SQOOP

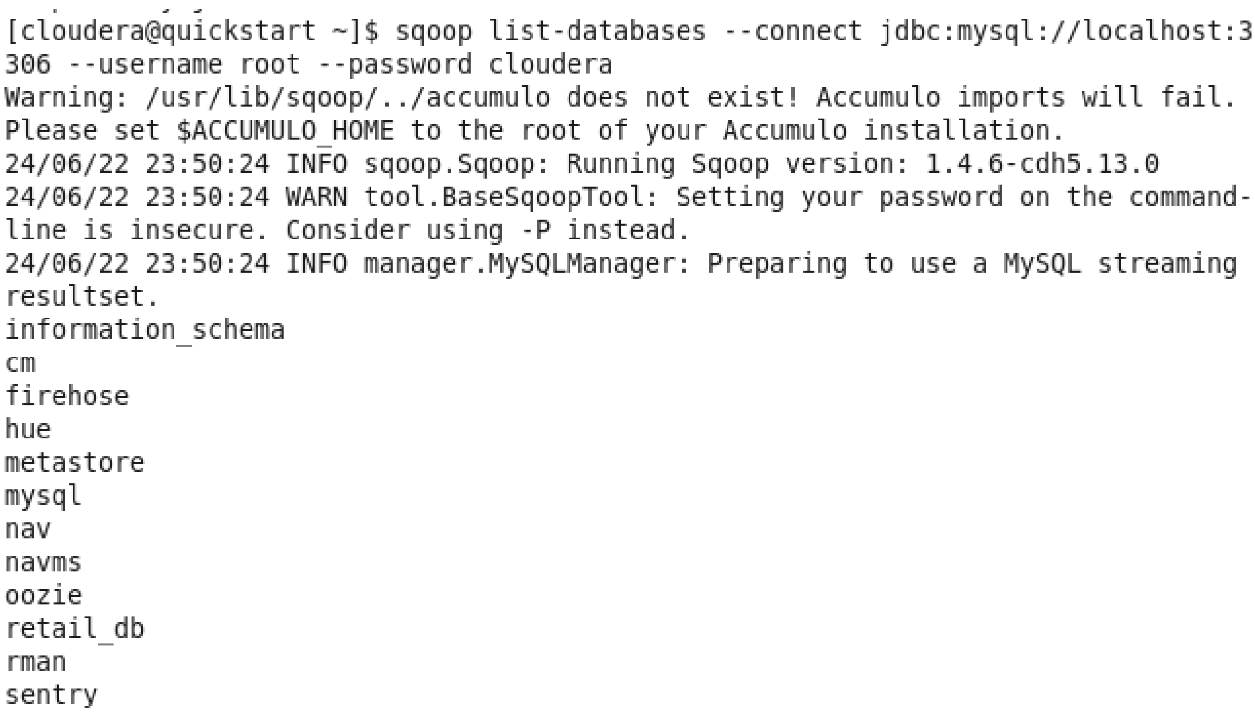
As RDBMS, we will be using MySQL database.

You can connect with MySQL database through the commands:

mysql -uroot -pcloudera

To list the databases in SQOOP:

sqoop list-databases --connect jdbc:mysql://localhost:3306 --username root --password cloudera



```
[cloudera@quickstart ~]$ sqoop list-databases --connect jdbc:mysql://localhost:3
306 --username root --password cloudera
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/22 23:50:24 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/06/22 23:50:24 WARN tool.BaseSqoopTool: Setting your password on the command-
line is insecure. Consider using -P instead.
24/06/22 23:50:24 INFO manager.MySQLManager: Preparing to use a MySQL streaming
resultset.
information_schema
cm
firehose
hue
metastore
mysql
nav
navms
oozie
retail_db
rman
sentry
```

Points to Remember:

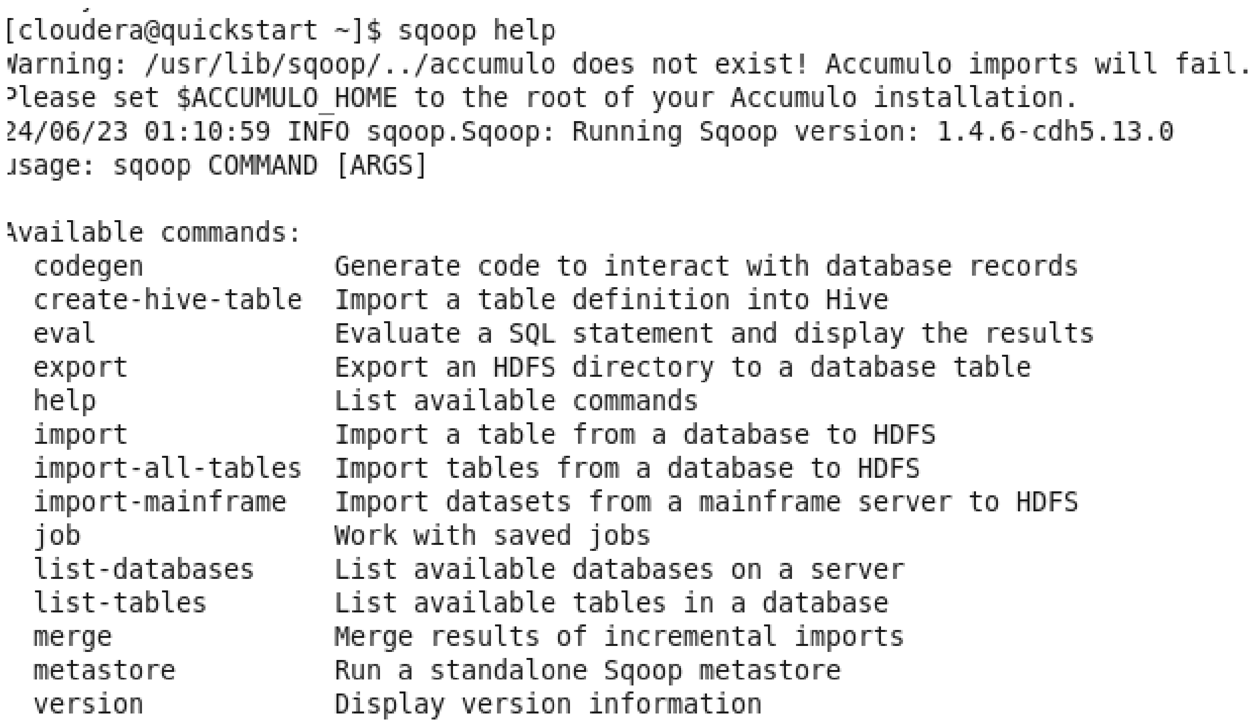
When we import the SQOOP:

1. Data is selected with Select command
2. Min and Max query is applied
3. Default no of splits : 4
4. Mapper and Reduce tasks gets executed

By Default : It will import the data in the user's home directory.
Eg: /user/cloudera

—---

All SQOOP commands:



```
[cloudera@quickstart ~]$ sqoop help
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/23 01:10:59 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
usage: sqoop COMMAND [ARGS]

Available commands:
  codegen            Generate code to interact with database records
  create-hive-table  Import a table definition into Hive
  eval               Evaluate a SQL statement and display the results
  export             Export an HDFS directory to a database table
  help               List available commands
  import             Import a table from a database to HDFS
  import-all-tables  Import tables from a database to HDFS
  import-mainframe   Import datasets from a mainframe server to HDFS
  job                Work with saved jobs
  list-databases     List available databases on a server
  list-tables        List available tables in a database
  merge              Merge results of incremental imports
  metastore          Run a standalone Sqoop metastore
  version            Display version information
```

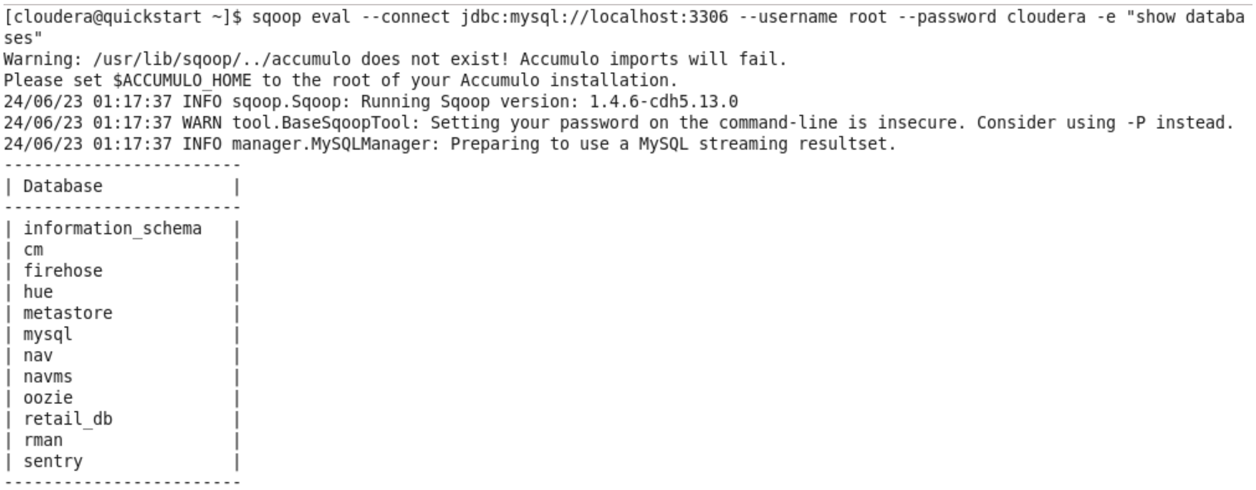
—---

****

SQOOP EVAL

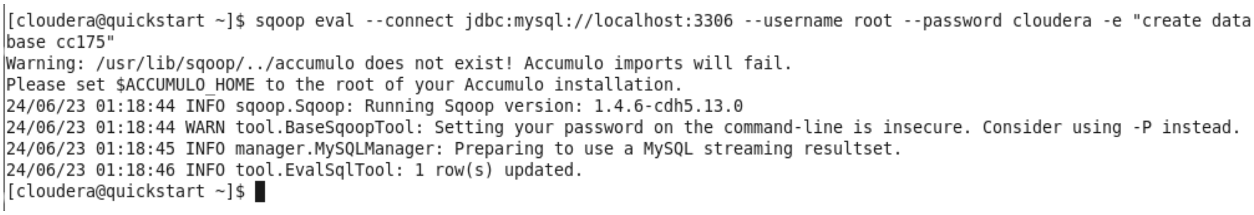
****

sqoop eval --connect jdbc:mysql://localhost:3306 --username root --password cloudera -e "show databases"



```
[cloudera@quickstart ~]$ sqoop eval --connect jdbc:mysql://localhost:3306 --username root --password cloudera -e "show databa
ses"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/23 01:17:37 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/06/23 01:17:37 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/06/23 01:17:37 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
------------------------
| Database             |
------------------------
| information_schema   |
| cm                   |
| firehose             |
| hue                  |
| metastore            |
| mysql                |
| nav                  |
| navms                |
| oozie                |
| retail_db            |
| rman                 |
| sentry               |
------------------------
```

sqoop eval --connect jdbc:mysql://localhost:3306 --username root --password cloudera -e "create database cc175"



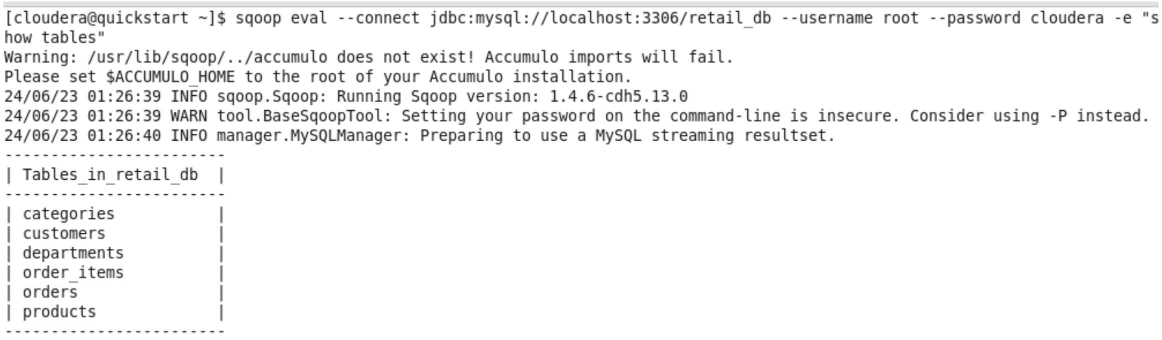
```
[cloudera@quickstart ~]$ sqoop eval --connect jdbc:mysql://localhost:3306 --username root --password cloudera -e "create data
base cc175"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/23 01:18:44 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/06/23 01:18:44 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/06/23 01:18:45 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
24/06/23 01:18:46 INFO tool.EvalSqlTool: 1 row(s) updated.
[cloudera@quickstart ~]$ █
```

Note 👏

It is not mandatory to start the Hadoop services, except for importing the data.

—---

sqoop eval --connect jdbc:mysql://localhost:3306/retail_db --username root --password cloudera -e "show tables"



```
[cloudera@quickstart ~]$ sqoop eval --connect jdbc:mysql://localhost:3306/retail_db --username root --password cloudera -e "s
how tables"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/23 01:26:39 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/06/23 01:26:39 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/06/23 01:26:40 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
-----------------------
| Tables_in_retail_db  |
-----------------------
| categories           |
| customers            |
| departments          |
| order_items          |
| orders               |
| products             |
-----------------------
```

—---

To store the connection username and password:

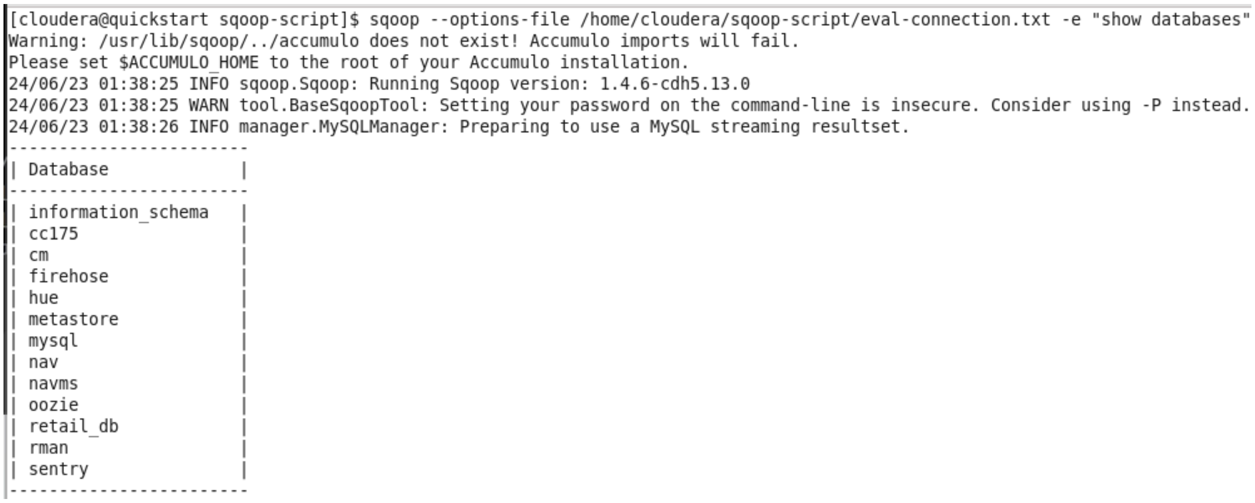
Approach 1:

We can create the files in which have the information such as below:

```
eval
--connect
jdbc:mysql://localhost:3306/retail_db
--username
root
--password
cloudera
```

And then can run the below commands:

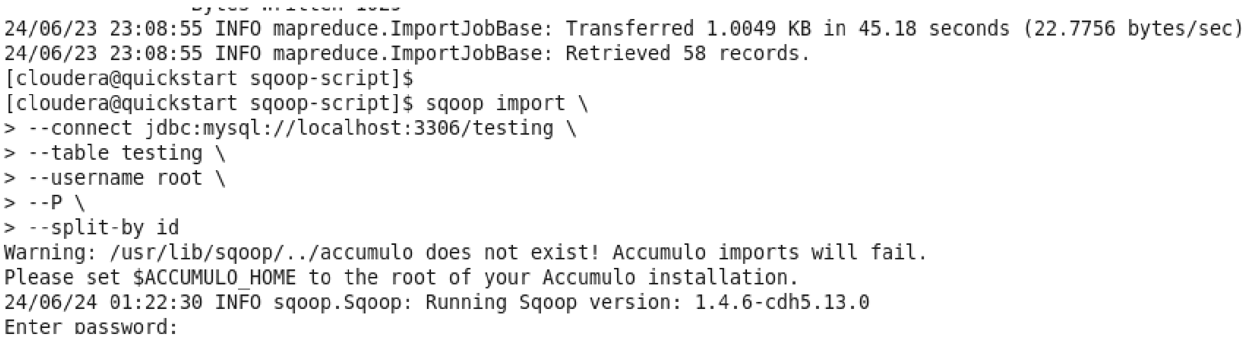
```
sqoop --options-file /home/cloudera/sqoop-script/eval-connection.txt -e "show databases"
```



```
[cloudera@quickstart sqoop-script]$ sqoop --options-file /home/cloudera/sqoop-script/eval-connection.txt -e "show databases"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/23 01:38:25 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/06/23 01:38:25 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/06/23 01:38:26 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
-----------------------
| Database            |
-----------------------
| information_schema  |
| cc175               |
| cm                  |
| firehose            |
| hue                 |
| metastore           |
| mysql               |
| nav                 |
| navms               |
| oozie               |
| retail_db           |
| rman                |
| sentry              |
-----------------------
```

Approach 2: (Manually asks to type the password.)

```
sqoop import \
--connect jdbc:mysql://localhost:3306/testing \
--table testing \
--username root \
--P \
--split-by id
```

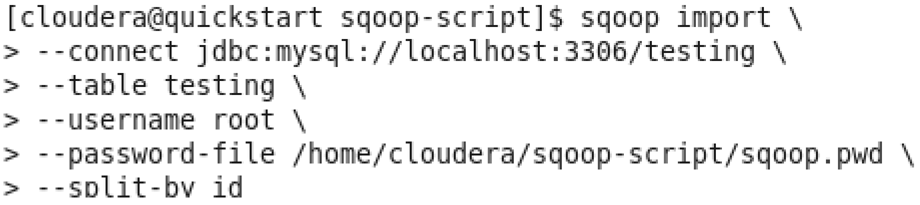


```
                Bytes Written 1029
24/06/23 23:08:55 INFO mapreduce.ImportJobBase: Transferred 1.0049 KB in 45.18 seconds (22.7756 bytes/sec)
24/06/23 23:08:55 INFO mapreduce.ImportJobBase: Retrieved 58 records.
[cloudera@quickstart sqoop-script]$
[cloudera@quickstart sqoop-script]$ sqoop import \
> --connect jdbc:mysql://localhost:3306/testing \
> --table testing \
> --username root \
> --P \
> --split-by id
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/24 01:22:30 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
```

Approach 3: Creating a password file

```
echo -n “cloudera” > sqoop.pwd
```

```
sqoop import \
--connect jdbc:mysql://localhost:3306/testing \
--table testing \
--username root \
--password-file file:///home/cloudera/sqoop-script/sqoop.pwd \
--split-by id
```



```
[cloudera@quickstart sqoop-script]$ sqoop import \
> --connect jdbc:mysql://localhost:3306/testing \
> --table testing \
> --username root \
> --password-file /home/cloudera/sqoop-script/sqoop.pwd \
> --split-by id
```

****

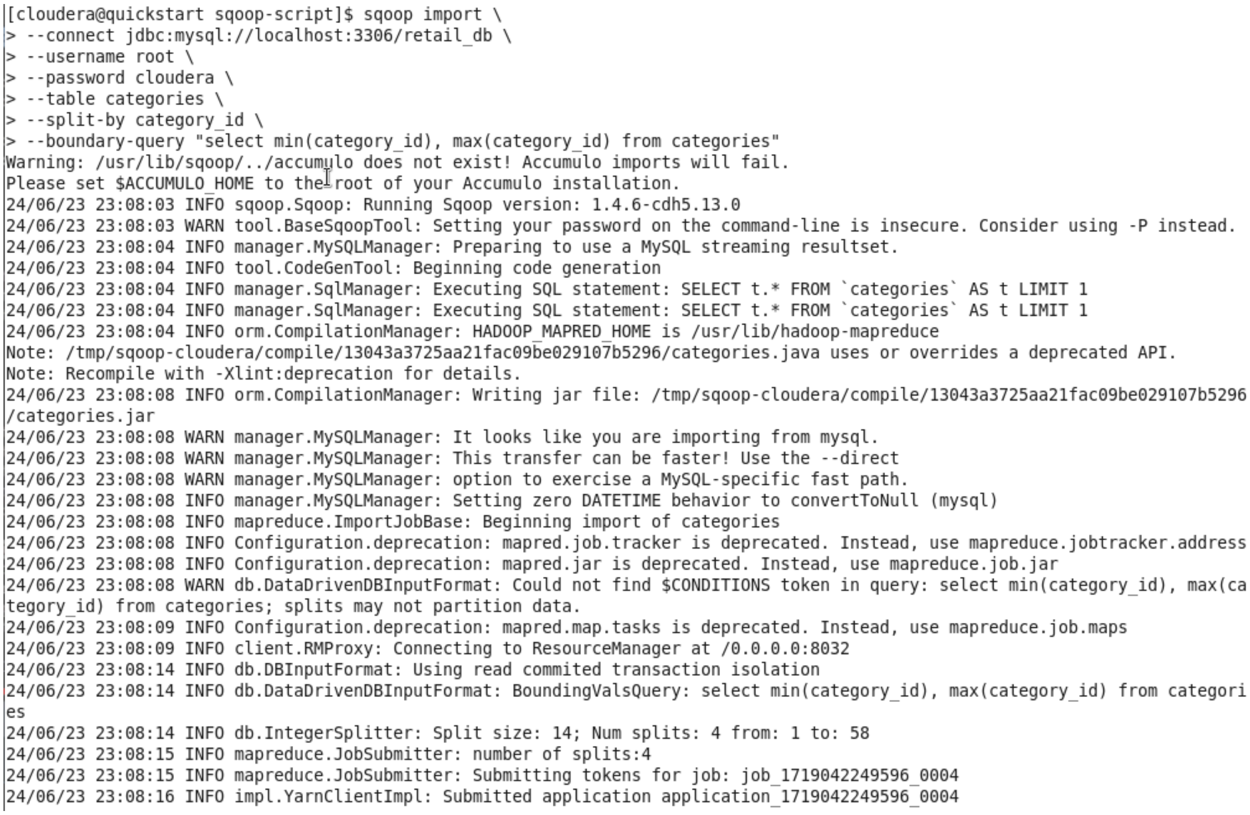
SQOOP IMPORT

****

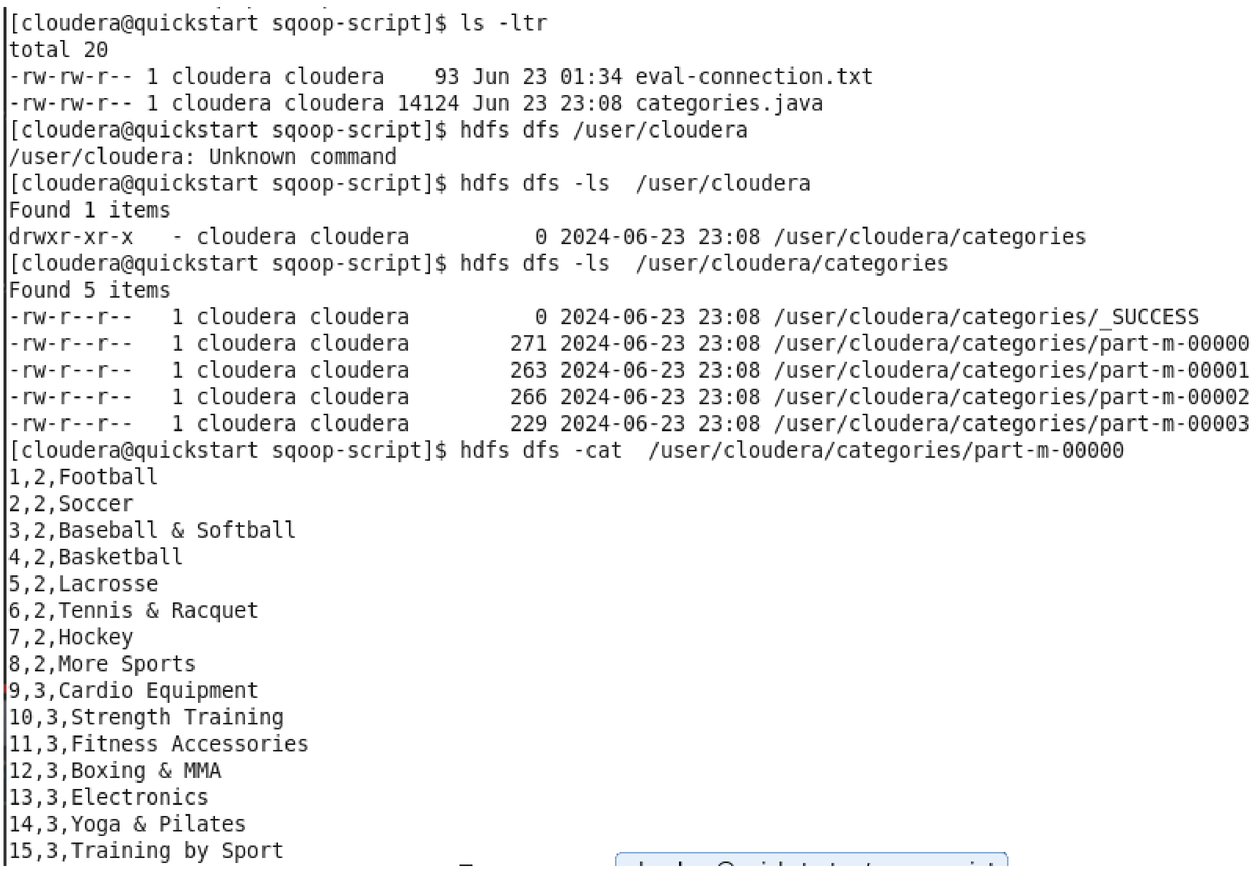
Boundary Query:

Boundary query is used to increase the performance.

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--password cloudera \
--table categories \
--split-by category_id \
--boundary-query "select min(category_id), max(category_id) from categories"
```



```
[cloudera@quickstart sqoop-script]$ sqoop import \
> --connect jdbc:mysql://localhost:3306/retail_db \
> --username root \
> --password cloudera \
> --table categories \
> --split-by category_id \
> --boundary-query "select min(category_id), max(category_id) from categories"
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/23 23:08:03 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
24/06/23 23:08:03 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
24/06/23 23:08:04 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
24/06/23 23:08:04 INFO tool.CodeGenTool: Beginning code generation
24/06/23 23:08:04 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `categories` AS t LIMIT 1
24/06/23 23:08:04 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `categories` AS t LIMIT 1
24/06/23 23:08:04 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/13043a3725aa21fac09be029107b5296/categories.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
24/06/23 23:08:08 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/13043a3725aa21fac09be029107b5296
/categories.jar
24/06/23 23:08:08 WARN manager.MySQLManager: It looks like you are importing from mysql.
24/06/23 23:08:08 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
24/06/23 23:08:08 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
24/06/23 23:08:08 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
24/06/23 23:08:08 INFO mapreduce.ImportJobBase: Beginning import of categories
24/06/23 23:08:08 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
24/06/23 23:08:08 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
24/06/23 23:08:08 WARN db.DataDrivenDBInputFormat: Could not find $CONDITIONS token in query: select min(category_id), max(ca
tegory_id) from categories; splits may not partition data.
24/06/23 23:08:09 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
24/06/23 23:08:09 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/06/23 23:08:14 INFO db.DBInputFormat: Using read commited transaction isolation
24/06/23 23:08:14 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: select min(category_id), max(category_id) from categori
es
24/06/23 23:08:14 INFO db.IntegerSplitter: Split size: 14; Num splits: 4 from: 1 to: 58
24/06/23 23:08:15 INFO mapreduce.JobSubmitter: number of splits:4
24/06/23 23:08:15 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1719042249596_0004
24/06/23 23:08:16 INFO impl.YarnClientImpl: Submitted application application_1719042249596_0004
```



```
[cloudera@quickstart sqoop-script]$ ls -ltr
total 20
-rw-rw-r-- 1 cloudera cloudera    93 Jun 23 01:34 eval-connection.txt
-rw-rw-r-- 1 cloudera cloudera 14124 Jun 23 23:08 categories.java
[cloudera@quickstart sqoop-script]$ hdfs dfs /user/cloudera
/user/cloudera: Unknown command
[cloudera@quickstart sqoop-script]$ hdfs dfs -ls  /user/cloudera
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2024-06-23 23:08 /user/cloudera/categories
[cloudera@quickstart sqoop-script]$ hdfs dfs -ls  /user/cloudera/categories
Found 5 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-23 23:08 /user/cloudera/categories/_SUCCESS
-rw-r--r--   1 cloudera cloudera        271 2024-06-23 23:08 /user/cloudera/categories/part-m-00000
-rw-r--r--   1 cloudera cloudera        263 2024-06-23 23:08 /user/cloudera/categories/part-m-00001
-rw-r--r--   1 cloudera cloudera        266 2024-06-23 23:08 /user/cloudera/categories/part-m-00002
-rw-r--r--   1 cloudera cloudera        229 2024-06-23 23:08 /user/cloudera/categories/part-m-00003
[cloudera@quickstart sqoop-script]$ hdfs dfs -cat  /user/cloudera/categories/part-m-00000
1,2,Football
2,2,Soccer
3,2,Baseball & Softball
4,2,Basketball
5,2,Lacrosse
6,2,Tennis & Racquet
7,2,Hockey
8,2,More Sports
9,3,Cardio Equipment
10,3,Strength Training
11,3,Fitness Accessories
12,3,Boxing & MMA
13,3,Electronics
14,3,Yoga & Pilates
15,3,Training by Sport
```

—---

Speed up Transfers: Direct Import:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table categories \
--direct
```

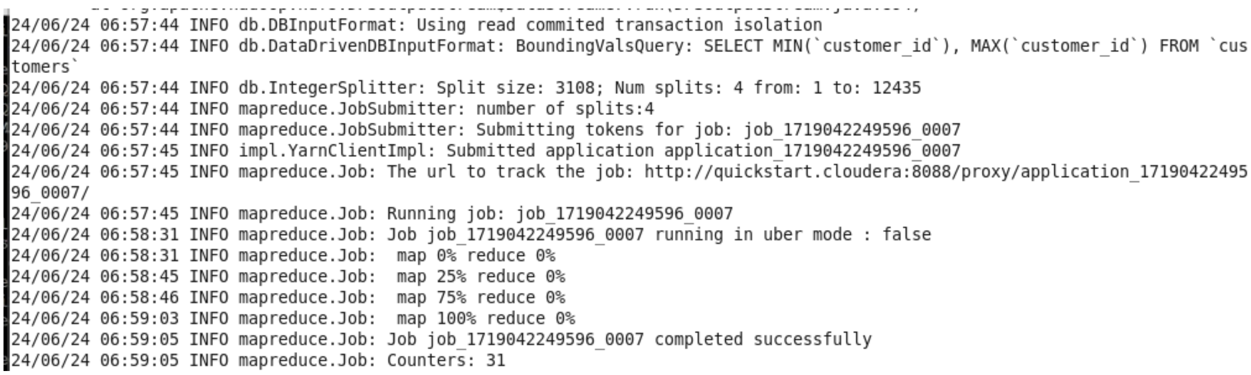
Note:

1. Sqoop can only perform --direct mode imports from PostgreSQL,Oracle and Netezza.
2. Binary formats like sequence file or Avro wont work with direct mode import.
3. In case of MySQL, when using --direct parameters, sqoop will takes advantages of MySQL native utility like mysqldump and mysqlimport, rather than using JDBC interface for transferring data.

—---

Target dir import:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table customers \
--target-dir /user/cloudera/customers
```

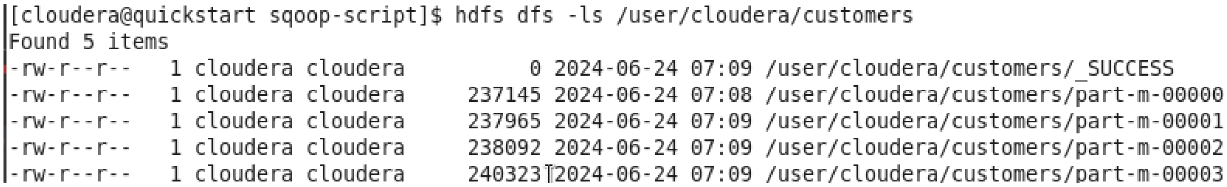


```
24/06/24 06:57:44 INFO db.DBInputFormat: Using read commited transaction isolation
24/06/24 06:57:44 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`customer_id`), MAX(`customer_id`) FROM `cus
tomers`
24/06/24 06:57:44 INFO db.IntegerSplitter: Split size: 3108; Num splits: 4 from: 1 to: 12435
24/06/24 06:57:44 INFO mapreduce.JobSubmitter: number of splits:4
24/06/24 06:57:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1719042249596_0007
24/06/24 06:57:45 INFO impl.YarnClientImpl: Submitted application application_1719042249596_0007
24/06/24 06:57:45 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_17190422495
96_0007/
24/06/24 06:57:45 INFO mapreduce.Job: Running job: job_1719042249596_0007
24/06/24 06:58:31 INFO mapreduce.Job: Job job_1719042249596_0007 running in uber mode : false
24/06/24 06:58:31 INFO mapreduce.Job:  map 0% reduce 0%
24/06/24 06:58:45 INFO mapreduce.Job:  map 25% reduce 0%
24/06/24 06:58:46 INFO mapreduce.Job:  map 75% reduce 0%
24/06/24 06:59:03 INFO mapreduce.Job:  map 100% reduce 0%
24/06/24 06:59:05 INFO mapreduce.Job: Job job_1719042249596_0007 completed successfully
24/06/24 06:59:05 INFO mapreduce.Job: Counters: 31
```

—--

Delete the target dir and overwrite it:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table customers \
--delete-target-dir \
--target-dir /user/cloudera/customers
```

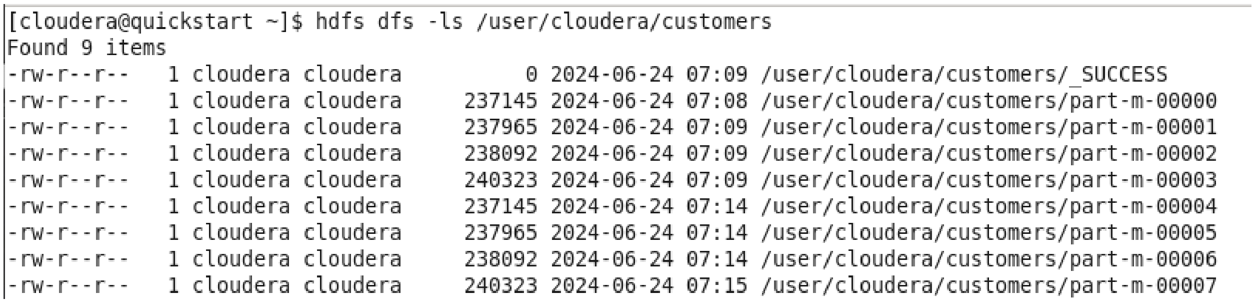


```
[cloudera@quickstart sqoop-script]$ hdfs dfs -ls /user/cloudera/customers
Found 5 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 07:09 /user/cloudera/customers/_SUCCESS
-rw-r--r--   1 cloudera cloudera     237145 2024-06-24 07:08 /user/cloudera/customers/part-m-00000
-rw-r--r--   1 cloudera cloudera     237965 2024-06-24 07:09 /user/cloudera/customers/part-m-00001
-rw-r--r--   1 cloudera cloudera     238092 2024-06-24 07:09 /user/cloudera/customers/part-m-00002
-rw-r--r--   1 cloudera cloudera     240323 2024-06-24 07:09 /user/cloudera/customers/part-m-00003
```

—--

Append data to an existing directory:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table customers \
--target-dir /user/cloudera/customers \
--append
```

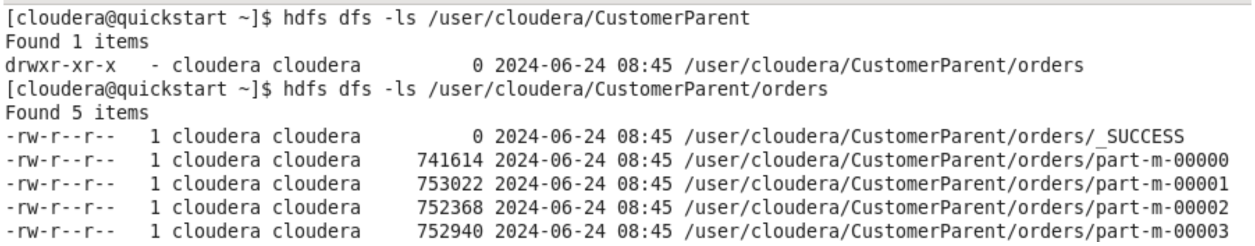


```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/customers
Found 9 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 07:09 /user/cloudera/customers/_SUCCESS
-rw-r--r--   1 cloudera cloudera     237145 2024-06-24 07:08 /user/cloudera/customers/part-m-00000
-rw-r--r--   1 cloudera cloudera     237965 2024-06-24 07:09 /user/cloudera/customers/part-m-00001
-rw-r--r--   1 cloudera cloudera     238092 2024-06-24 07:09 /user/cloudera/customers/part-m-00002
-rw-r--r--   1 cloudera cloudera     240323 2024-06-24 07:09 /user/cloudera/customers/part-m-00003
-rw-r--r--   1 cloudera cloudera     237145 2024-06-24 07:14 /user/cloudera/customers/part-m-00004
-rw-r--r--   1 cloudera cloudera     237965 2024-06-24 07:14 /user/cloudera/customers/part-m-00005
-rw-r--r--   1 cloudera cloudera     238092 2024-06-24 07:14 /user/cloudera/customers/part-m-00006
-rw-r--r--   1 cloudera cloudera     240323 2024-06-24 07:15 /user/cloudera/customers/part-m-00007
```

—--

Importing the data inside parent directory:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table orders \
--warehouse-dir /user/cloudera/CustomerParent
```

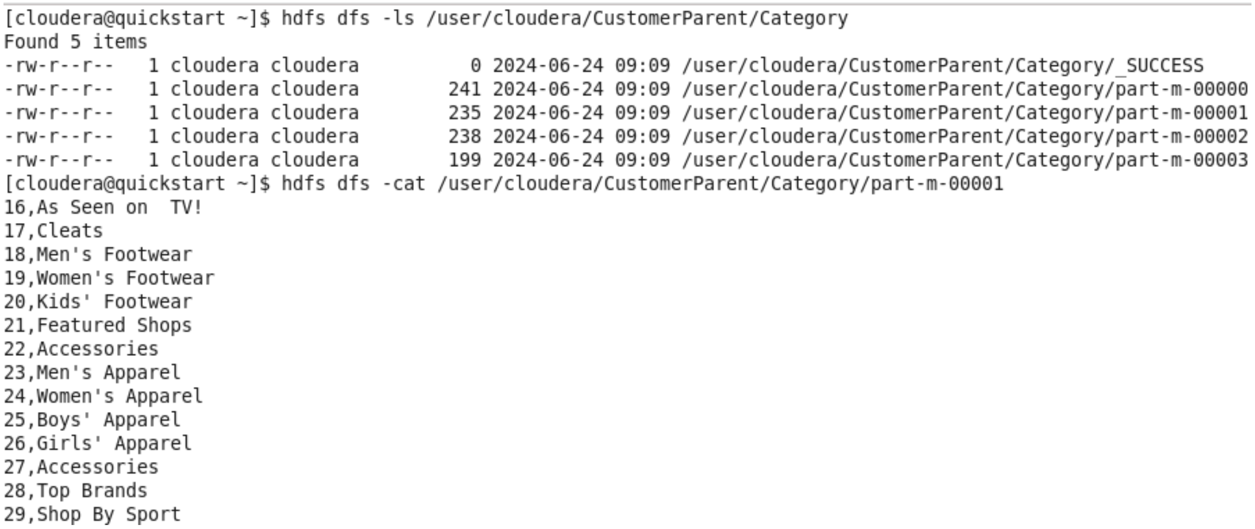


```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/CustomerParent
Found 1 items
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 08:45 /user/cloudera/CustomerParent/orders
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/CustomerParent/orders
Found 5 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 08:45 /user/cloudera/CustomerParent/orders/_SUCCESS
-rw-r--r--   1 cloudera cloudera     741614 2024-06-24 08:45 /user/cloudera/CustomerParent/orders/part-m-00000
-rw-r--r--   1 cloudera cloudera     753022 2024-06-24 08:45 /user/cloudera/CustomerParent/orders/part-m-00001
-rw-r--r--   1 cloudera cloudera     752368 2024-06-24 08:45 /user/cloudera/CustomerParent/orders/part-m-00002
-rw-r--r--   1 cloudera cloudera     752940 2024-06-24 08:45 /user/cloudera/CustomerParent/orders/part-m-00003
```

—--

Import all rows of a table from MySQL, but specific columns of table:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--table categories \
--username root \
--P \
--target-dir /user/cloudera/CustomerParent \
--columns "category_id,category_name"
```

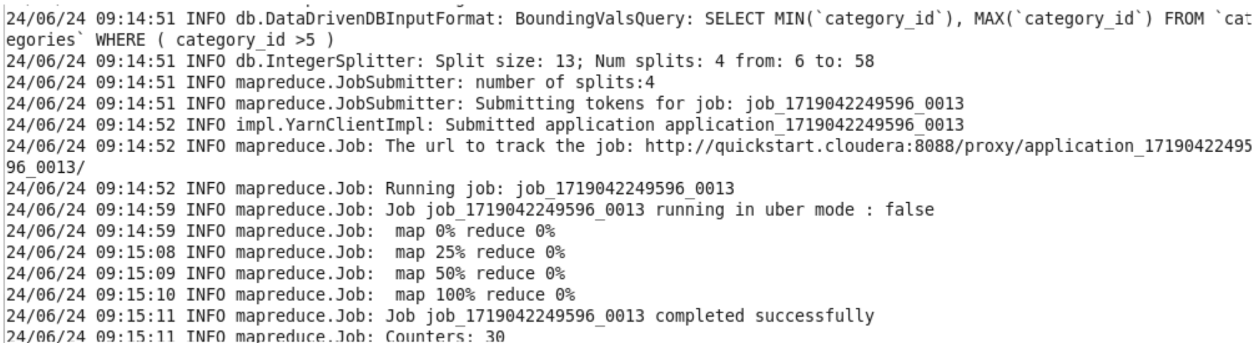


```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/CustomerParent/Category
Found 5 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 09:09 /user/cloudera/CustomerParent/Category/_SUCCESS
-rw-r--r--   1 cloudera cloudera        241 2024-06-24 09:09 /user/cloudera/CustomerParent/Category/part-m-00000
-rw-r--r--   1 cloudera cloudera        235 2024-06-24 09:09 /user/cloudera/CustomerParent/Category/part-m-00001
-rw-r--r--   1 cloudera cloudera        238 2024-06-24 09:09 /user/cloudera/CustomerParent/Category/part-m-00002
-rw-r--r--   1 cloudera cloudera        199 2024-06-24 09:09 /user/cloudera/CustomerParent/Category/part-m-00003
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/CustomerParent/Category/part-m-00001
16,As Seen on  TV!
17,Cleats
18,Men's Footwear
19,Women's Footwear
20,Kids' Footwear
21,Featured Shops
22,Accessories
23,Men's Apparel
24,Women's Apparel
25,Boys' Apparel
26,Girls' Apparel
27,Accessories
28,Top Brands
29,Shop By Sport
```

—---

Use WHERE clause:

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--table categories \
--username root \
--P \
--target-dir /user/cloudera/CustomerParent1 \
--columns "category_id,category_name" \
--where "category_id >5"
```



```
24/06/24 09:14:51 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`category_id`), MAX(`category_id`) FROM `cat
egories` WHERE ( category_id >5 )
24/06/24 09:14:51 INFO db.IntegerSplitter: Split size: 13; Num splits: 4 from: 6 to: 58
24/06/24 09:14:51 INFO mapreduce.JobSubmitter: number of splits:4
24/06/24 09:14:51 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1719042249596_0013
24/06/24 09:14:52 INFO impl.YarnClientImpl: Submitted application application_1719042249596_0013
24/06/24 09:14:52 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_17190422495
96_0013/
24/06/24 09:14:52 INFO mapreduce.Job: Running job: job_1719042249596_0013
24/06/24 09:14:59 INFO mapreduce.Job: Job job_1719042249596_0013 running in uber mode : false
24/06/24 09:14:59 INFO mapreduce.Job:  map 0% reduce 0%
24/06/24 09:15:08 INFO mapreduce.Job:  map 25% reduce 0%
24/06/24 09:15:09 INFO mapreduce.Job:  map 50% reduce 0%
24/06/24 09:15:10 INFO mapreduce.Job:  map 100% reduce 0%
24/06/24 09:15:11 INFO mapreduce.Job: Job job_1719042249596_0013 completed successfully
24/06/24 09:15:11 INFO mapreduce.Job: Counters: 30
```

—---

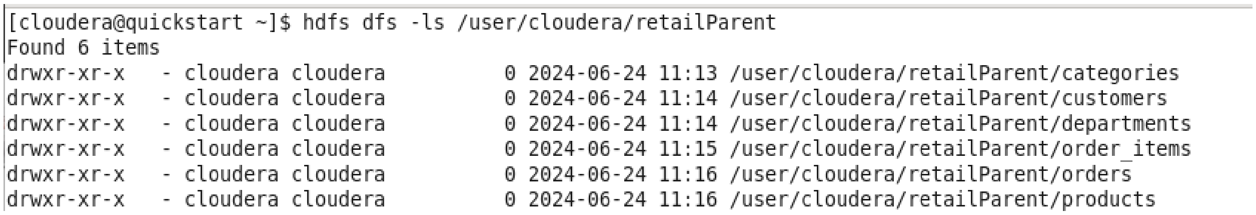
Import all tables of MySQL DB into HDFS 🙂

--target-dir parameter is not allowed
--warehouse-dir parameter is allowed

Note:

1. Each table must have a single column primary key or --autoreset-to-one-mapper option must be used.
2. We must intend to import all columns of each table
   (it means we cannot use --columns)

sqoop import-all-tables \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--warehouse-dir /user/cloudera/retailParent \
--autoreset-to-one-mapper



```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/retailParent
Found 6 items
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 11:13 /user/cloudera/retailParent/categories
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 11:14 /user/cloudera/retailParent/customers
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 11:14 /user/cloudera/retailParent/departments
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 11:15 /user/cloudera/retailParent/order_items
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 11:16 /user/cloudera/retailParent/orders
drwxr-xr-x   - cloudera cloudera          0 2024-06-24 11:16 /user/cloudera/retailParent/products
```

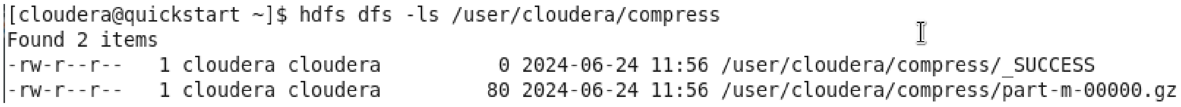
sqoop import-all-tables \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--warehouse-dir /user/cloudera/retailParent_exclude \
--exclude-tables "departments"

—---

Compressing the imported data:

Syntax 1: gzip (By default)

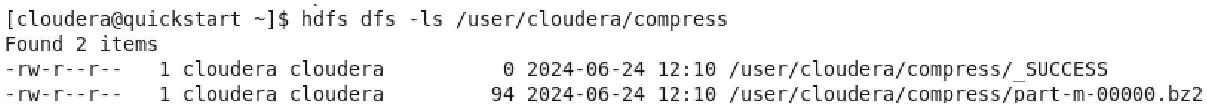
```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table "departments" \
--m 1 \
--compress \
--target-dir /user/cloudera/compress
```



```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/compress
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 11:56 /user/cloudera/compress/_SUCCESS
-rw-r--r--   1 cloudera cloudera         80 2024-06-24 11:56 /user/cloudera/compress/part-m-00000.gz
```

Syntax 2: Bzip2

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments \
--m 1 \
--compress --compression-codec org.apache.hadoop.io.compress.BZip2Codec \
--target-dir /user/cloudera/compress
```



```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/compress
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 12:10 /user/cloudera/compress/_SUCCESS
-rw-r--r--   1 cloudera cloudera         94 2024-06-24 12:10 /user/cloudera/compress/part-m-00000.bz2
```

—---

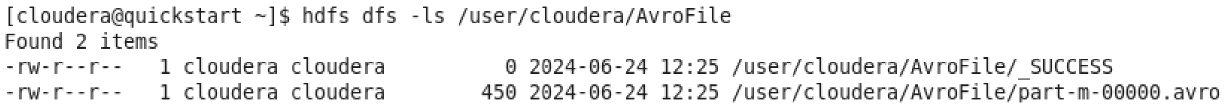
Import MySQL data into HDFS in various file format:

Syntax 1: (SEQUENCE file format)

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments \
--m 1 \
--target-dir /user/cloudera/compress \
--as-sequencefile
```

Syntax 2: (AVRO DATA FILE)

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments \
--m 1 \
--target-dir /user/cloudera/AvroFile \
--as-avrodatafile
```



```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/AvroFile
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-24 12:25 /user/cloudera/AvroFile/_SUCCESS
-rw-r--r--   1 cloudera cloudera        450 2024-06-24 12:25 /user/cloudera/AvroFile/part-m-00000.avro
```

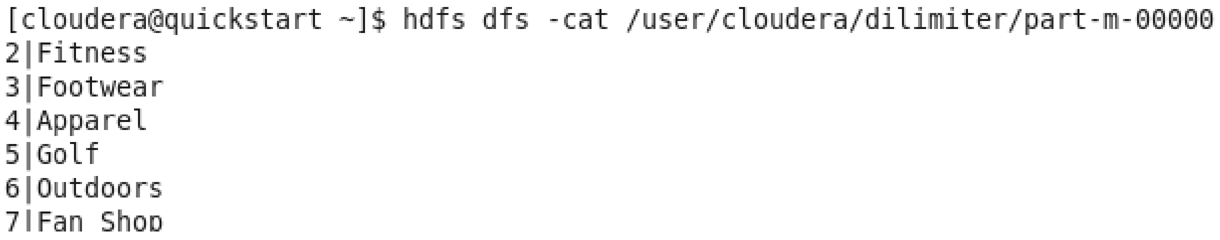
—---

Delimiter 👍

Note:

1. The file format is by default textfile
2. By default delimiter is taken as ‘,’

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments \
--m 1 \
--target-dir /user/cloudera/dilimiter \
--fields-terminated-by '|'
```



```
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/dilimiter/part-m-00000
2|Fitness
3|Footwear
4|Apparel
5|Golf
6|Outdoors
7|Fan Shop
```

—---

Incremental Import:

```
create table inc_imp (id int, name varchar(15, city varchar(15 ));
```

—---

```
INSERT INTO inc_imp VALUES (1, 'Komal', 'Mumbai');
INSERT INTO inc_imp VALUES (2, 'Komi', 'Ireland');
INSERT INTO inc_imp VALUES (3, 'Tadano', 'Ireland');
```

—---

```
sqoop import \
--connect jdbc:mysql://localhost:3306/testing \
--username root \
--P \
--table inc_imp \
--m 1 \
--target-dir /user/cloudera/incremental_import
```

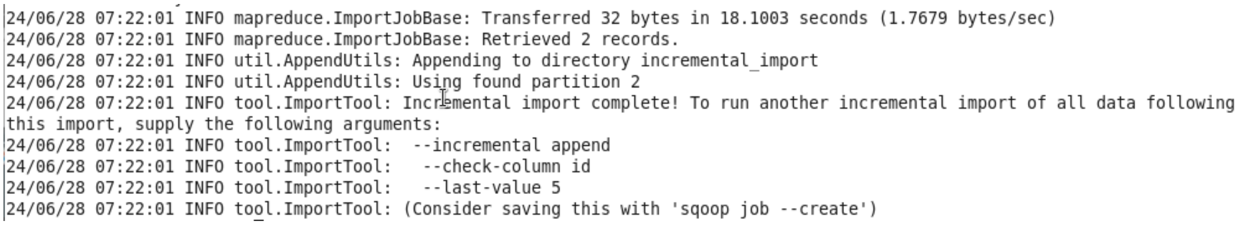
—--

```
INSERT INTO inc_imp VALUES (4, 'Kom', 'Mumbai');
INSERT INTO inc_imp VALUES (5, 'Komi-san', 'Ireland');
```

—--

(Hard-coded values 👍)

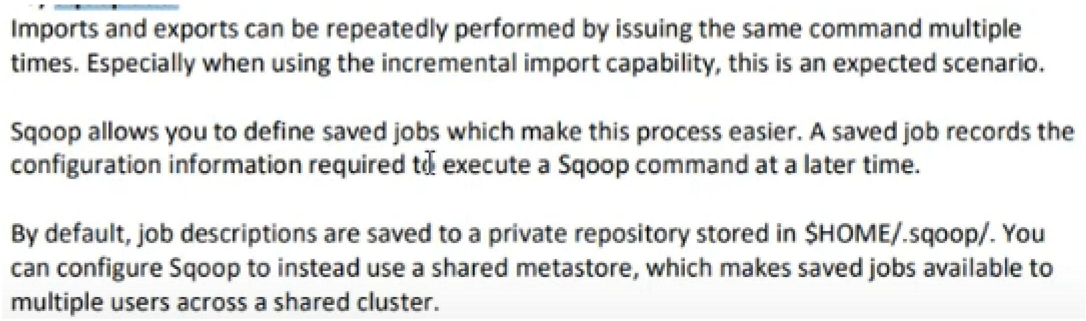
```
sqoop import \
--connect jdbc:mysql://localhost:3306/testing \
--username root \
--P \
--table inc_imp \
--m 1 \
--target-dir /user/cloudera/incremental_import \
--incremental append --check-column id --last-value 3
```



```
24/06/28 07:22:01 INFO mapreduce.ImportJobBase: Transferred 32 bytes in 18.1003 seconds (1.7679 bytes/sec)
24/06/28 07:22:01 INFO mapreduce.ImportJobBase: Retrieved 2 records.
24/06/28 07:22:01 INFO util.AppendUtils: Appending to directory incremental_import
24/06/28 07:22:01 INFO util.AppendUtils: Using found partition 2
24/06/28 07:22:01 INFO tool.ImportTool: Incremental import complete! To run another incremental import of all data following
this import, supply the following arguments:
24/06/28 07:22:01 INFO tool.ImportTool:  --incremental append
24/06/28 07:22:01 INFO tool.ImportTool:   --check-column id
24/06/28 07:22:01 INFO tool.ImportTool:   --last-value 5
24/06/28 07:22:01 INFO tool.ImportTool: (Consider saving this with 'sqoop job --create')
```

—---

Sqoop Job:



Imports and exports can be repeatedly performed by issuing the same command multiple times. Especially when using the incremental import capability, this is an expected scenario.

Sqoop allows you to define saved jobs which make this process easier. A saved job records the configuration information required to execute a Sqoop command at a later time.

By default, job descriptions are saved to a private repository stored in $HOME/.sqoop/. You can configure Sqoop to instead use a shared metastore, which makes saved jobs available to multiple users across a shared cluster.

```
sqoop job --create inc_imp_id2 \
-- import --connect jdbc:mysql://localhost:3306/testing \
--username root \
--P \
--table inc_imp \
--m 1 \
--target-dir /user/cloudera/incremental_import_id2 \
--incremental append --check-column id --last-value 0
```

—--

```
sqoop job --exec inc_imp_id2
```

—--

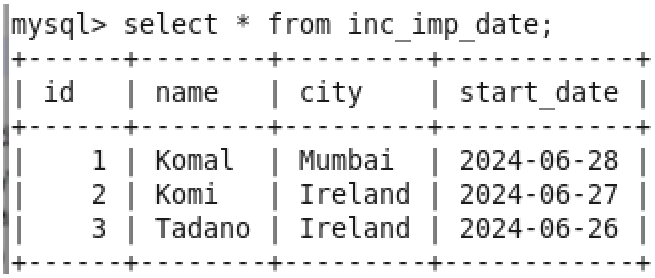
```
sqoop job --show inc_imp_id | grep ‘incremental.last.value’
```

—--

Incremental import by date:

```
create table inc_imp_date (id int, name varchar(15), city varchar(15 ),start_date date);

INSERT INTO inc_imp VALUES (1, 'Komal', 'Mumbai',now()-interval 1 day);
INSERT INTO inc_imp VALUES (2, 'Komi', 'Ireland', now()- interval 2 day);
INSERT INTO inc_imp VALUES (3, 'Tadano', 'Ireland', now()-interval 3 day);
```



```
mysql> select * from inc_imp_date;
+------+--------+---------+------------+
| id   | name   | city    | start_date |
+------+--------+---------+------------+
|    1 | Komal  | Mumbai  | 2024-06-28 |
|    2 | Komi   | Ireland | 2024-06-27 |
|    3 | Tadano | Ireland | 2024-06-26 |
+------+--------+---------+------------+
```

```
sqoop job --create inc_imp_dt \
-- import --connect jdbc:mysql://localhost:3306/testing \
--username root \
--P \
--table inc_imp_date \
--m 1 \
--target-dir /user/cloudera/incremental_import_id_date \
--incremental append --check-column start_date --last-value 0000-00-00
```

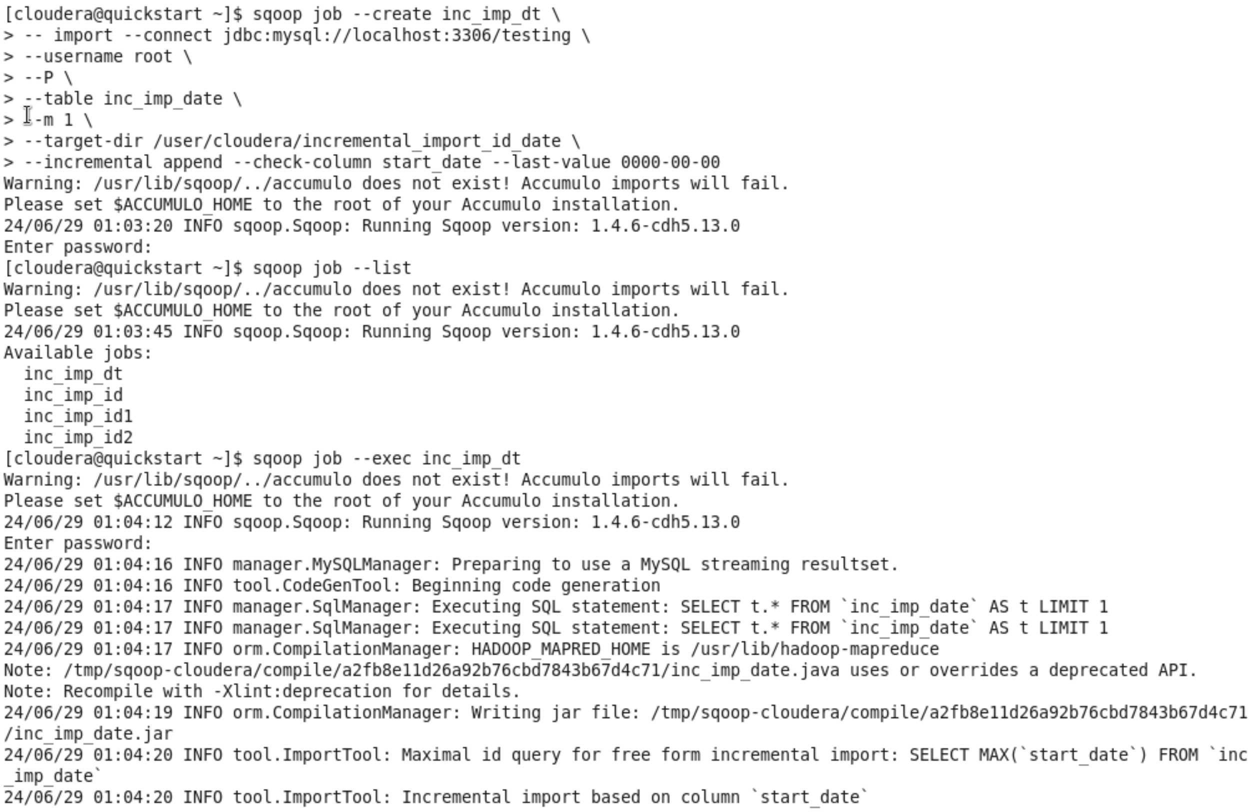
—--

```
sqoop job --list
```

—---

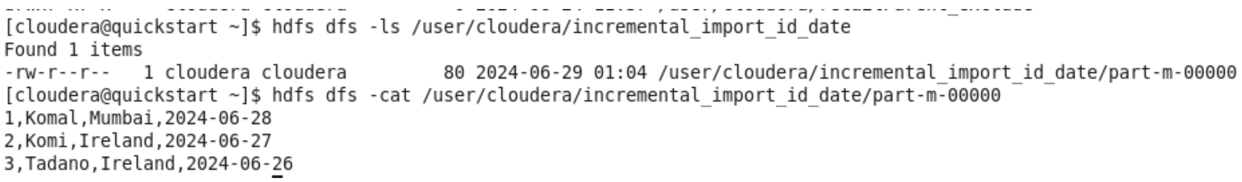
```
sqoop job --exec inc_imp_dt
```

—---



```
[cloudera@quickstart ~]$ sqoop job --create inc_imp_dt \
> -- import --connect jdbc:mysql://localhost:3306/testing \
> --username root \
> --P \
> --table inc_imp_date \
> --m 1 \
> --target-dir /user/cloudera/incremental_import_id_date \
> --incremental append --check-column start_date --last-value 0000-00-00
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/29 01:03:20 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
[cloudera@quickstart ~]$ sqoop job --list
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/29 01:03:45 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Available jobs:
  inc_imp_dt
  inc_imp_id
  inc_imp_id1
  inc_imp_id2
[cloudera@quickstart ~]$ sqoop job --exec inc_imp_dt
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
24/06/29 01:04:12 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
Enter password:
24/06/29 01:04:16 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
24/06/29 01:04:16 INFO tool.CodeGenTool: Beginning code generation
24/06/29 01:04:17 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `inc_imp_date` AS t LIMIT 1
24/06/29 01:04:17 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `inc_imp_date` AS t LIMIT 1
24/06/29 01:04:17 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/a2fb8e11d26a92b76cbd7843b67d4c71/inc_imp_date.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
24/06/29 01:04:19 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/a2fb8e11d26a92b76cbd7843b67d4c71
/inc_imp_date.jar
24/06/29 01:04:20 INFO tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`start_date`) FROM `inc
_imp_date`
24/06/29 01:04:20 INFO tool.ImportTool: Incremental import based on column `start_date`
```

—---



```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/incremental_import_id_date
Found 1 items
-rw-r--r--   1 cloudera cloudera         80 2024-06-29 01:04 /user/cloudera/incremental_import_id_date/part-m-00000
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/incremental_import_id_date/part-m-00000
1,Komal,Mumbai,2024-06-28
2,Komi,Ireland,2024-06-27
3,Tadano,Ireland,2024-06-26
```

—---

Add 2 new row:

INSERT INTO inc_imp_date VALUES (4, 'Kom', 'Mumbai', now());
INSERT INTO inc_imp_date VALUES (5, 'Komi-san', 'Ireland',now());

—---

sqoop job --exec inc_imp_dt

—---

24/06/29 01:15:32 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/8b9e98d080ae505666e98727f95a6531/inc_imp_date.jar
24/06/29 01:15:34 INFO tool.ImportTool: Maximal id query for free form incremental import: SELECT MAX(`start_date`) FROM `inc_imp_date`
24/06/29 01:15:34 INFO tool.ImportTool: Incremental import based on column `start_date`
24/06/29 01:15:34 INFO tool.ImportTool: Lower bound value: '2024-06-28'

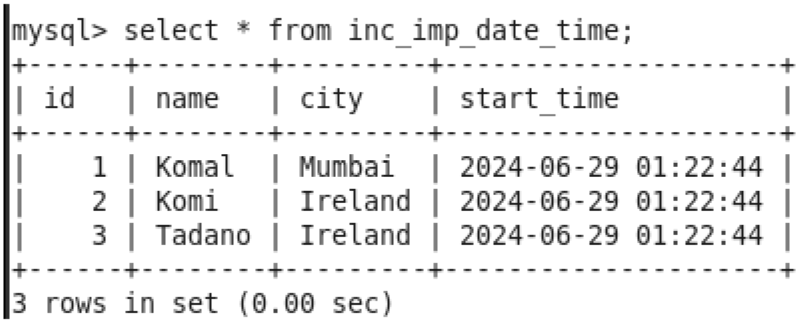
```
24/06/29 01:15:34 INFO tool.ImportTool: Upper bound value: '2024-06-29'
24/06/29 01:15:34 WARN manager.MySQLManager: It looks like you are importing from mysql.
24/06/29 01:15:34 WARN manager.MySQLManager: This transfer can be faster! Use the
--direct
24/06/29 01:15:34 WARN manager.MySQLManager: option to exercise a MySQL-specific fast
path.
24/06/29 01:15:34 INFO manager.MySQLManager: Setting zero DATETIME behavior to
convertToNull (mysql)
24/06/29 01:15:34 INFO mapreduce.ImportJobBase: Beginning import of inc_imp_date
24/06/29 01:15:34 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead,
use mapreduce.jobtracker.address
24/06/29 01:15:34 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use
mapreduce.job.jar
24/06/29 01:15:34 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead,
use mapreduce.job.maps
24/06/29 01:15:34 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/06/29 01:16:55 INFO db.DBInputFormat: Using read commited transaction isolation
24/06/29 01:16:55 INFO mapreduce.JobSubmitter: number of splits:1
24/06/29 01:16:55 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1719042249596_0041
24/06/29 01:16:56 INFO impl.YarnClientImpl: Submitted application
application_1719042249596_0041
24/06/29 01:16:56 INFO mapreduce.Job: The url to track the job:
http://quickstart.cloudera:8088/proxy/application_1719042249596_0041/
24/06/29 01:16:56 INFO mapreduce.Job: Running job: job_1719042249596_0041
24/06/29 01:17:04 INFO mapreduce.Job: Job job_1719042249596_0041 running in uber mode :
false
24/06/29 01:17:04 INFO mapreduce.Job:  map 0% reduce 0%
24/06/29 01:17:10 INFO mapreduce.Job:  map 100% reduce 0%
24/06/29 01:17:11 INFO mapreduce.Job: Job job_1719042249596_0041 completed
successfully
24/06/29 01:17:11 INFO mapreduce.Job: Counters: 30
```

—---

Incremental import lastmodified 😣

```
create table inc_imp_date_time (id int, name varchar(15), city varchar(15 ),start_time
timestamp);

INSERT INTO inc_imp_date_time VALUES (1, 'Komal', 'Mumbai',now());
INSERT INTO inc_imp_date_time VALUES (2, 'Komi', 'Ireland', now());
INSERT INTO inc_imp_date_time VALUES (3, 'Tadano', 'Ireland', now());
```

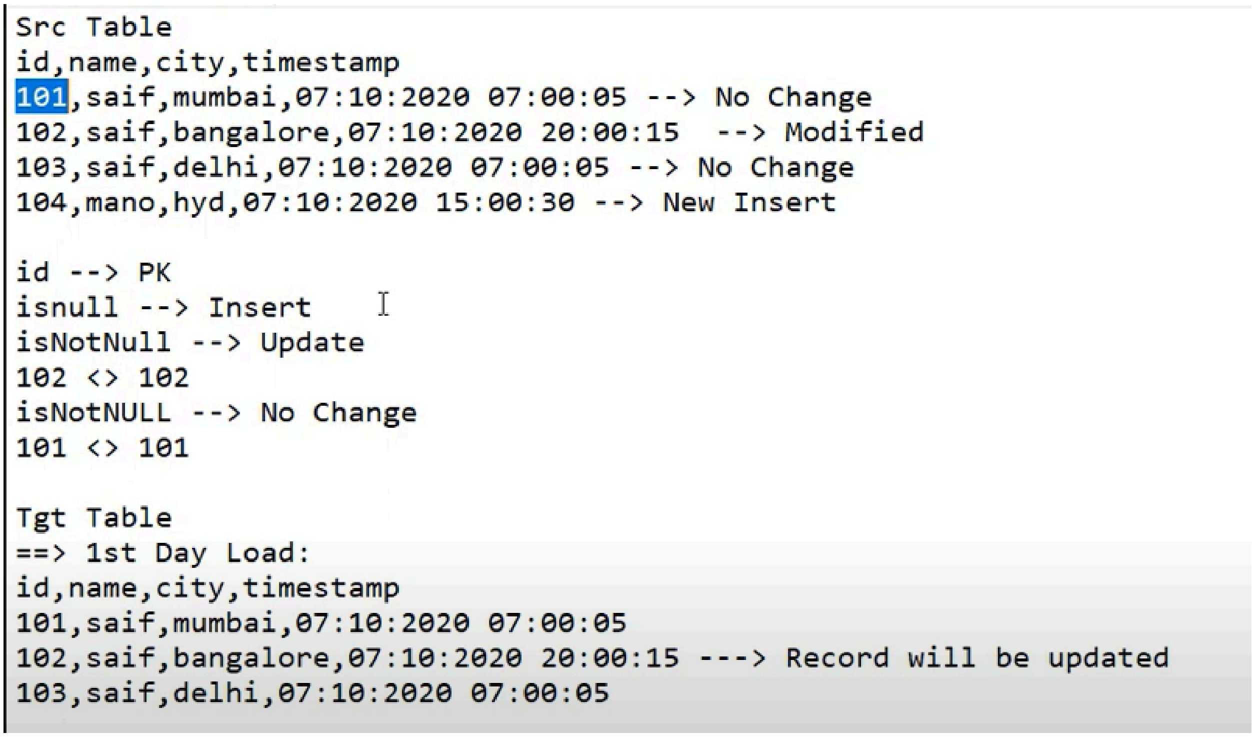


```
mysql> select * from inc_imp_date_time;
+------+--------+---------+---------------------+
| id   | name   | city    | start_time          |
+------+--------+---------+---------------------+
|    1 | Komal  | Mumbai  | 2024-06-29 01:22:44 |
|    2 | Komi   | Ireland | 2024-06-29 01:22:44 |
|    3 | Tadano | Ireland | 2024-06-29 01:22:44 |
+------+--------+---------+---------------------+
3 rows in set (0.00 sec)
```

—---

sqoop job --create inc_imp_dt_time \
-- import --connect jdbc:mysql://localhost:3306/testing \
--username root \
--P \
--table inc_imp_date_time \
--m 1 \
--target-dir /user/cloudera/incremental_import_id_date_time \
--incremental lastmodified --check-column start_time --last-value "0000-00-00 00:00:00"
--merge-key id

Concept explanation:



```
Src Table
id,name,city,timestamp
101,saif,mumbai,07:10:2020 07:00:05 --> No Change
102,saif,bangalore,07:10:2020 20:00:15  --> Modified
103,saif,delhi,07:10:2020 07:00:05 --> No Change
104,mano,hyd,07:10:2020 15:00:30 --> New Insert

id --> PK
isnull --> Insert
isNotNull --> Update
102 <> 102
isNotNULL --> No Change
101 <> 101

Tgt Table
==> 1st Day Load:
id,name,city,timestamp
101,saif,mumbai,07:10:2020 07:00:05
102,saif,bangalore,07:10:2020 20:00:15 ---> Record will be updated
103,saif,delhi,07:10:2020 07:00:05
```

Note:

```
24/06/29 01:40:49 INFO mapreduce.Job:  map 0% reduce 0%
24/06/29 01:40:55 INFO mapreduce.Job:  map 100% reduce 0%
24/06/29 01:40:56 INFO mapreduce.Job: Job job_1719042249596_0042 completed successfully
```

(only mapper is used)

—---

```
UPDATE inc_imp_date_time  SET name='Komu', city='Navi-Mumbai' WHERE id=2;
INSERT INTO inc_imp_date_time VALUES (4, 'Komi-san', 'Ireland',now());
```

—--

Again Execute the sqoop job:

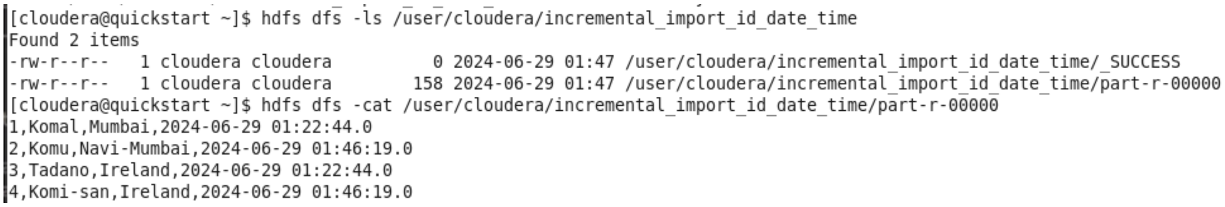
```
sqoop job --exec inc_imp_dt_time
```

```
24/06/29 01:47:18 INFO mapreduce.Job:  map 0% reduce 0%
24/06/29 01:47:27 INFO mapreduce.Job:  map 100% reduce 0%
24/06/29 01:47:33 INFO mapreduce.Job:  map 100% reduce 100%
24/06/29 01:47:34 INFO mapreduce.Job: Job job_1719042249596_0044 completed successfully
```

(Here we can see the reduce job also gets executed)

Note:

So, in Sqoop, it's the only scenario where the reducer is working.



```
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/incremental_import_id_date_time
Found 2 items
-rw-r--r--   1 cloudera cloudera          0 2024-06-29 01:47 /user/cloudera/incremental_import_id_date_time/_SUCCESS
-rw-r--r--   1 cloudera cloudera        158 2024-06-29 01:47 /user/cloudera/incremental_import_id_date_time/part-r-00000
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/incremental_import_id_date_time/part-r-00000
1,Komal,Mumbai,2024-06-29 01:22:44.0
2,Komu,Navi-Mumbai,2024-06-29 01:46:19.0
3,Tadano,Ireland,2024-06-29 01:22:44.0
4,Komi-san,Ireland,2024-06-29 01:46:19.0
```

—---

****

SQOOP EXPORT

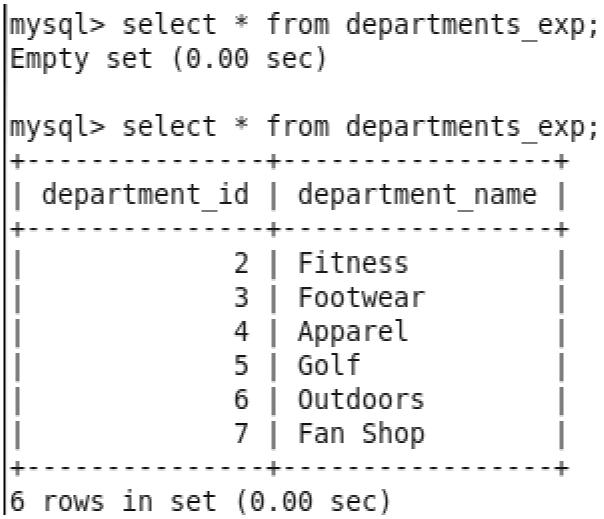
****

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments \
--m 1 \
--target-dir /user/cloudera/departments
```

—---

Sqoop Export:

```
sqoop export \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments_exp \
--export-dir /user/cloudera/departments
```



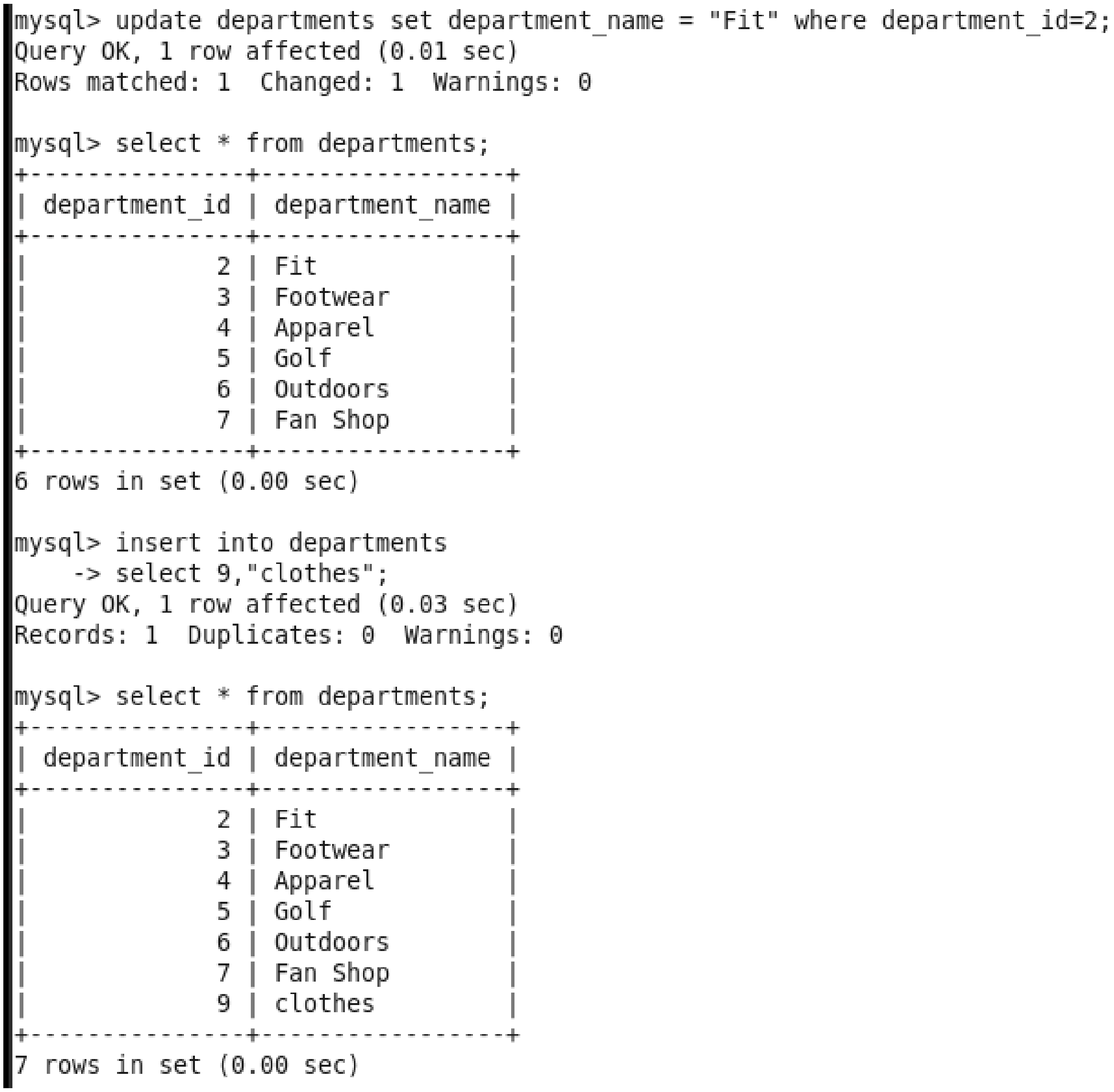
```
mysql> select * from departments_exp;
Empty set (0.00 sec)

mysql> select * from departments_exp;
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             2 | Fitness         |
|             3 | Footwear        |
|             4 | Apparel         |
|             5 | Golf            |
|             6 | Outdoors        |
|             7 | Fan Shop        |
+---------------+-----------------+
6 rows in set (0.00 sec)
```

—--

Sqoop export updateonly:

It only exports the updated existing value, not the newly inserted value.



```
mysql> update departments set department_name = "Fit" where department_id=2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from departments;
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             2 | Fit             |
|             3 | Footwear        |
|             4 | Apparel         |
|             5 | Golf            |
|             6 | Outdoors        |
|             7 | Fan Shop        |
+---------------+-----------------+
6 rows in set (0.00 sec)

mysql> insert into departments
    -> select 9,"clothes";
Query OK, 1 row affected (0.03 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> select * from departments;
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|             2 | Fit             |
|             3 | Footwear        |
|             4 | Apparel         |
|             5 | Golf            |
|             6 | Outdoors        |
|             7 | Fan Shop        |
|             9 | clothes         |
+---------------+-----------------+
7 rows in set (0.00 sec)
```

```
sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments \
--m 1 \
--delete-target-dir \
--target-dir /user/cloudera/departments
```

—---

```
sqoop export \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments_exp \
--export-dir /user/cloudera/departments \
--update-mode updateonly --update-key departement_id
```

—---

Sqoop allowinsert 👍

It boths insert and updates the data.

Note: But it will append the data, not override it.

```
sqoop export \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table departments_exp \
--export-dir /user/cloudera/departments \
--update-mode allowinsert --update-key department_id
```

So, to override it:

You can create the staging table where you can perform update and insert and then can export into main target table.

****

ADDITIONAL COMMANDS

****

Sqoop --null-string (Handling the null values):

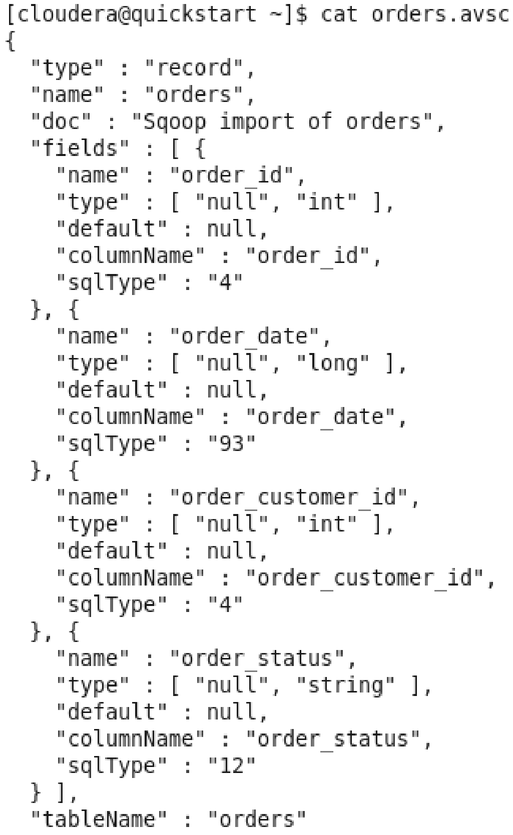
```
sqoop import \
--connect jdbc:mysql://localhost:3306/testing \
--username root \
--P \
--table inc_imp_date \
--m 1 \
--target-dir /user/cloudera/inc_imp_date_new \
--null-string NA \
```

--null-non-string 9999

—---

Sqoop --map-column-java 👍👍

sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \
--P \
--table orders \
--m 1 \
--target-dir /user/cloudera/orders_map \
--as-avrodatafile



```
[cloudera@quickstart ~]$ cat orders.avsc
{
  "type" : "record",
  "name" : "orders",
  "doc" : "Sqoop import of orders",
  "fields" : [ {
    "name" : "order_id",
    "type" : [ "null", "int" ],
    "default" : null,
    "columnName" : "order_id",
    "sqlType" : "4"
  }, {
    "name" : "order_date",
    "type" : [ "null", "long" ],
    "default" : null,
    "columnName" : "order_date",
    "sqlType" : "93"
  }, {
    "name" : "order_customer_id",
    "type" : [ "null", "int" ],
    "default" : null,
    "columnName" : "order_customer_id",
    "sqlType" : "4"
  }, {
    "name" : "order_status",
    "type" : [ "null", "string" ],
    "default" : null,
    "columnName" : "order_status",
    "sqlType" : "12"
  } ],
  "tableName" : "orders"
```

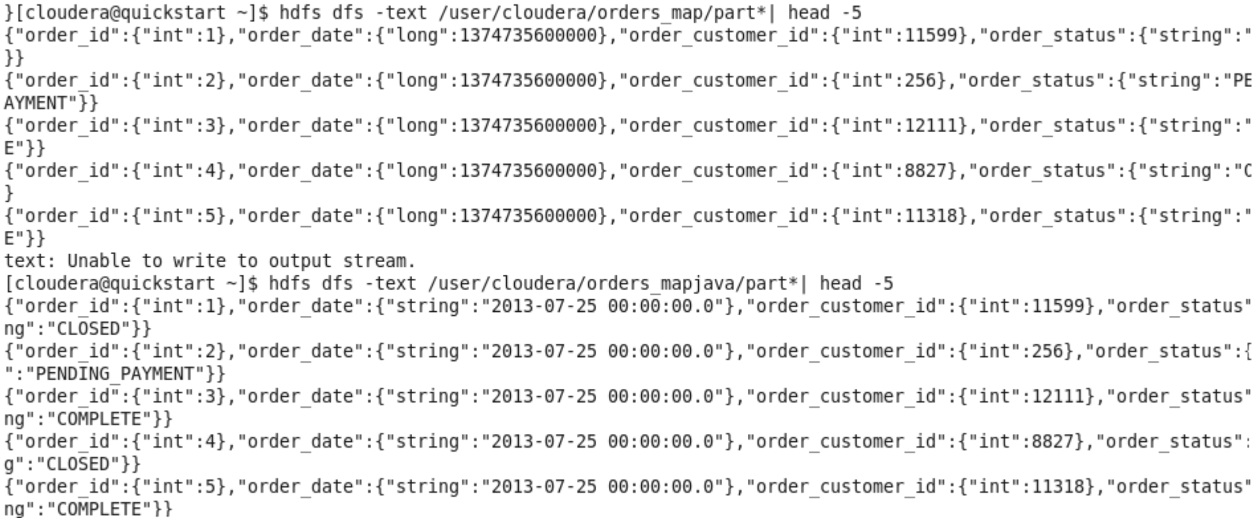
—---

Explicitly we need to change the data type of column:

For above eg, lets change the order_date column datatype from date to string.

sqoop import \
--connect jdbc:mysql://localhost:3306/retail_db \
--username root \

```
--P \
--table orders \
--m 1 \
--target-dir /user/cloudera/orders_mapjava \
--as-avrodatafile \
--map-column-java order_date=String
```



```
}[cloudera@quickstart ~]$ hdfs dfs -text /user/cloudera/orders_map/part*| head -5
{"order_id":{"int":1},"order_date":{"long":1374735600000},"order_customer_id":{"int":11599},"order_status":{"string":"
}}
{"order_id":{"int":2},"order_date":{"long":1374735600000},"order_customer_id":{"int":256},"order_status":{"string":"PE
AYMENT"}}
{"order_id":{"int":3},"order_date":{"long":1374735600000},"order_customer_id":{"int":12111},"order_status":{"string":"
E"}}
{"order_id":{"int":4},"order_date":{"long":1374735600000},"order_customer_id":{"int":8827},"order_status":{"string":"C
}
{"order_id":{"int":5},"order_date":{"long":1374735600000},"order_customer_id":{"int":11318},"order_status":{"string":"
E"}}
text: Unable to write to output stream.
[cloudera@quickstart ~]$ hdfs dfs -text /user/cloudera/orders_mapjava/part*| head -5
{"order_id":{"int":1},"order_date":{"string":"2013-07-25 00:00:00.0"},"order_customer_id":{"int":11599},"order_status"
ng":"CLOSED"}}
{"order_id":{"int":2},"order_date":{"string":"2013-07-25 00:00:00.0"},"order_customer_id":{"int":256},"order_status":{
":"PENDING_PAYMENT"}}
{"order_id":{"int":3},"order_date":{"string":"2013-07-25 00:00:00.0"},"order_customer_id":{"int":12111},"order_status"
ng":"COMPLETE"}}
{"order_id":{"int":4},"order_date":{"string":"2013-07-25 00:00:00.0"},"order_customer_id":{"int":8827},"order_status":
g":"CLOSED"}}
{"order_id":{"int":5},"order_date":{"string":"2013-07-25 00:00:00.0"},"order_customer_id":{"int":11318},"order_status"
ng":"COMPLETE"}}
```

Note always delete that .avsc file or else it will not get overwrite.

—---