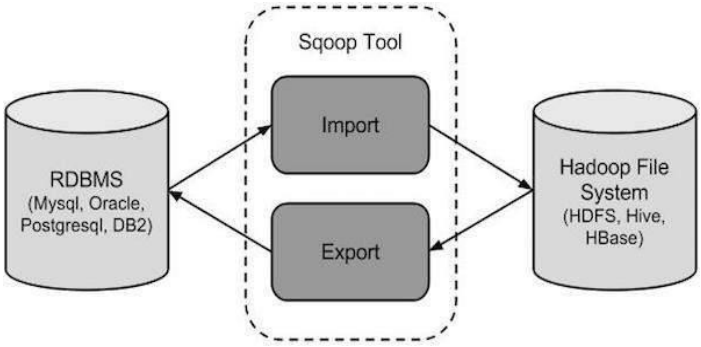
EXP 6: SQOOP

Aim: Perform SQOOP installation and commands.

Sqoop: It is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases. The traditional application management system, that is, the interaction of applications with relational database using RDBMS, is one of the sources that generate Big Data. Such Big Data, generated by RDBMS, is stored in **Relational Database Servers** in the relational database structure.
When Big Data storages and analyzers such as MapReduce, Hive, HBase, Cassandra, Pig, etc. of the Hadoop ecosystem came into picture, they required a tool to interact with the relational database servers for importing and exporting the Big Data residing in them. Here, Sqoop occupies a place in the Hadoop ecosystem to provide feasible interaction between relational database server and Hadoop's HDFS.

Sqoop: "SQL to Hadoop and Hadoop to SQL"
Sqoop is a tool designed to transfer data between Hadoop and relational database servers. It is used to import data from relational databases such as MySQL, Oracle to Hadoop HDFS, and export from Hadoop file system to relational databases. It is provided by the Apache Software Foundation.
The following image describes the workflow of Sqoop.



- **Sqoop Import:** The import tool imports individual tables from RDBMS to HDFS. Each row in a table is treated as a record in HDFS. All records are stored as text data in text files or as binary data in Avro and Sequence files.
- **Sqoop Export:** The export tool exports a set of files from HDFS back to an RDBMS. The files given as input to Sqoop contain records, which are called as rows in table. Those are read and parsed into a set of records and delimited with user-specified delimiter.

STEPS:

1. Extract the Sqoop Package from the tar file pasted on the Desktop. The extracted package can be seen, listed in the list of files and folders of the Desktop using the **ls** command.

```
jdk@ubuntu: ~
jdk@ubuntu:~$ ls Desktop
file1.txt                          WordCount$IntSumReducer.class
sqoop-1.4.6.bin__hadoop-2.0.4-alpha  WordCount.java
wc.jar                             WordCount$TokenizerMapper.class
WordCount.class
jdk@ubuntu:~$
jdk@ubuntu:~$ sudo mv Desktop/sqoop-1.4.6.bin__hadoop-2.0.4-alpha /usr/lib/sqoop
[sudo] password for jdk:
jdk@ubuntu:~$
```

2. Move this extracted package from Desktop to the directory **/usr/lib/sqoop** using the **sudo mv** command.

3. Sqoop environment can be set up only by appending the following lines by executing **sudo gedit ~/.bashrc** command.

```
jdk@ubuntu:~$ sudo gedit ~/.bashrc

(gedit:7097): IBUS-WARNING **: The owner of /home/jdk/.config/ibus/bus is not ro
ot!

(gedit:7097): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

** (gedit:7097): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-spell-enabled not supported

** (gedit:7097): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-encoding not supported

** (gedit:7097): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-position not supported
jdk@ubuntu:~$
```

Append the following lines in this file.

```
export SQOOP_HOME=/usr/lib/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
```

```
Open                                                    Save
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export HADOOP_INSTALL=/usr/local/hadoop
export PATH=$PATH:$HADOOP_INSTALL/bin
export PATH=$PATH:$HADOOP_INSTALL/sbin
export HADOOP_MAPRED_HOME=$HADOOP_INSTALL
export HADOOP_COMMON_HOME=$HADOOP_INSTALL
export HADOOP_HDFS_HOME=$HADOOP_INSTALL
export YARN_HOME=$HADOOP_INSTALL
export HADOOP_CLASSPATH=$JAVA_HOME/lib/tools.jar
export PIG_HOME=/usr/local/pig
export PATH=$PATH:$PIG_HOME/bin
export HIVE_HOME=/usr/local/hive
export PATH=$PATH:$HIVE_HOME/bin
export SQOOP_HOME=/usr/lib/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
```

4. Now save this bashrc file permanently by the command **source ~/.bashrc**

```
jdk@ubuntu: ~
jdk@ubuntu:~$ source ~/.bashrc
jdk@ubuntu:~$
```

5. To configure Sqoop with Hadoop we need to edit a file **sqoop-env.sh** which is present in the directory path

```
jdk@ubuntu: /usr/lib/sqoop/conf
jdk@ubuntu:~$ cd $SQOOP_HOME/conf
jdk@ubuntu:/usr/lib/sqoop/conf$
```

$SQOOP_HOME/conf.

Now move the contents of the template file sqoop-env-template.sh to sqoop-env.sh using the **mv** command.

```
jdk@ubuntu: /usr/lib/sqoop/conf
jdk@ubuntu:~$ cd $SQOOP_HOME/conf
jdk@ubuntu:/usr/lib/sqoop/conf$ mv sqoop-env-template.sh sqoop-env.sh
jdk@ubuntu:/usr/lib/sqoop/conf$
```

To add contents in the **sqoop-env.sh** file use the command:

sudo gedit sqoop-env.sh

```
jdk@ubuntu:/usr/lib/sqoop/conf$ sudo gedit sqoop-env.sh
[sudo] password for jdk:

(gedit:2690): IBUS-WARNING **: The owner of /home/jdk/.config/ibus/bus is not ro
ot!

(gedit:2690): Gtk-WARNING **: Calling Inhibit failed: GDBus.Error:org.freedeskto
p.DBus.Error.ServiceUnknown: The name org.gnome.SessionManager was not provided
by any .service files

** (gedit:2690): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-spell-enabled not supported

** (gedit:2690): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-encoding not supported

** (gedit:2690): WARNING **: Set document metadata failed: Setting attribute met
adata::gedit-position not supported
jdk@ubuntu:/usr/lib/sqoop/conf$
```

Add these lines in the file code:

export HADOOP_COMMON_HOME=/usr/local/hadoop
export HADOOP_MAPRED_HOME=/usr/local/hadoop

Open | sqoop-env.sh /usr/lib/sqoop/conf | Save

```
# the License.  You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

# included in all the hadoop scripts with source command
# should not be executable directly
# also should not be passed any arguments, since we need original $*

# Set Hadoop-specific environment variables here.

#Set path to where bin/hadoop is available
#export HADOOP_COMMON_HOME=
export HADOOP_COMMON_HOME=/usr/local/hadoop

#Set path to where hadoop-*-core.jar is available
#export HADOOP_MAPRED_HOME=
export HADOOP_MAPRED_HOME=/usr/local/hadoop

#set the path to where bin/hbase is available
#export HBASE_HOME=

#Set the path to where bin/hive is available
#export HIVE_HOME=
```

sh | Tab Width: 8 | Ln 28, Col 44 | INS

6. Now copy, add or download the **mysql-connector-java-5.1.36.tar.gz** file onto the Desktop. Extract this file in the same file location.

```
jdk@ubuntu:~$ cd Desktop
jdk@ubuntu:~/Desktop$ ls
file1.txt                          WordCount$IntSumReducer.class
mysql-connector-java-5.1.36.tar.gz WordCount.java
wc.jar                             WordCount$TokenizerMapper.class
WordCount.class
jdk@ubuntu:~/Desktop$ tar mysql-connector-java-5.1.36.tar.gz
tar: Old option 'g' requires an argument.
Try 'tar --help' or 'tar --usage' for more information.
jdk@ubuntu:~/Desktop$ tar -zxf mysql-connector-java-5.1.36.tar.gz
jdk@ubuntu:~/Desktop$
```

7. Move this extracted file to the location **/usr/lib/sqoop/lib** using the **mv** command.

```
jdk@ubuntu: ~/Desktop/mysql-connector-java-5.1.36
jdk@ubuntu:~/Desktop$ cd mysql-connector-java-5.1.36
jdk@ubuntu:~/Desktop/mysql-connector-java-5.1.36$ ls
build.xml  COPYING  mysql-connector-java-5.1.36-bin.jar  README.txt
CHANGES    docs     README                                src
jdk@ubuntu:~/Desktop/mysql-connector-java-5.1.36$ mv mysql-connector-java-5.1.36
-bin.jar /usr/lib/sqoop/lib
jdk@ubuntu:~/Desktop/mysql-connector-java-5.1.36$ ls /usr/lib/sqoop/lib
ant-contrib-1.0b3.jar          kite-data-mapreduce-1.0.0.jar
ant-eclipse-1.0-jvm1.2.jar     kite-hadoop-compatibility-1.0.0.jar
avro-1.7.5.jar                 mysql-connector-java-5.1.36-bin.jar
avro-mapred-1.7.5-hadoop2.jar  opencsv-2.3.jar
commons-codec-1.4.jar          paranamer-2.3.jar
commons-compress-1.4.1.jar     parquet-avro-1.4.1.jar
commons-io-1.4.jar             parquet-column-1.4.1.jar
commons-jexl-2.1.1.jar         parquet-common-1.4.1.jar
commons-logging-1.1.1.jar      parquet-encoding-1.4.1.jar
hsqldb-1.8.0.10.jar            parquet-format-2.0.0.jar
jackson-annotations-2.3.0.jar  parquet-generator-1.4.1.jar
jackson-core-2.3.1.jar         parquet-hadoop-1.4.1.jar
jackson-core-asl-1.9.13.jar    parquet-jackson-1.4.1.jar
jackson-databind-2.3.1.jar     slf4j-api-1.6.1.jar
jackson-mapper-asl-1.9.13.jar  snappy-java-1.0.5.jar
kite-data-core-1.0.0.jar       xz-1.0.jar
kite-data-hive-1.0.0.jar
```

8. To check if Sqoop has been installed correctly we move to the directory **$SQOOP_HOME/bin** and use the command **sqoop version** to check for sqoop installation success.

```
jdk@ubuntu:/usr/lib/sqoop/bin$ cd $SQOOP_HOME/bin
jdk@ubuntu:/usr/lib/sqoop/bin$ sqoop version
Warning: /usr/lib/sqoop/../hbase does not exist! HBase imports will fail.
Please set $HBASE_HOME to the root of your HBase installation.
Warning: /usr/lib/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/lib/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
16/07/01 22:51:06 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6
Sqoop 1.4.6
git commit id c0c5a81723759fa575844a0a1eae8f510fa32c25
Compiled by root on Mon Apr 27 14:38:36 CST 2015
jdk@ubuntu:/usr/lib/sqoop/bin$
```

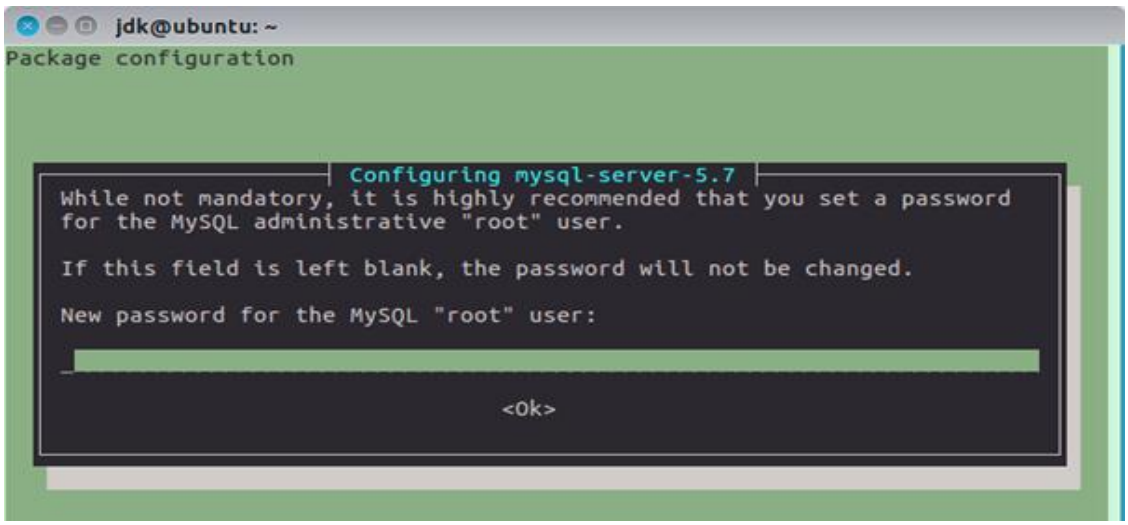
MYSQL INSTALLATION

1. After Sqoop installation MySQL has to be installed as well. Firstly, install all the required libraries using the command **sudo apt-get install mysql-server**.

```
jdk@ubuntu: ~
jdk@ubuntu:~$ sudo apt-get install mysql-server
[sudo] password for jdk:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libhtml-template-perl mysql-client-5.7 mysql-client-core-5.7
  mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libipc-sharedcache-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libhtml-template-perl mysql-client-5.7 mysql-client-core-5.7
  mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 8 newly installed, 0 to remove and 39 not upgraded.
Need to get 18.0 MB of archives.
After this operation, 160 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

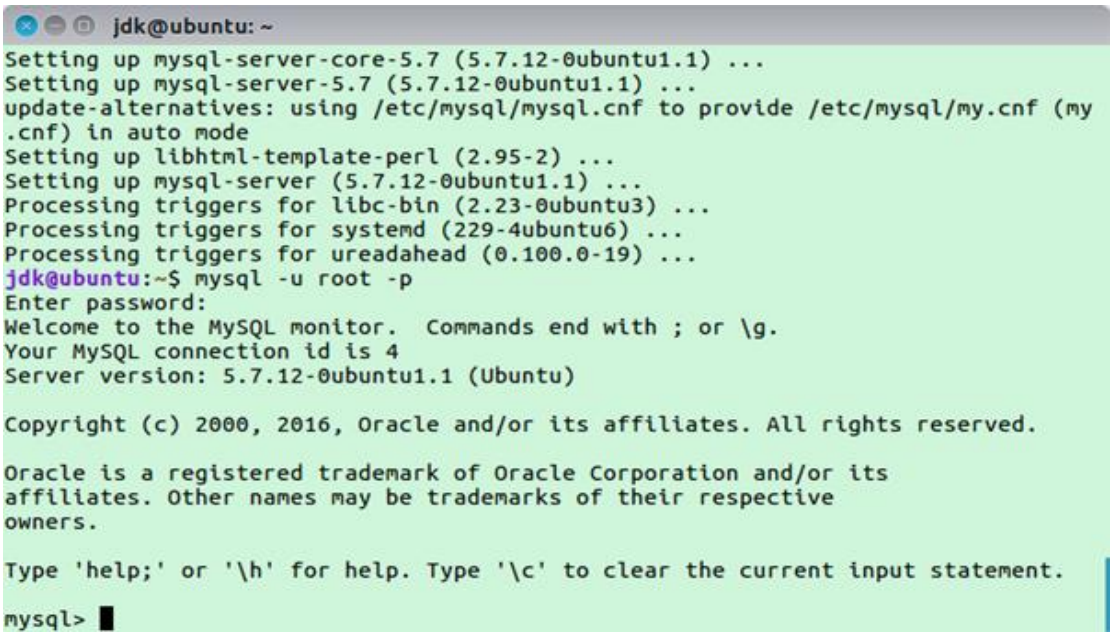
After getting this message press **Y** to continue. The libraries get downloaded successfully.

2. Now a screen like the one below appears. It prompts the user to set a password for the user “root”. So, we can set the password as we wish it to be.



3. To login to the MySQL user, use the following command:

mysql -u root -p



It will be asked to enter the password for the corresponding user. Enter the password. Now the MySQL script will run and the user will be logged in. This verifies the successful completion of the MySQL installation onto the system.

COMMANDS:

//login to mysql use below command. **(MYSQL)**

[cloudera@quickstart ~]$ mysql -uroot -pcloudera
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 5.1.73 Source distribution
Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database nptz;
Query OK, 1 row affected (0.00 sec)

mysql> use nptz;
Database changed

mysql> create table employee (id int, name varchar(15),salary int, destination varchar(10));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into employee values(1,'zoya',20000,'java developer');
Query OK, 1 row affected, 1 warning (0.10 sec)

mysql> insert into employee values(2,'pranali',30000,'c++ developer');
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> insert into employee values(3,'nazmeen',20000,'python developer');
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> insert into employee values(4,'taniya',40000,'android developer');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into employee values(5,'vasim',20000,'web developer');
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> insert into employee values(6,'arshad',30000,'web developer');
Query OK, 1 row affected, 1 warning (0.01 sec)

mysql> select * from employee;

id	name	salary	destination
1	zoya	20000	java devel
2	pranali	30000	c++ develo
3	nazmeen	20000	python dev
4	taniya	40000	android de
5	vasim	20000	web develo
6	arshad	30000	web develo

6 rows in set (0.01 sec)
//back to main directory
//sql to hadoop **(SQOOP)**

[cloudera@quickstart ~]$ **sqoop** import --connect jdbc:mysql://localhost:3306/nptz --username root -P --split-by id --columns id,name,salary,destination --table employee --target-dir /user/cloudera/nptz --fields-terminated-by "\t"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
19/09/28 23:53:10 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
19/09/28 23:53:16 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
19/09/28 23:53:16 INFO tool.CodeGenTool: Beginning code generation
19/09/28 23:53:18 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
19/09/28 23:53:18 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
19/09/28 23:53:18 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce

```
Note: /tmp/sqoop-cloudera/compile/e45a2d7895e9ba3c0103f4ce50876547/employee.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
19/09/28 23:53:33 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/e45a2d7895e9ba3c0103f4ce50876547/employee.jar
19/09/28 23:53:33 WARN manager.MySQLManager: It looks like you are importing from mysql.
19/09/28 23:53:33 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
19/09/28 23:53:33 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
19/09/28 23:53:33 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
19/09/28 23:53:33 INFO mapreduce.ImportJobBase: Beginning import of employee
19/09/28 23:53:33 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
19/09/28 23:53:35 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
19/09/28 23:53:42 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
19/09/28 23:53:42 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/09/28 23:53:58 INFO db.DBInputFormat: Using read commited transaction isolation
19/09/28 23:53:58 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`id`), MAX(`id`) FROM `employee`
19/09/28 23:53:58 INFO db.IntegerSplitter: Split size: 1; Num splits: 4 from: 1 to: 6
19/09/28 23:53:59 INFO mapreduce.JobSubmitter: number of splits:4
19/09/28 23:54:01 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1569733481333_0002
19/09/28 23:54:04 INFO impl.YarnClientImpl: Submitted application application_1569733481333_0002
19/09/28 23:54:04 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1569733481333_0002/
19/09/28 23:54:04 INFO mapreduce.Job: Running job: job_1569733481333_0002
19/09/28 23:54:52 INFO mapreduce.Job: Job job_1569733481333_0002 running in uber mode : false
19/09/28 23:54:52 INFO mapreduce.Job:  map 0% reduce 0%
19/09/28 23:55:51 INFO mapreduce.Job:  map 100% reduce 0%
19/09/28 23:55:57 INFO mapreduce.Job: Job job_1569733481333_0002 completed successfully
19/09/28 23:55:58 INFO mapreduce.Job: Counters: 30
        File System Counters
                FILE: Number of bytes read=0
                FILE: Number of bytes written=686532
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=393
                HDFS: Number of bytes written=155
                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=4
                Total time spent by all maps in occupied slots (ms)=215502
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=215502
                Total vcore-milliseconds taken by all map tasks=215502
                Total megabyte-milliseconds taken by all map tasks=220674048
        Map-Reduce Framework
                Map input records=6
                Map output records=6
                Input split bytes=393
                Spilled Records=0
                Failed Shuffles=0
```

```
                Merged Map outputs=0
                GC time elapsed (ms)=2775
                CPU time spent (ms)=14580
                Physical memory (bytes) snapshot=743948288
                Virtual memory (bytes) snapshot=6305464320
                Total committed heap usage (bytes)=591921152
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=155
19/09/28 23:55:59 INFO mapreduce.ImportJobBase: Transferred 155 bytes in 136.7866 seconds (1.1332
bytes/sec)
19/09/28 23:55:59 INFO mapreduce.ImportJobBase: Retrieved 6 records.

[cloudera@quickstart ~]$
```

//to see the loaded file in hadoop from mysql using SQOOP:

```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/nptz/
Found 5 items
-rw-r--r--   1 cloudera cloudera        0 2019-09-28 23:55 /user/cloudera/nptz/_SUCCESS
-rw-r--r--   1 cloudera cloudera       51 2019-09-28 23:55 /user/cloudera/nptz/part-m-00000
-rw-r--r--   1 cloudera cloudera       27 2019-09-28 23:55 /user/cloudera/nptz/part-m-00001
-rw-r--r--   1 cloudera cloudera       26 2019-09-28 23:55 /user/cloudera/nptz/part-m-00002
-rw-r--r--   1 cloudera cloudera       51 2019-09-28 23:55 /user/cloudera/nptz/part-m-00003

[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/nptz/
Found 5 items
-rw-r--r--   1 cloudera cloudera        0 2019-09-28 23:55 /user/cloudera/nptz/_SUCCESS
-rw-r--r--   1 cloudera cloudera       51 2019-09-28 23:55 /user/cloudera/nptz/part-m-00000
-rw-r--r--   1 cloudera cloudera       27 2019-09-28 23:55 /user/cloudera/nptz/part-m-00001
-rw-r--r--   1 cloudera cloudera       26 2019-09-28 23:55 /user/cloudera/nptz/part-m-00002
-rw-r--r--   1 cloudera cloudera       51 2019-09-28 23:55 /user/cloudera/nptz/part-m-00003

[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/nptz/part-m-00000
1       zoya    20000   java devel    //column size is less(varchar)
2       pranali 30000   c++ develo    //column size is less(varchar)

[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/nptz/part-m-00001
3       nazmeen         20000   python dev

[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/nptz/part-m-00002
4       taniya  40000   android de

[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/nptz/part-m-00003
5       vasim   20000   web develo
6       arshad  30000   web develo

[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/nptz/part-m-00000
1       zoya    20000   java devel
2       pranali 30000   c++ develo
```

//sql to hive

[cloudera@quickstart ~]$ **sqoop** import --connect jdbc:mysql://localhost:3306/nptz --username root -P --split-by id --columns id,name,salary,destination --table employee --target-dir /user/cloudera/group2 --fields-terminated-by "\t" --hive-import --hive-table default.employee
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
19/09/29 00:21:01 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
19/09/29 00:21:05 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
19/09/29 00:21:05 INFO tool.CodeGenTool: Beginning code generation
19/09/29 00:21:07 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
19/09/29 00:21:07 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
19/09/29 00:21:07 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/eee807a7078c62605ce1e5c0d68dc3a2/employee.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
19/09/29 00:21:16 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/eee807a7078c62605ce1e5c0d68dc3a2/employee.jar
19/09/29 00:21:16 WARN manager.MySQLManager: It looks like you are importing from mysql.
19/09/29 00:21:16 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
19/09/29 00:21:16 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
19/09/29 00:21:16 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
19/09/29 00:21:16 INFO mapreduce.ImportJobBase: Beginning import of employee
19/09/29 00:21:16 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
19/09/29 00:21:17 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
19/09/29 00:21:20 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
19/09/29 00:21:20 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
19/09/29 00:21:33 INFO db.DBInputFormat: Using read commited transaction isolation
19/09/29 00:21:33 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`id`), MAX(`id`) FROM `employee`
19/09/29 00:21:33 INFO db.IntegerSplitter: Split size: 1; Num splits: 4 from: 1 to: 6
19/09/29 00:21:33 INFO mapreduce.JobSubmitter: number of splits:4
19/09/29 00:21:35 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1569733481333_0003
19/09/29 00:21:37 INFO impl.YarnClientImpl: Submitted application application_1569733481333_0003
19/09/29 00:21:37 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1569733481333_0003/
19/09/29 00:21:37 INFO mapreduce.Job: Running job: job_1569733481333_0003
19/09/29 00:22:00 INFO mapreduce.Job: Job job_1569733481333_0003 running in uber mode : false
19/09/29 00:22:00 INFO mapreduce.Job: map 0% reduce 0%
19/09/29 00:22:38 INFO mapreduce.Job: map 25% reduce 0%
19/09/29 00:22:40 INFO mapreduce.Job: map 100% reduce 0%
19/09/29 00:22:42 INFO mapreduce.Job: Job job_1569733481333_0003 completed successfully
19/09/29 00:22:43 INFO mapreduce.Job: Counters: 30
	File System Counters
		FILE: Number of bytes read=0
		FILE: Number of bytes written=687212
		FILE: Number of read operations=0
		FILE: Number of large read operations=0
		FILE: Number of write operations=0
		HDFS: Number of bytes read=393
		HDFS: Number of bytes written=155

```
                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=4
                Total time spent by all maps in occupied slots (ms)=136303
                Total time spent by all reduces in occupied slots (ms)=0
                Total time spent by all map tasks (ms)=136303
                Total vcore-milliseconds taken by all map tasks=136303
                Total megabyte-milliseconds taken by all map tasks=139574272
        Map-Reduce Framework
                Map input records=6
                Map output records=6
                Input split bytes=393
                Spilled Records=0
                Failed Shuffles=0
                Merged Map outputs=0
                GC time elapsed (ms)=1370
                CPU time spent (ms)=12520
                Physical memory (bytes) snapshot=772935680
                Virtual memory (bytes) snapshot=6283862016
                Total committed heap usage (bytes)=582483968
        File Input Format Counters
                Bytes Read=0
        File Output Format Counters
                Bytes Written=155
19/09/29 00:22:43 INFO mapreduce.ImportJobBase: Transferred 155 bytes in 83.1543 seconds (1.864
bytes/sec)
19/09/29 00:22:43 INFO mapreduce.ImportJobBase: Retrieved 6 records.
19/09/29 00:22:43 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t
LIMIT 1
19/09/29 00:22:43 INFO hive.HiveImport: Loading uploaded data into Hive
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.0.jar!/hive-
log4j.properties
OK
Time taken: 12.093 seconds
Loading data to table default.employee
Table default.employee stats: [numFiles=5, numRows=0, totalSize=243, rawDataSize=0]
OK
Time taken: 4.27 seconds
```

//to check whether data is loaded in hive

```
[cloudera@quickstart ~]$ hive

Logging initialized using configuration in file:/etc/hive/conf.dist/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> select * from employee;
OK
1       zoya    20000   java
2       taniya  30000   android
3       pranali 40000   python
4       nazmeen         30000   python3
1       zoya    20000   java devel
2       pranali 30000   c++ develo
3       nazmeen         20000   python dev
4       taniya  40000   android de
5       vasim   20000   web develo
6       arshad  30000   web develo
Time taken: 4.205 seconds, Fetched: 10 row(s)
```