.TextSplitter: If your database sorts in a case-insensitive order, this may result in a partial import or duplicate records.

16/03/01 13:00:11 WARN db.TextSplitter: You are strongly encouraged to choose an integral split column.

16/03/01 13:00:11 INFO mapreduce.JobSubmitter: number of splits:4

16/03/01 13:00:12 INFO mapreduce.JobSubmitter: Submitting tokens for job: job\_1456782715090\_0004

16/03/01 13:00:13 INFO impl.YarnClientImpl: Submitted application application\_1456782715090\_0004

16/03/01 13:00:13 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application\_1456782715090\_0004/

16/03/01 13:00:13 INFO mapreduce.Job: Running job: job\_1456782715090\_0004

16/03/01 13:00:47 INFO mapreduce.Job: Job job\_1456782715090\_0004 running in uber mode : false

16/03/01 13:00:48 INFO mapreduce.Job:  map 0% reduce 0%

16/03/01 13:01:43 INFO mapreduce.Job:  map 25% reduce 0%

16/03/01 13:01:46 INFO mapreduce.Job:  map 50% reduce 0%

16/03/01 13:01:48 INFO mapreduce.Job:  map 100% reduce 0%

16/03/01 13:01:48 INFO mapreduce.Job: Job job\_1456782715090\_0004 completed successfully

16/03/01 13:01:48 INFO mapreduce.Job: Counters: 30

    File System Counters

        FILE: Number of bytes read=0

        FILE: Number of bytes written=548096

        FILE: Number of read operations=0

        FILE: Number of large read operations=0

        FILE: Number of write operations=0

        HDFS: Number of bytes read=409

        HDFS: Number of bytes written=77

        HDFS: Number of read operations=16

        HDFS: Number of large read operations=0

        HDFS: Number of write operations=8

    Job Counters

        Launched map tasks=4

        Other local map tasks=5

        Total time spent by all maps in occupied slots (ms)=216810

        Total time spent by all reduces in occupied slots (ms)=0

        Total time spent by all map tasks (ms)=216810

        Total vcore-seconds taken by all map tasks=216810

        Total megabyte-seconds taken by all map tasks=222013440

    Map-Reduce Framework

        Map input records=10

        Map output records=10

        Input split bytes=409

        Spilled Records=0

        Failed Shuffles=0

        Merged Map outputs=0

        GC time elapsed (ms)=2400

        CPU time spent (ms)=5200

        Physical memory (bytes) snapshot=418557952

        Virtual memory (bytes) snapshot=6027804672

        Total committed heap usage (bytes)=243007488

    File Input Format Counters

        Bytes Read=0

    File Output Format Counters

        Bytes Written=77

16/03/01 13:01:48 INFO mapreduce.ImportJobBase: Transferred 77 bytes in 104.1093 seconds (0.7396 bytes/sec)

16/03/01 13:01:48 INFO mapreduce.ImportJobBase: Retrieved 10 records.

16/03/01 13:01:49 INFO manager.SqlManager: Executing SQL statement: SELECT t.\* FROM `customer` AS t LIMIT 1

16/03/01 13:01:49 INFO hive.HiveImport: Loading uploaded data into Hive

Logging initialized using configuration in jar:file:/usr/jars/hive-common-1.1.0-cdh5.5.0.jar!/hive-log4j.properties

OK

Time taken: 2.163 seconds

Loading data to table sqoop\_workspace.customers

chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/sqoop\_workspace.db/customers/part-m-00000': User does not belong to supergroup

chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/sqoop\_workspace.db/customers/part-m-00001': User does not belong to supergroup

chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/sqoop\_workspace.db/customers/part-m-00002': User does not belong to supergroup

chgrp: changing ownership of 'hdfs://quickstart.cloudera:8020/user/hive/warehouse/sqoop\_workspace.db/customers/part-m-00003': User does not belong to supergroup

Table sqoop\_workspace.customers stats: [numFiles=4, totalSize=77]

OK

Time taken: 1.399 seconds

Finally, let’s  verify the output in hive.

**Finally, verify output in Hive:**

hive> show databases;

OK

default

sqoop\_workspace

Time taken: 0.034 seconds, Fetched: 2 row(s)

hive> use sqoop\_workspace;

OK

Time taken: 0.063 seconds

hive> show tables;

OK

customers

Time taken: 0.036 seconds, Fetched: 1 row(s)

hive> show create table customers;

OK

CREATE TABLE `customers`(

  `id` string,

  `name` string)

COMMENT 'Imported by sqoop on 2016/03/01 13:01:49'

ROW FORMAT DELIMITED

  FIELDS TERMINATED BY ','

  LINES TERMINATED BY '\n'

STORED AS INPUTFORMAT

  'org.apache.hadoop.mapred.TextInputFormat'

OUTPUTFORMAT

  'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'

LOCATION

  'hdfs://quickstart.cloudera:8020/user/hive/warehouse/sqoop\_workspace.db/customers'

TBLPROPERTIES (

  'COLUMN\_STATS\_ACCURATE'='true',

  'numFiles'='4',

  'totalSize'='77',

  'transient\_lastDdlTime'='1456866115')

Time taken: 0.26 seconds, Fetched: 18 row(s)

hive> select \* from customers;  
OK  
1    John  
2    Kevin  
19    Alex  
3    Mark  
4    Jenna  
5    Robert  
6    Zoya  
7    Sam  
8    George  
9    Peter  
Time taken: 1.123 seconds, Fetched: 10 row(s).