Lab: Using Sqoop to Import Data

|  |  |
| --- | --- |
| **Objective:** | Move data between HDFS and a RDBMS. |
| **Successful Outcome:** | You will have imported data from MySQL into folders in HDFS, and exported data from HDFS into a MySQL table. |
| **Before You Begin:** | Go into terminal on hadoop-master. |

### Perform the following steps:

1. Install MySQL
   1. If necessary, run the following commands to install MySQL on hadoop-master:

# yum -y install mysql mysql-server

* 1. Start the server with the following command:

# service mysqld start

1. Create a Table in MySQL
   1. As the root user, change directories to /root/materials:
   2. View the contents of salaries.txt:

# cat salaries.txt

The comma-separated fields represent a gender, age, salary and zip code.

* 1. Notice there is a salaries.sql script that defines a new table in MySQL named salaries. For this script to work, you need to copy salaries.txt into the publicly-available /tmp folder:

# cp salaries.txt /tmp

* 1. Now run the salaries.sql script using the following command:

# mysql salaries < salaries.sql

1. View the Table
   1. To verify the table is populated in MySQL, open the mysql prompt:

# mysql

* 1. Switch to the test database, which is where the salaries table was created:

mysql> use test;

* 1. Run the show tables command and verify salaries is defined:

mysql> show tables;

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

| Tables\_in\_test |

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

| salaries |

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

1 row in set (0.00 sec)

* 1. Select 10 items from the table to verify it is populated:

mysql> select \* from salaries limit 10;

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

| gender | age | salary | zipcode | id |

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

| F | 66 | 41000 | 95103 | 1 |

| M | 40 | 76000 | 95102 | 2 |

| F | 58 | 95000 | 95103 | 3 |

| F | 68 | 60000 | 95105 | 4 |

| M | 85 | 14000 | 95102 | 5 |

| M | 14 | 0 | 95105 | 6 |

| M | 52 | 2000 | 94040 | 7 |

| M | 67 | 99000 | 94040 | 8 |

| F | 43 | 11000 | 94041 | 9 |

| F | 37 | 65000 | 94040 | 10 |

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

1. Grant the Necessary Privileges
   1. Enter the following command at the mysql prompt to grant access to node2 and node3 to connect to the mysql-server running on hadoop-master:

grant all privileges on \*.\* to 'root'@'%' with grant option;

* 1. Exit the mysql prompt:

mysql> exit

1. Import the Table into HDFS
   1. Enter the following Sqoop command (all on a single line), which imports the salaries table in the test database into HDFS:

# sqoop import

--connect jdbc:mysql://sandbox/test

--table salaries

--username root

* 1. A MapReduce job should start executing, and it may take a couple minutes for the job to complete.

1. Verify the Import
   1. View the contents of the salaries folder:

# hadoop fs -ls salaries

* 1. You should see a new folder named salaries. View its contents:

# hadoop fs -ls salaries

Found 4 items

-rw-r--r-- 1 root hdfs 272 part-m-00000

-rw-r--r-- 1 root hdfs 241 part-m-00001

-rw-r--r-- 1 root hdfs 238 part-m-00002

-rw-r--r-- 1 root hdfs 272 part-m-00003

* 1. Notice there are some new files in the salaries folder named part-m-0000x. Why are there this many of these files?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Use the cat command to view the contents of the files. For example:

# hadoop fs -cat salaries/part-m-00000

Notice the contents of these files are the rows from the salaries table in MySQL. You have now successfully imported data from a MySQL database into HDFS. Notice you imported the entire table with all of its columns. In the next step, you will import only specific columns of a table.

1. Specify Columns to Import
   1. Using the --columns argument, write a Sqoop command that imports the salary and age columns (in that order) of the salaries table into a directory in HDFS named salaries2. In addition, set the -m argument to 1 so that the result is a single file.
   2. After the import, verify you only have one part-m fie in salaries2:

# hadoop fs -ls salaries2

Found 1 items

-rw-r--r-- 1 root hdfs 482 salaries2/part-m-00000

* 1. Verify the contents of part-m-00000 are only the 2 columns you specified:

# hadoop fs -cat salaries2/part-m-00000

The last few lines should look like the following:

69000.0,97

91000.0,48

0.0,1

48000.0,45

3000.0,39

14000.0,84

1. Importing from a Query
   1. Write a Sqoop import command that imports the rows from salaries in MySQL whose salary column is greater than 90,000.00. Use gender as the --split-by value, specify only 2 mappers, and import the data into the salaries3 folder in HDFS.

**TIP**: The Sqoop command will look similar to the ones you have been using throughout this lab, except you will use --query instead of --table. Recall that when you use a --query command you must also define a --split-by column, or define -m to be 1.

Also, do not forget to add $CONDITIONS to the WHERE clause of your query, as demonstrated earlier in this Unit.

* 1. To verify the result, view the contents of the files in salaries3. You should have only two output files.
  2. View the contents of part-m-00000 and part-m-00001. Notice one file contains females, and the other file contains males. Why? \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Verify the output files contain only records whose salary is greater than 90,000.00.

1. Put the Export Data into HDFS
   1. Now let’s export data from HDFS to the database. Start by viewing the contents of the data, which is in a file named salarydata.txt:

# tail salarydata.txt

M,49,29000,95103

M,44,34000,95102

M,99,25000,94041

F,93,96000,95105

F,75,9000,94040

F,14,0,95102

M,68,1000,94040

F,45,78000,94041

M,40,6000,95103

F,82,5000,95050

Notice the records in this file contain 4 values separated by commas, and the values represent a gender, age, salary and zip code, respectively.

* 1. Create a new directory in HDFS named salarydata.
  2. Put salarydata.txt into the salarydata directory in HDFS.

1. Create a Table in the Database
   1. There is a script in the Lab3.2 folder that creates a table in MySQL that matches the records in salarydata.txt. View the SQL script:

# more salaries2.sql

* 1. Run this script using the following command:

# mysql test < salaries2.sql

* 1. Verify the table was created successfully in MySQL:

# mysql

mysql> use test;

mysql> describe salaries2;

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

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

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

| gender | varchar(1) | YES | | NULL | |

| age | int(11) | YES | | NULL | |

| salary | double | YES | | NULL | |

| zipcode | int(11) | YES | | NULL | |

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

* 1. Exit the mysql prompt:

mysql> exit

1. Export the Data
   1. Run a Sqoop command that exports the salarydata folder in HDFS into the salaries2 table in MySQL. At the end of the MapReduce output, you should see a log event stating that 10,000 records were exported.
   2. Verify it worked by viewing the table’s contents from the mysql prompt. The output should look like the following:

mysql> use test;

mysql> select \* from salaries2 limit 10;

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

| gender | age | salary | zipcode |

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

| M | 57 | 39000 | 95050 |

| F | 63 | 41000 | 95102 |

| M | 55 | 99000 | 94040 |

| M | 51 | 58000 | 95102 |

| M | 75 | 43000 | 95101 |

| M | 94 | 11000 | 95051 |

| M | 28 | 6000 | 94041 |

| M | 14 | 0 | 95102 |

| M | 3 | 0 | 95101 |

| M | 25 | 26000 | 94040 |

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

**RESULT**: You have imported the data from MySQL to HDFS using the entire table, specific columns, and also using the result of a query. You have also exported a folder of data in HDFS into a table in MySQL.

**SOLUTIONS**:

Step 7.1 is the following command (entered on a single line):

# sqoop import --connect jdbc:mysql://hadoop-master/test

--table salaries

--columns salary,age

-m 1

--target-dir salaries2

--username root

Step 8.1:

sqoop import --connect jdbc:mysql://hadoop-master/test

--query "select \* from salaries s where s.salary > 90000.00 and \$CONDITIONS"

--split-by gender

-m 2

--target-dir salaries3

--username root

Step 11

sqoop export

--connect jdbc:mysql://hadoop-master/test

--table salaries2

--export-dir salarydata

--input-fields-terminated-by ","  
--username root

**ANSWERS:**

Step 6.3: The MapReduce job that executed the Sqoop command used four mappers, so there are four output files (one from each mapper).

Step 8.3: You used gender as the split-by column, so all records with the same gender are sent to the same mapper.