

Apache Sqoop Exercise 2



IMPORTANT

Copyright Infringement and Illegal Content Sharing Notice

All course content designs, video, audio, text, graphics, logos, images are Copyright© and are protected by India and international copyright laws. All rights reserved.

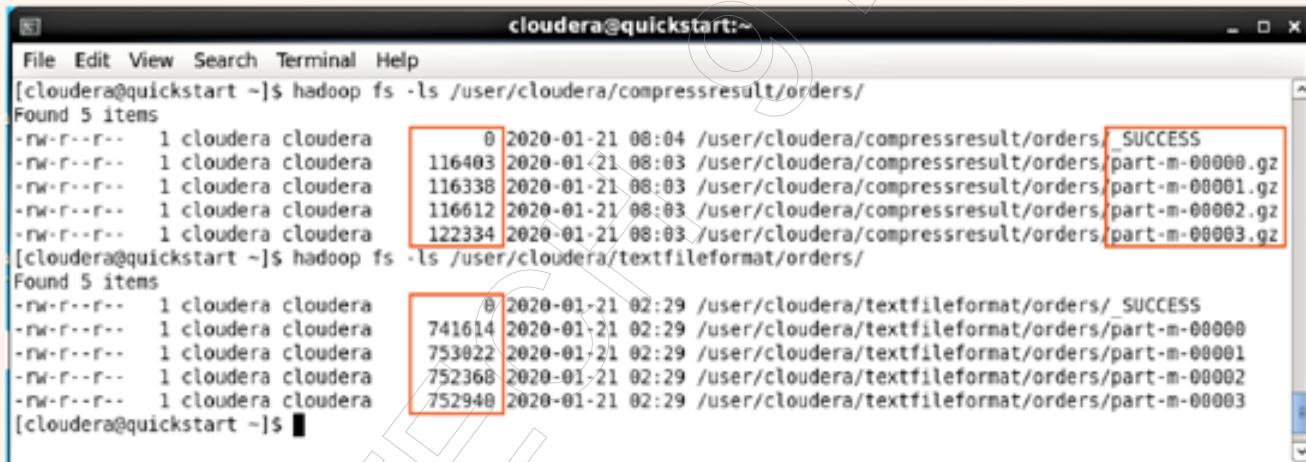
Permission to download the contents (wherever applicable) for the sole purpose of individual reading and preparing yourself to crack the interview only. Any other use of study materials – including reproduction, modification, distribution, republishing, transmission, display – without the prior written permission of Author is strictly prohibited.

Trendytech Insights legal team, along with thousands of our students, actively searches the Internet for copyright infringements. Violators subject to prosecution.

Compression techniques:

You can compress your data by using the default (**gzip**) algorithm with the **-z** or **--compress** argument.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--compress \
--warehouse-dir /user/cloudera/compressresult
```



```
cloudera@quickstart:~$ hadoop fs -ls /user/cloudera/compressresult/orders/
Found 5 items
-rw-r--r-- 1 cloudera cloudera 0 2020-01-21 08:04 /user/cloudera/compressresult/orders/_SUCCESS
116403 2020-01-21 08:03 /user/cloudera/compressresult/orders/part-m-00000.gz
116338 2020-01-21 08:03 /user/cloudera/compressresult/orders/part-m-00001.gz
116612 2020-01-21 08:03 /user/cloudera/compressresult/orders/part-m-00002.gz
122334 2020-01-21 08:03 /user/cloudera/compressresult/orders/part-m-00003.gz
[cloudera@quickstart ~]$ hadoop fs -ls /user/cloudera/textfileformat/orders/
Found 5 items
-rw-r--r-- 1 cloudera cloudera 0 2020-01-21 02:29 /user/cloudera/textfileformat/orders/_SUCCESS
741614 2020-01-21 02:29 /user/cloudera/textfileformat/orders/part-m-00000
753022 2020-01-21 02:29 /user/cloudera/textfileformat/orders/part-m-00001
752368 2020-01-21 02:29 /user/cloudera/textfileformat/orders/part-m-00002
752948 2020-01-21 02:29 /user/cloudera/textfileformat/orders/part-m-00003
```

Note: we will see **.gz** extension)

We can specify any Hadoop compression codec using the **--compression-codec** argument:

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba --password cloudera \
--table orders \
--compression-codec BZip2Codec \
--warehouse-dir /user/cloudera/bzipcomprelult
```

Import the data with selected columns:

We can select a subset of columns and control their ordering by using the **--columns** argument.

For example: **--columns customer_id, customer_fname**

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table customers \
--columns customer_id, customer_fname, customer_city \
--warehouse-dir /user/cloudera/customersresult
```

```
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/customersresult/customers/part-m-00000 | head
1,Richard,Brownsville
2,Mary,Littleton
3,Ann,Caguas
4,Mary,San Marcos
5,Robert,Caguas
6,Mary,Passaic
7,Melissa,Caguas
8,Megan,Lawrence
9,Mary,Caguas
10,Melissa,Stafford
cat: Unable to write to output stream.
[cloudera@quickstart ~]$
```

```
File Edit View Search Terminal Help
[cloudera@quickstart ~]$ hadoop fs -cat /user/cloudera/customersresult1/customers/part-m-00000 | head
Brownsville,Richard,1
Littleton,Mary,2
Caguas,Ann,3
San Marcos,Mary,4
Caguas,Robert,5
Passaic,Mary,6
Caguas,Melissa,7
Lawrence,Megan,8
Caguas,Mary,9
Stafford,Melissa,10
cat: Unable to write to output stream.
[cloudera@quickstart ~]$
```

import with WHERE clause:

You can control which rows are imported by adding a SQL **WHERE** clause to the import statement.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--columns order_id,order_customer_id,order_status \
--where "order_status in ('complete','closed') " \
--warehouse-dir /user/cloudera/customimportresult
```



```
cloudera@quickstart:~$ sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--columns order_id,order_customer_id,order_status \
--where "order_status in ('complete','closed') " \
--warehouse-dir /user/cloudera/customimportresult

20/01/21 20:35:28 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/01/21 20:35:30 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/01/21 20:35:30 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/01/21 20:35:43 INFO db.DBInputFormat: Using read committed transaction isolation
20/01/21 20:35:43 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`order_id`), MAX(`order_id`) FROM `orders` WHERE ( `order_status` IN ('complete','closed') )
20/01/21 20:35:43 INFO db.IntegerSplitter: Split size: 17220; Num splits: 4 from: 1 to: 60000
20/01/21 20:35:43 INFO mapreduce.JobSubmitter: number of splits:4
20/01/21 20:35:44 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579617549308_0010
20/01/21 20:35:45 INFO impl.YarnClientImpl: Submitted application application_1579617549308_0010
20/01/21 20:35:45 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1579617549308_0010/
20/01/21 20:35:45 INFO mapreduce.Job: Running job: job_1579617549308_0010
[cloudera@quickstart ~]$
```

Chek the output of the imported customimportresult file:

```
hadoop fs -cat /user/cloudera/customimportresult/orders/part-m-00000
```



```
cloudera@quickstart:~$ hadoop fs -cat /user/cloudera/customimportresult/orders/part-m-00000 | head
1,11599,CLOSED
3,12111,COMPLETE
4,8827,CLOSED
5,11318,COMPLETE
6,7130,COMPLETE
7,4530,COMPLETE
12,1837,CLOSED
15,2568,COMPLETE
17,2667,COMPLETE
18,1205,CLOSED
cat: Unable to write to output stream.
[cloudera@quickstart ~]$
```

Sqoop import execution flow

- Get metadata of the table by running simple query.
- Build the POJO class with appropriate getters and setters.
- Compile the POJO class into jar file
- Run boundary vals query or boundary query to get min and max by split column (default is primary key column).
- Compute split size max - min.
- Divide it with number of mappers and compute splits.
- Submit map reduce job with number of mappers equal to 4 by default.
- Each map task will run select query on the source table with where condition based on the splits to read the data.
- Data will be written to the files in the location specified.

Sqoop boundary query

- Whenever we run sqoop import with number of mappers greater than 1, a **bounding vals query** will run.
- **Bounding Vals Query** is run on primary key field to get min and max value of it.
- Using the min and max split size is computed.
- We can customize the **bounding vals query** by using **--boundary-query**.
- Query should return 2 values as one row. These values will be used to compute the split size.
- We can hardcode the min and max values with an intention to remove outliers.

Insert one record in Mysql - orders table:

Login to MySQL:

```
mysql -u root -p
```



A terminal window titled "cloudera@quickstart:~". The window shows the command "mysql -u root -p" being run, followed by a prompt "Enter password: [redacted]".

Enter password: **cloudera**

Use Database *retail_db*

```
use retail_db;
```



A terminal window titled "cloudera@quickstart:~". The window shows the command "use retail_db;" followed by "Database changed". Then it runs "show tables;" which lists the following tables:
+-----+
| Tables_in_retail_db |
+-----+
| categories |
| customers |
| departments |
| order_items |
| orders |
| products |
+-----+
6 rows in set (0.00 sec)

Insert one record in the existing *orders* table:

```
insert into orders values  
(200000,'2014-07-23 00:00:00',99999,'COMPLETE');
```

A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows the following text:
File Edit View Search Terminal Help
mysql> insert into orders values (200000,'2014-07-23 00:00:00',99999,'COMPLETE');
Query OK, 1 row affected (0.18 sec)
mysql>

```
commit;
```

Now check the *orders* table to confirm the insert record:

```
select * from orders;
```

A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows the following text:
File Edit View Search Terminal Help
mysql> select * from orders;
+-----+-----+-----+-----+
| id | date | status| price|
+-----+-----+-----+-----+
68875	2014-07-04 00:00:00	ON_HOLD	10637
68876	2014-07-06 00:00:00	COMPLETE	4124
68877	2014-07-07 00:00:00	ON_HOLD	9692
68878	2014-07-08 00:00:00	COMPLETE	6753
68879	2014-07-09 00:00:00	COMPLETE	778
68880	2014-07-13 00:00:00	COMPLETE	1117
68881	2014-07-19 00:00:00	PENDING_PAYMENT	2518
68882	2014-07-22 00:00:00	ON_HOLD	10000
68883	2014-07-23 00:00:00	COMPLETE	5522
200000	2014-07-23 00:00:00	COMPLETE	99999
+-----+-----+-----+-----+
68884 rows in set (0.13 sec)
mysql>

Now, import the updated orders table using Sqoop:

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/ordersnew
```

```
cloudera@quickstart:~$ sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/ordersnew

File Edit View Search Terminal Help
29/04/08 09:36:38 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `orders` AS t LIMIT 1
29/04/08 09:36:36 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `orders` AS t LIMIT 3
29/04/08 09:36:36 INFO orm.CompilationManager: MOODP_HARDFLOAT_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/2285a2fd4ce0e5ea9b9dec05d755/orders.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
29/04/08 09:36:40 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/2285a2fd4ce0e5ea9b9dec05d755/orders.jar
29/04/08 09:36:40 WARN manager.MySQLManager: It looks like you are importing from mysql.
29/04/08 09:36:40 WARN manager.MySQLManager: This transfer can be faster! Use the direct
29/04/08 09:36:40 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path
29/04/08 09:36:40 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
29/04/08 09:36:40 INFO mapreduce.ImportJobBase: Beginning import of orders
29/04/08 09:36:40 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
29/04/08 09:36:40 INFO Configuration.deprecation: mapred.job is deprecated. Instead, use mapreduce.job.jar
29/04/08 09:36:42 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
29/04/08 09:36:42 INFO client.RMProxy: Connecting to ResourceManager at 0.0.0.0:0003
29/04/08 09:36:42 INFO mapreduce.Job: Using JobConf key value as mapreduce.job.name for parameter mapreduce.job.name
29/04/08 09:36:58 INFO db.DataDrivenDBInputFormat: BoundingValsQuery SELECT MIN(`order_id`), MAX(`order_id`) FROM `orders`
29/04/08 09:36:58 INFO db.IntegerSplitters: Split size: 49999; Num splits: 4 from: 1 to: 200000
29/04/08 09:36:59 INFO mapreduce.JobSubmitter: number of splits:4
29/04/08 09:37:04 INFO impl.YarnClientImpl: Submitted application application_1586348454768_0002
29/04/08 09:37:04 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1586348454768_0002
29/04/08 09:37:04 INFO mapreduce.Job: Running job: job_1586348454768_0002
29/04/08 09:37:34 INFO mapreduce.Job: Job job_1586348454768_0002 running in map mode; false
29/04/08 09:37:34 INFO mapreduce.Job: map 0% reduce 0%
```

Check the output in hdfs:

```
hdfs dfs -ls /user/cloudera/ordersnew/orders/
```

```
cloudera@quickstart:~$ hdfs dfs -ls /user/cloudera/ordersnew/
Found 1 items
drwxr-xr-x  - cloudera cloudera      0 2020-04-08 09:16 /user/cloudera/ordersnew/orders
[cloudera@quickstart ~]$ hdfs dfs -ls /user/cloudera/ordersnew/orders/
Found 5 items
-rw-r--r--  1 cloudera cloudera      0 2020-04-08 09:16 /user/cloudera/ordersnew/orders/_SUCCESS
-rw-r--r--  1 cloudera cloudera 2174455 2020-04-08 09:16 /user/cloudera/ordersnew/orders/part-m-00000
-rw-r--r--  1 cloudera cloudera 825489 2020-04-08 09:16 /user/cloudera/ordersnew/orders/part-m-00001
-rw-r--r--  1 cloudera cloudera      0 2020-04-08 09:16 /user/cloudera/ordersnew/orders/part-m-00002
-rw-r--r--  1 cloudera cloudera     44 2020-04-08 09:16 /user/cloudera/ordersnew/orders/part-m-00003
[cloudera@quickstart ~]$
```

Check the contents of mapper files:

```
hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00000
```

```
cloudera@quickstart:~$ hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00000 | head
1,2013-07-25 00:00:00.0,11599,CLOSED
2,2013-07-25 00:00:00.0,256,PENDING PAYMENT
3,2013-07-25 00:00:00.0,12111,COMPLETE
4,2013-07-25 00:00:00.0,8827,CLOSED
5,2013-07-25 00:00:00.0,11318,COMPLETE
6,2013-07-25 00:00:00.0,7130,COMPLETE
7,2013-07-25 00:00:00.0,4530,COMPLETE
8,2013-07-25 00:00:00.0,2911,PROCESSING
9,2013-07-25 00:00:00.0,5657,PENDING PAYMENT
10,2013-07-25 00:00:00.0,5648,PENDING PAYMENT
cat: Unable to write to output stream.
```

```
hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00001
```

```
cloudera@quickstart:~$ hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00001 | head
50001,2014-06-02 00:00:00.0,10731,PENDING PAYMENT
50002,2014-06-02 00:00:00.0,8837,CANCELED
50003,2014-06-02 00:00:00.0,9410,PENDING
50004,2014-06-02 00:00:00.0,7897,COMPLETE
50005,2014-06-02 00:00:00.0,9216,PENDING PAYMENT
50006,2014-06-02 00:00:00.0,11020,COMPLETE
50007,2014-06-02 00:00:00.0,1573,PENDING
50008,2014-06-02 00:00:00.0,6296,COMPLETE
50009,2014-06-02 00:00:00.0,5082,PROCESSING
50010,2014-06-02 00:00:00.0,5061,COMPLETE
cat: Unable to write to output stream.
[cloudera@quickstart ~]$
```

```
hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00002
```

```
cloudera@quickstart:~$ hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00002
[cloudera@quickstart ~]$
```

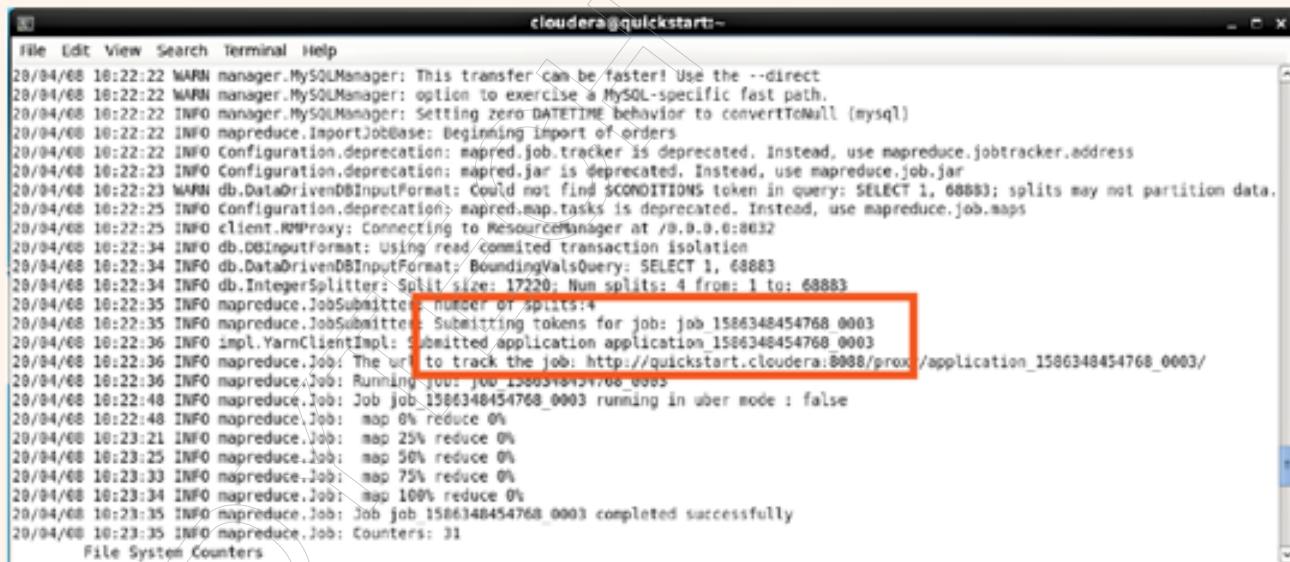
```
hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00003
```

```
cloudera@quickstart:~$ hdfs dfs -cat /user/cloudera/ordersnew/orders/part-m-00003
200000,2014-07-23 00:00:00.0,99999,COMPLETE
[cloudera@quickstart ~]$
```

To solve the previous problem we need to customize the boundary query:

Example of boundary vals query where we can pass hardcoded **min** and **max** values using which splits will be computed.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--boundary-query "SELECT 1, 68883" \
--warehouse-dir /user/cloudera/ordersboundval
```



```
cloudera@quickstart:~$ sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--boundary-query "SELECT 1, 68883" \
--warehouse-dir /user/cloudera/ordersboundval

File Edit View Search Terminal Help
20/04/08 10:22:22 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
20/04/08 10:22:22 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
20/04/08 10:22:22 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
20/04/08 10:22:22 INFO mapreduce.ImportJobBase: Beginning import of orders
20/04/08 10:22:22 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
20/04/08 10:22:23 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/04/08 10:22:23 WARN db.DataDrivenDBInputFormat: Could not find SCONDITIONS token in query: SELECT 1, 68883; splits may not partition data.
20/04/08 10:22:25 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/04/08 10:22:25 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/04/08 10:22:34 INFO db.DBInputFormat: Using read-committed transaction isolation
20/04/08 10:22:34 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT 1, 68883
20/04/08 10:22:34 INFO db.IntegerSplitter: Split size: 17220; Num splits: 4 from: 1 to: 68883
20/04/08 10:22:35 INFO mapreduce.JobSubmitter: number of splits:4
20/04/08 10:22:35 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1586348454768_0003
20/04/08 10:22:36 INFO impl.YarnClientImpl: Submitted application application_1586348454768_0003
20/04/08 10:22:36 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1586348454768_0003/
20/04/08 10:22:36 INFO mapreduce.Job: Running job: job_1586348454768_0003
20/04/08 10:22:48 INFO mapreduce.Job: Job job_1586348454768_0003 running in uber mode : false
20/04/08 10:22:48 INFO mapreduce.Job: map 0% reduce 0%
20/04/08 10:23:21 INFO mapreduce.Job: map 25% reduce 0%
20/04/08 10:23:25 INFO mapreduce.Job: map 50% reduce 0%
20/04/08 10:23:33 INFO mapreduce.Job: map 75% reduce 0%
20/04/08 10:23:34 INFO mapreduce.Job: map 100% reduce 0%
20/04/08 10:23:35 INFO mapreduce.Job: Job job_1586348454768_0003 completed successfully
20/04/08 10:23:35 INFO mapreduce.Job: Counters: 31
File System Counters
```

Check the output in hdfs:

```
hdfs dfs -ls /user/cloudera/ordersboundval/orders/
```

```
cloudera@quickstart:~$ hdfs dfs -ls /user/cloudera/ordersboundval/orders/
Found 5 items
-rw-r--r-- 1 cloudera cloudera 0 2013-04-08 10:40 /user/cloudera/ordersboundval/orders/ SUCCESS
->741614 1 cloudera cloudera 0 2013-04-08 10:40 /user/cloudera/ordersboundval/orders/part-m-00000
->753022 1 cloudera cloudera 0 2013-04-08 10:40 /user/cloudera/ordersboundval/orders/part-m-00001
->752368 1 cloudera cloudera 0 2013-04-08 10:40 /user/cloudera/ordersboundval/orders/part-m-00002
->752940 1 cloudera cloudera 0 2013-04-08 10:40 /user/cloudera/ordersboundval/orders/part-m-00003
[cloudera@quickstart ~]$
```

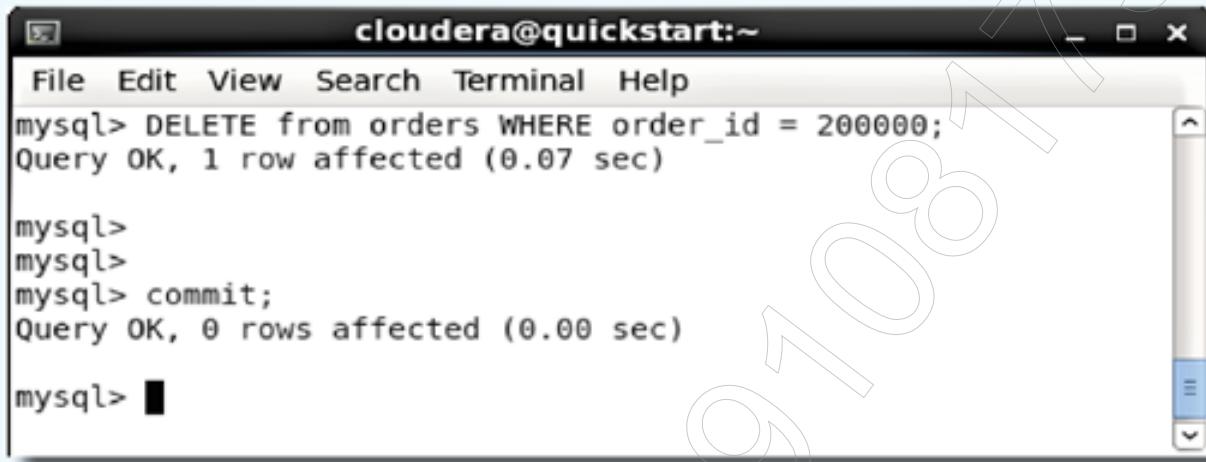
```
cloudera@quickstart:~$ hdfs dfs -cat /user/cloudera/ordersboundval/orders/part-m-00000 | tail
17212,2013-11-09 00:00:00.0,11387,COMPLETE
17213,2013-11-09 00:00:00.0,5166,COMPLETE
17214,2013-11-09 00:00:00.0,585,CLOSED
17215,2013-11-09 00:00:00.0,8326,COMPLETE
17216,2013-11-09 00:00:00.0,5729,COMPLETE
17217,2013-11-09 00:00:00.0,486,COMPLETE
17218,2013-11-09 00:00:00.0,1870,CLOSED
17219,2013-11-09 00:00:00.0,7749,CLOSED
17220,2013-11-09 00:00:00.0,4821,COMPLETE
17221,2013-11-09 00:00:00.0,2366,PENDING
[cloudera@quickstart ~]$
[cloudera@quickstart ~]$ hdfs dfs -cat /user/cloudera/ordersboundval/orders/part-m-00003 | tail
68874,2014-07-03 00:00:00.0,1601,COMPLETE
68875,2014-07-04 00:00:00.0,10637,ON HOLD
68876,2014-07-06 00:00:00.0,4124,COMPLETE
68877,2014-07-07 00:00:00.0,9692,ON HOLD
68878,2014-07-08 00:00:00.0,6753,COMPLETE
68879,2014-07-09 00:00:00.0,778,COMPLETE
68880,2014-07-13 00:00:00.0,1117,COMPLETE
68881,2014-07-19 00:00:00.0,2518,PENDING PAYMENT
68882,2014-07-22 00:00:00.0,10009,ON HOLD
68883,2014-07-23 00:00:00.0,5533,COMPLETE
```

..... Max record

Now we can see with the above command all the mappers did the same amount of work, which is ideally good.

Now, delete the record which we have inserted in *orders* table in Mysql:

```
DELETE from orders WHERE order_id = 200000;
```



A screenshot of a terminal window titled "cloudera@quickstart:~". The window shows the following MySQL session:

```
File Edit View Search Terminal Help
mysql> DELETE from orders WHERE order_id = 200000;
Query OK, 1 row affected (0.07 sec)

mysql>
mysql>
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> ■
```

```
commit;
```

Let us see another example of boundary vals query where we can even customize the min and max values using non primary key column.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table order_items \
--boundary-query "SELECT min(order_item_order_id),
max(order_item_order_id) FROM order_items WHERE order_
item_order_id ge 10000" \
--warehouse-dir /user/cloudera/bvqresult
```

WHERE clause also internally treated as BoundingValsQuery:

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--columns order_id,order_customer_id,order_status \
--where "order_status in ('processing')"\ \
--warehouse-dir /user/cloudera/wherelauseresult
```

```
cloudera@quickstart:~$ sqoop-import \
  --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
  --username retail_dba \
  --password cloudera \
  --table orders \
  --columns order_id,order_customer_id,order_status \
  --where "order_status in ('processing')"\ \
  --warehouse-dir /user/cloudera/wherelauseresult
20/01/21 22:08:03 INFO db.DefaultInputFormat: Using read committed transaction isolation
20/01/21 22:08:03 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(order_id) , MAX(order_id) FROM `orders` WHERE `order_status` in ('processing')
20/01/21 22:08:03 INFO db.IntegerSplitter: Split size: 17215; Num splits: 4 from <=> to: 6036
20/01/21 22:08:03 INFO mapreduce.JobSubmitter: Number of splits: 4
20/01/21 22:08:04 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579617549308_0013
20/01/21 22:08:05 INFO impl.YarnClientImpl: Submitted application application_1579617549308_0013
20/01/21 22:08:05 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1579617549308_0013/
20/01/21 22:08:05 INFO mapreduce.Job: Running job: job_1579617549308_0013
20/01/21 22:08:17 INFO mapreduce.Job: Job job_1579617549308_0013 running in uber mode : false
20/01/21 22:08:17 INFO mapreduce.Job: map 9% reduce 0%
20/01/21 22:09:02 INFO mapreduce.Job: map 50% reduce 0%
```

cloudera@quickstart:~\$ sqoop-import \
 --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
 --username retail_dba \
 --password cloudera \
 --table orders \
 --columns order_id,order_date,order_customer_id,order_status \
 --where "order_status in ('processing')"\ \
 --warehouse-dir /user/cloudera/wherelauseresult

order_id	order_date	order_customer_id	order_status
1	2013-07-25 00:00:00	11599	CLOSED
2	2013-07-25 00:00:00	250	PENDING PAYMENT
3	2013-07-25 00:00:00	12111	COMPLETE
4	2013-07-25 00:00:00	8827	CLOSED
5	2013-07-25 00:00:00	11318	COMPLETE
6	2013-07-25 00:00:00	7130	COMPLETE
7	2013-07-25 00:00:00	1630	COMPLETE
8	2013-07-25 00:00:00	2911	PROCESSING
9	2013-07-25 00:00:00	3657	PENDING PAYMENT
10	2013-07-25 00:00:00	5648	PENDING PAYMENT
		68867	2014-06-23 00:00:00
		68868	2014-06-24 00:00:00
		68869	2014-06-25 00:00:00
		68870	2014-06-26 00:00:00
		68871	2014-06-28 00:00:00
		68872	2014-06-29 00:00:00
		68873	2014-06-30 00:00:00
		68874	2014-07-01 00:00:00
		68875	2014-07-04 00:00:00
		68876	2014-07-06 00:00:00
		68877	2014-07-07 00:00:00
		68878	2014-07-08 00:00:00
		68879	2014-07-09 00:00:00
		68880	2014-07-13 00:00:00
		68881	2014-07-19 00:00:00
		68882	2014-07-22 00:00:00
		68883	2014-07-23 00:00:00

Example2: WHERE clause converted as BoundingValsQuery:

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--where "order_status IN ('COMPLETE', 'CLOSED') AND \
order_date LIKE '2013-08%" \
--warehouse-dir /user/cloudera/bvqresult
```

```
cloudera@quickstart:~$ sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--where "order_status IN ('COMPLETE', 'CLOSED') AND \
order_date LIKE '2013-08%" \
--warehouse-dir /user/cloudera/bvqresult

20/01/23 01:36:19 INFO mapreduce.ImportJobBase: Beginning import of orders
20/01/23 01:36:19 INFO Configuration.deprecation: mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
20/01/23 01:36:19 INFO Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
20/01/23 01:36:19 INFO Configuration.deprecation: mapred.map.tasks is deprecated. Instead, use mapreduce.job.maps
20/01/23 01:36:19 INFO client.HMProxy: Connecting to ResourceManager at /0.0.0.0:8032
20/01/23 01:36:21 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
        at java.lang.Object.wait(Native Method)
        at java.lang.Thread.join(Thread.java:1261)
        at java.lang.Thread.join(Thread.java:1355)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:785)
        at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
20/01/23 01:36:22 INFO db.DBInputFormat: Using read committed transaction isolation
20/01/23 01:36:22 INFO db.DataDrivenDBInputFormat: BoundingValsQuery: SELECT MIN(`order_id`), MAX(`order_id`) FROM `orders`
WHERE `order_status` IN ('COMPLETE', 'CLOSED') AND `order_date` LIKE '2013-08%'
20/01/23 01:36:23 INFO db.IntegerSplitter: Split size: 16853; Num splits: 4 from: 1297 to: 68710
20/01/23 01:36:24 INFO mapreduce.JobSubmitter: number of splits:4
20/01/23 01:36:24 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579752885082_0013
20/01/23 01:36:25 INFO impl.YarnClientImpl: Submitted application application_1579752885082_0013
20/01/23 01:36:25 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_157975288
5082_0013/
20/01/23 01:36:25 INFO mapreduce.Job: Running job: job_1579752885082_0013
```

Sqoop import using split by

Sqoop **--split-by** comes into picture when there is no primary key or the primary key column is not evenly distributed. When to use split-by:

1. When there is no primary key then we can use split-by to indicate the column based on which mappers should divide the work.
2. We can use split-by when your primary key has lot of outliers, then we can use any other column as split-by column (to get a better performance).

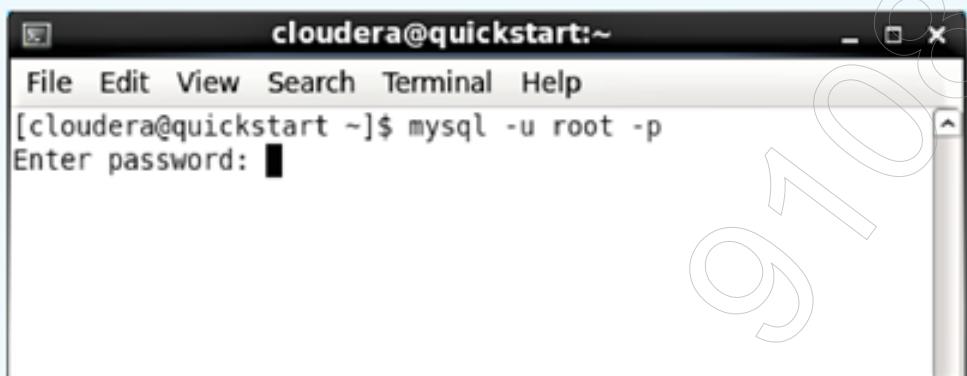
Understanding Sqoop SPLIT-BY

- Let us understand what happens when we run typical sqoop import and when **--split-by** argument should be used.
- Whenever we run sqoop import with number of mappers greater than 1, a bounding vals query will run.
- Bounding Vals Query is run on primary key field to get min and max value of it.
- Using the min and max split size is computed.
- If there is no primary key or unique key, import will fail unless number of mappers is set to 1 or specify a field using split-by.
- It is better practice to use an indexed field which do not contain null values as part of split-by.

Create a table orders_no_pk in MySQL without any primary key:

Login to MySQL:

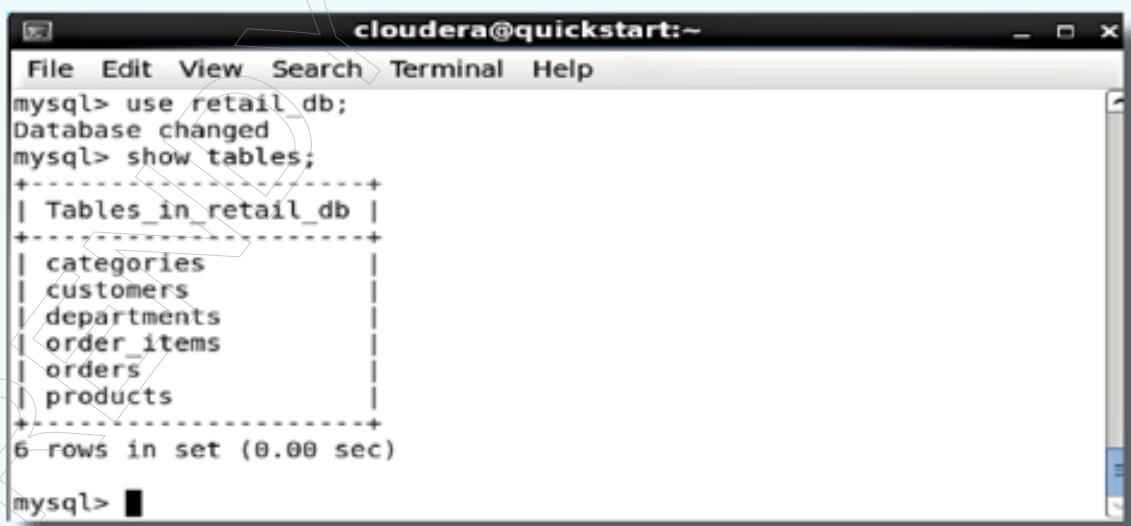
```
mysql -u root -p
```



Enter password: **cloudera**

Use Database *retail_db*

```
use retail_db;
```



Create a *orders_no_pk* table in *retail_db* database:

```
CREATE TABLE orders_no_pk (
    order_id int(11) NOT NULL,
    order_date datetime NOT NULL,
    order_customer_id int(11) NOT NULL,
    order_status varchar(45) NOT NULL
);
```

cloudera@quickstart:~

```
File Edit View Search Terminal Help
mysql> CREATE TABLE orders_no_pk (
    ->     order_id int(11) NOT NULL,
    ->     order_date datetime NOT NULL,
    ->     order_customer_id int(11) NOT NULL,
    ->     order_status varchar(45) NOT NULL
    -> );
Query OK, 0 rows affected (0.16 sec)

mysql>
```

Copying data from *orders* table to *orders_no_pk* table:

```
insert into orders_no_pk select order_id, order_date,
order_customer_id, order_status from orders;
```

cloudera@quickstart:~

```
File Edit View Search Terminal Help
mysql> insert into orders_no_pk select order_id, order_date, order_customer_id,
order status from orders;
Query OK, 68883 rows affected (1.26 sec)
Records: 68883 Duplicates: 0 Warnings: 0

mysql>
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> ■
```

commit;

Now, import the *orders_no_pk* table into HDFS using Sqoop:

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders_no_pk \
--warehouse-dir /ordersnopk
```

```
cloudera@quickstart:~$ sqoop-import \
> --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
> --username retail_dba \
> --password cloudera \
> --table orders_no_pk \
> --warehouse-dir /ordersnopk
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/04/09 01:35:24 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
20/04/09 01:35:24 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
20/04/09 01:35:24 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/04/09 01:35:24 INFO tool.CodeGenTool: Beginning code generation
20/04/09 01:35:25 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `orders_no_pk` AS t LIMIT 1
20/04/09 01:35:25 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `orders_no_pk` AS t LIMIT 1
20/04/09 01:35:25 INFO orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/lib/hadoop-mapreduce
Note: /tmp/sqoop-cloudera/compile/59a42379d3d3f41b7deb8564122873b/orders_no_pk.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
20/04/09 01:35:29 INFO orm.CompilationManager: Writing jar file: /tmp/sqoop-cloudera/compile/59a42379d3d3f41b7deb8564122873b/orders_no_pk.jar
20/04/09 01:35:29 WARN manager.MySQLManager: It looks like you are importing from mysql.
20/04/09 01:35:29 WARN manager.MySQLManager: This transfer can be faster! Use the --direct
20/04/09 01:35:29 WARN manager.MySQLManager: option to exercise a MySQL-specific fast path.
20/04/09 01:35:29 INFO manager.MySQLManager: Setting zero DATETIME behavior to convertToNull (mysql)
20/04/09 01:35:29 ERROR tool.ImportTool: Import failed: No primary key could be found for table orders_no_pk. Please specify one with --split-by or perform a sequential import with '--m 1'.
[cloudera@quickstart ~]$
```

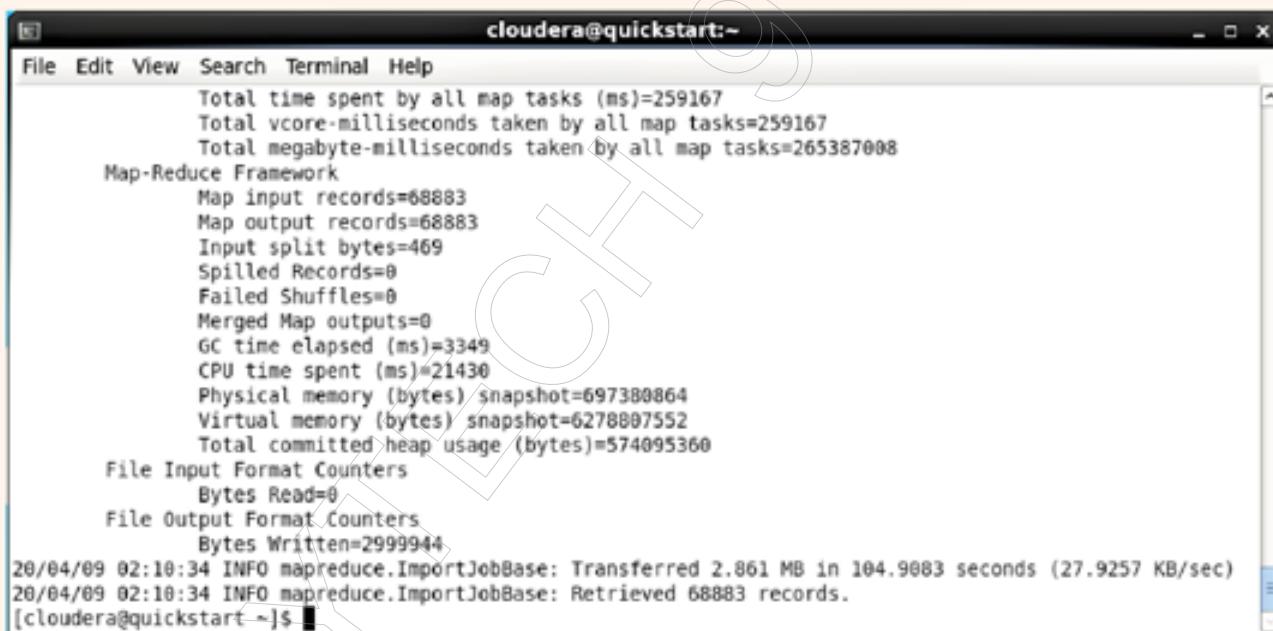
JOB FAILS

Above import fails because the *orders_no_pk* table doesn't have Primary key and Sqoop doesn't know how to divide records among Mappers.

Import the same *orders_no_pk* table into HDFS using **split-by**:

When there is no primary key **--split-by** is the ideal solution for import control.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders_no_pk \
--split-by order_id \
--target-dir /ordersnopksplit
```



```
cloudera@quickstart:~$ sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders_no_pk \
--split-by order_id \
--target-dir /ordersnopksplit

cloudera@quickstart:~$ Total time spent by all map tasks (ms)=259167
Total vcore-milliseconds taken by all map tasks=259167
Total megabyte-milliseconds taken by all map tasks=265387008
Map-Reduce Framework
  Map input records=68883
  Map output records=68883
  Input split bytes=469
  Spilled Records=0
  Failed Shuffles=0
  Merged Map outputs=0
  GC time elapsed (ms)=3349
  CPU time spent (ms)=21430
  Physical memory (bytes) snapshot=697380864
  Virtual memory (bytes) snapshot=6278807552
  Total committed heap usage (bytes)=574095360
File Input Format Counters
  Bytes Read=0
File Output Format Counters
  Bytes Written=2999944
20/04/09 02:10:34 INFO mapreduce.ImportJobBase: Transferred 2.861 MB in 104.9083 seconds (27.9257 KB/sec)
20/04/09 02:10:34 INFO mapreduce.ImportJobBase: Retrieved 68883 records.
[cloudera@quickstart ~]$
```

Note: Split column should have numeric values. It is not recommended to use split-by on a text column.

Dealing with SPLIT-BY or primary key on non numeric fields:

sqoop import using non numeric field fail with hint to use

`org.apache.sqoop.splitter.allow_text_splitter=true`.

It is also applicable when we use non numeric field as part of **split-by** clause.

```
sqoop import \
-Dorg.apache.sqoop.splitter.allow_text_splitter=true \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table categories \
--split-by "category_name" \
--warehouse-dir /user/cloudera/splitonnonnumeric \
--delete-target-dir
```

The screenshot shows a terminal window titled "cloudera@quickstart:~". The user runs the command "hadoop fs -ls /user/cloudera/splitonnonnumeric/categories" which lists seven items. Then, they run "hadoop fs -cat /user/cloudera/splitonnonnumeric/categories/part-m-00001 | head" to view the contents of the first partition, which displays a list of category names.

```
File Edit View Search Terminal Help
cloudera@quickstart:~$ hadoop fs -ls /user/cloudera/splitonnonnumeric/categories
Found 7 items
-rw-r--r-- 1 cloudera cloudera 0 2020-01-23 03:32 /user/cloudera/splitonnonnumeric/categories/
-rw-r--r-- 1 cloudera cloudera 0 2020-01-23 03:31 /user/cloudera/splitonnonnumeric/categories/part-m-00000
-rw-r--r-- 1 cloudera cloudera 344 2020-01-23 03:32 /user/cloudera/splitonnonnumeric/categories/part-m-00001
-rw-r--r-- 1 cloudera cloudera 313 2020-01-23 03:32 /user/cloudera/splitonnonnumeric/categories/part-m-00002
-rw-r--r-- 1 cloudera cloudera 124 2020-01-23 03:32 /user/cloudera/splitonnonnumeric/categories/part-m-00003
-rw-r--r-- 1 cloudera cloudera 228 2020-01-23 03:32 /user/cloudera/splitonnonnumeric/categories/part-m-00004
-rw-r--r-- 1 cloudera cloudera 20 2020-01-23 03:32 /user/cloudera/splitonnonnumeric/categories/part-m-00005
[cloudera@quickstart:~]$ hadoop fs -cat /user/cloudera/splitonnonnumeric/categories/part-m-00001 | head
1,2,Football
3,2,Baseball & Softball
4,2,Basketball
9,3,Cardio Equipment
11,3,Fitness Accessories
12,3,Boxing & MMA
13,3,Electronics
16,3,As Seen on TV!
17,4,Cleats
21,4,Featured Shops
[cloudera@quickstart:~]$
```

Importing table without primary key with single mapper:

```
sqoop import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders_no_pk \
--num-mappers 1 \
--warehouse-dir /user/cloudera/npkresult \
--split-by order_id \
--delete-target-dir
```

```
cloudera@quickstart:~
```

```

File Edit View Search Terminal Help
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:705)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
20/01/23 07:58:29 INFO db.DBInputFormat: Using read committed transaction isolation
20/01/23 07:58:30 INFO mapreduce.JobSubmitter: number of splits:1
20/01/23 07:58:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1579752885082_0018
20/01/23 07:58:31 INFO impl.YarnClientImpl: Submitted application application_1579752885082_0018
20/01/23 07:58:31 INFO mapreduce.Job: The url to track the job: http://quickstart.cloudera:8088/proxy/application_1579752885082_0018/
20/01/23 07:58:31 INFO mapreduce.Job: Running job: job_1579752885082_0018
20/01/23 07:58:44 INFO mapreduce.Job: Job job_1579752885082_0018 running in uber mode : false
20/01/23 07:58:44 INFO mapreduce.Job: map 0% reduce-0%
20/01/23 07:58:56 INFO mapreduce.Job: map 100% reduce-0%
20/01/23 07:58:58 INFO mapreduce.Job: Job job_1579752885082_0018 completed successfully
20/01/23 07:58:59 INFO mapreduce.Job: Counters: 30
    File System Counters
        FILE: Number of bytes read=0

```

Sqoop autoreset to one mapper:

```
sqoop import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders_no_pk \
--warehouse-dir /user/cloudera/npkresult \
--delete-target-dir \
--autoreset-to-one-mapper \
--num-mappers 8
```

Sqoop autoreset to one: Example 2

```
sqoop import-all-tables \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--warehouse-dir /user/cloudera/autoreset1mresult \
--autoreset-to-one-mapper \
--num-mappers 2
```

The screenshot shows a terminal window titled "cloudera@quickstart:~". The user has run the command "hadoop fs -ls /user/cloudera/autoreset1mresult" which lists seven items in the directory:

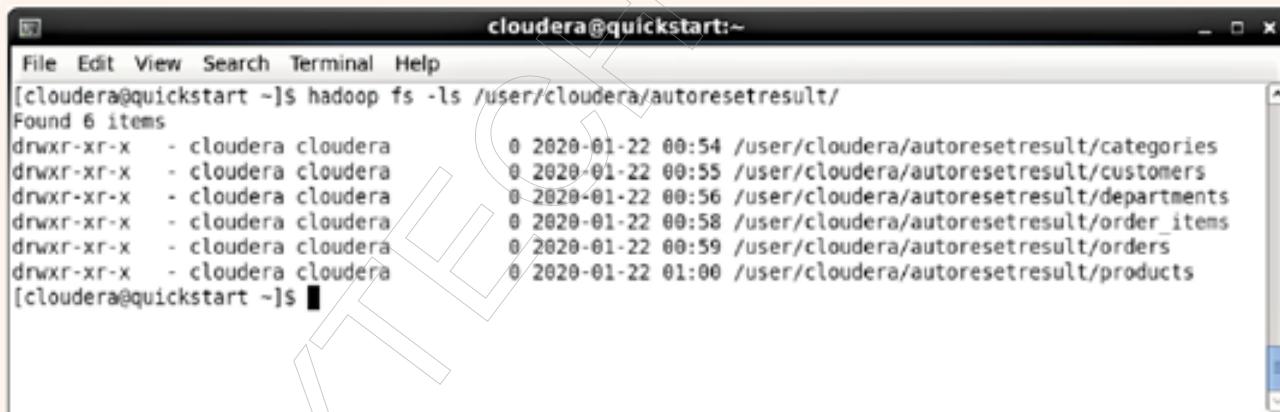
File Type	Last Modified	Path
drwxr-xr-x	2020-01-23 08:25	/user/cloudera/autoreset1mresult/categories
drwxr-xr-x	2020-01-23 08:26	/user/cloudera/autoreset1mresult/customers
drwxr-xr-x	2020-01-23 08:26	/user/cloudera/autoreset1mresult/departments
drwxr-xr-x	2020-01-23 08:27	/user/cloudera/autoreset1mresult/order_items
drwxr-xr-x	2020-01-23 08:28	/user/cloudera/autoreset1mresult/orders
drwxr-xr-x	2020-01-23 08:28	/user/cloudera/autoreset1mresult/orders_npk
drwxr-xr-x	2020-01-23 08:29	/user/cloudera/autoreset1mresult/products

Import tables with or without primary key:

While importing all tables from RDBMS to HDFS, If a table does not have a primary key defined then import will fail.

Sqoop argument **--autoreset-to-one-mapper** uses one mapper if a table with no primary key is encountered.

```
sqoop-import-all-tables \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--autoreset-to-one-mapper \
--warehouse-dir /user/cloudera/autoresetresult
```



The screenshot shows a terminal window titled "cloudera@quickstart:~". The user has run the command "hadoop fs -ls /user/cloudera/autoresetresult/" which lists six items, each corresponding to a table imported from the MySQL database:

File Path	Modification Time
/user/cloudera/autoresetresult/categories	2020-01-22 00:54
/user/cloudera/autoresetresult/customers	2020-01-22 00:55
/user/cloudera/autoresetresult/departments	2020-01-22 00:56
/user/cloudera/autoresetresult/order_items	2020-01-22 00:58
/user/cloudera/autoresetresult/orders	2020-01-22 00:59
/user/cloudera/autoresetresult/products	2020-01-22 01:00

Delimiters

Delimiters may be specified by following arguments

- **--fields-terminated-by** <char>
 - **--lines-terminated-by** <char>

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--fields-terminated-by '|' \
--lines-terminated-by ';' \
--target-dir /user/cloudera/delimiterresult
```

```
cloudera@quickstart:~
```

```
File Edit View Search Terminal Help
00:00:00.0|1594|PENDING PAYMENT;17171|2013-11-09 00:00:00.0|2772|PENDING PAYMENT;17172|2013-11-09 00:00:0
0.0|3838|COMPLETE;17173|2013-11-09 00:00:00.0|5410|PAYMENT REVIEW;17174|2013-11-09 00:00:00.0|8430|SUSPECT
ED FRAUD;17175|2013-11-09 00:00:00.0|8532|COMPLETE;17176|2013-11-09 00:00:00.0|8247|COMPLETE;17177|2013-11
-09 00:00:00.0|9314|PENDING PAYMENT;17178|2013-11-09 00:00:00.0|2746|PROCESSING;17179|2013-11-09 00:00:00.
0|2023|COMPLETE;17180|2013-11-09 00:00:00.0|5985|PENDING PAYMENT;17181|2013-11-09 00:00:00.0|4764|PROCES
SING;17182|2013-11-09 00:00:00.0|3634|PENDING;17183|2013-11-09 00:00:00.0|1443|ON HOLD;17184|2013-11-09 00:0
0.0|9417|COMPLETE;17185|2013-11-09 00:00:00.0|10843|PROCESSING;17186|2013-11-09 00:00:00.0|1846|PROCES
SING;17187|2013-11-09 00:00:00.0|1972|PENDING PAYMENT;17188|2013-11-09 00:00:00.0|454|PENDING PAYMENT;17189
|2013-11-09 00:00:00.0|5225|PROCESSING;17190|2013-11-09 00:00:00.0|6021|COMPLETE;17191|2013-11-09 00:00:00.
0|1455|PENDING PAYMENT;17192|2013-11-09 00:00:00.0|9320|PENDING;17193|2013-11-09 00:00:00.0|1593|PENDING
PAYMENT;17194|2013-11-09 00:00:00.0|2581|CANCELED;17195|2013-11-09 00:00:00.0|621|COMPLETE;17196|2013-11-0
9 00:00:00.0|3293|COMPLETE;17197|2013-11-09 00:00:00.0|1540|SUSPECTED FRAUD;17198|2013-11-09 00:00:00.0|64
2|CLOSED;17199|2013-11-09 00:00:00.0|7246|PENDING PAYMENT;17200|2013-11-09 00:00:00.0|4846|PENDING PAYMENT
;17201|2013-11-09 00:00:00.0|10506|PENDING PAYMENT;17202|2013-11-09 00:00:00.0|4145|PROCESSING;17203|2013-
11-09 00:00:00.0|6725|COMPLETE;17204|2013-11-09 00:00:00.0|3960|CLOSED;17205|2013-11-09 00:00:00.0|2715|CL
OSED;17206|2013-11-09 00:00:00.0|2848|PROCESSING;17207|2013-11-09 00:00:00.0|8986|COMPLETE;17208|2013-11-0
9 00:00:00.0|1364|CLOSED;17209|2013-11-09 00:00:00.0|336|CLOSED;17210|2013-11-09 00:00:00.0|12143|PENDING
PAYMENT;17211|2013-11-09 00:00:00.0|1595|COMPLETE;17212|2013-11-09 00:00:00.0|11387|COMPLETE;17213|2013-11
-09 00:00:00.0|6166|COMPLETE;17214|2013-11-09 00:00:00.0|585|CLOSED;17215|2013-11-09 00:00:00.0|8326|COMPL
ETE;17216|2013-11-09 00:00:00.0|5729|COMPLETE;17217|2013-11-09 00:00:00.0|486|COMPLETE;17218|2013-11-09 00
:00:00.0|1870|CLOSED;17219|2013-11-09 00:00:00.0|7749|CLOSED;17220|2013-11-09 00:00:00.0|4821|COMPLETE;172
[cloudera@quickstart ~]$
```

Create a Hive table based on a database table:

The **--create-hive-table** argument populates a Hive metastore with a definition for a table based on a database.

```
sqoop create-hive-table \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--hive-table emps \
--fields-terminated-by ','
```

created a empty table in hive based on metadata in mysql.

```
cloudera@quickstart:~ [cloudera@quickstart ~]$ sqoop create-hive-table \
> --connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
> --username retail_dba \
> --password cloudera \
> --table orders \
> --hive-table emps \
> --fields-terminated-by ',';
Warning: /usr/lib/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
20/01/22 03:35:11 INFO sqoop.Sqoop: Running Sqoop version: 1.4.6-cdh5.13.0
20/01/22 03:35:11 WARN tool.BaseSqoopTool: Setting your password on the command-line is insecure. Consider using -P instead.
20/01/22 03:35:11 INFO manager.MySQLManager: Preparing to use a MySQL streaming resultset.
20/01/22 03:35:12 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `orders` AS t LIMIT 1
20/01/22 03:35:13 INFO manager.SqlManager: Executing SQL statement: SELECT t.* FROM `orders` AS t LIMIT 1
20/01/22 03:35:13 WARN hive.TableDefWriter: Column order_date had to be cast to a less precise type in Hive
20/01/22 03:35:15 INFO hive.HiveImport: Loading uploaded data into Hive
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.0.jar!/hive-log4j.properties
OK
Time taken: 48.454 seconds
[cloudera@quickstart ~]$
```

cloudera@quickstart:~

```
File Edit View Search Terminal Help
mysql> desc orders;
+-----+-----+
| Field | Type |
+-----+-----+
| order_id | int(11) |
| order_date | datetime |
| order_customer_id | int(11) |
| order_status | varchar(45) |
+-----+-----+
4 rows in set (0.09 sec)

mysql>
```

cloudera@quickstart:~

```
File Edit View Search Terminal Help
hive> show tables;
OK
emps
Time taken: 0.042 seconds, Fetched: 1 row(s)
hive> desc emps;
OK
order_id          int
order_date        string
order_customer_id int
order_status      string
Time taken: 0.151 seconds, Fetched: 4 row(s)
hive>
```

Note: error out if the hive table already exists.

Scoop Verbose:

Run your Sqoop job with the `--verbose` flag to generate more logs and debugging information.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--verbose \
--target-dir /user/cloudera/verboseresult
```

Note: We can see the bounding val queries framed properly.

Sqoop append:

By default, imports go to a new target location. If the destination directory already exists in HDFS, Sqoop will refuse to import.

The **--append** argument will append data to an existing dataset in HDFS.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--target-dir /user/cloudera/appendresult \
--append
```

1st run folder structure

2nd run folder structure

Delete target directory if exists:

```
sqoop-import \  
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \  
--username retail_dba \  
--password cloudera \  
--table orders \  
--target-dir /user/cloudera/appendresult \  
--delete-target-dir
```

Controlling Parallelism:

Specify the number of map tasks (parallel processes) to import data by using the **-m** or **--num-mappers** argument.

```
sqoop-import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--target-dir /user/cloudera/mapperresult \
--delete-target-dir \
--num-mappers 8
```

```
cloudera@quickstart:~$ hadoop fs -ls /user/cloudera/mapperresult
Found 9 items
-rw-r--r-- 1 cloudera cloudera      0 2020-01-22 09:37 /user/cloudera/mapperresult/_SUCCESS
-rw-r--r-- 1 cloudera cloudera 366460 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00000
-rw-r--r-- 1 cloudera cloudera 375196 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00001
-rw-r--r-- 1 cloudera cloudera 376486 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00002
-rw-r--r-- 1 cloudera cloudera 376494 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00003
-rw-r--r-- 1 cloudera cloudera 376169 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00004
-rw-r--r-- 1 cloudera cloudera 376199 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00005
-rw-r--r-- 1 cloudera cloudera 376098 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00006
-rw-r--r-- 1 cloudera cloudera 376842 2020-01-22 09:37 /user/cloudera/mapperresult/part-m-00007
```

Displaying schema of mysql table from terminal:

```
sqoop eval \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
-e "DESCRIBE customers"
```

Field	Type	Null	Key	Default	Extra
customer_id	int(11)	NO	PRI	[null]	auto_increment
customer_fname	varchar(45)	NO		[null]	
customer_lname	varchar(45)	NO		[null]	
customer_email	varchar(45)	NO		[null]	
customer_password	varchar(45)	NO		[null]	
customer_street	varchar(255)	NO		[null]	
customer_city	varchar(45)	NO		[null]	
customer_state	varchar(45)	NO		[null]	
customer_zipcode	varchar(45)	NO		[null]	

Dealing with nulls while importing data

When we import data into text files, we might have to explicitly deal with null values.

We can specify non string nulls using **--null-non-string** and string nulls using **--null-string**.

```
sqoop import \
--connect jdbc:mysql://quickstart.cloudera:3306/retail_db \
--username retail_dba \
--password cloudera \
--table orders \
--warehouse-dir /user/cloudera/nullstringresult \
--delete-target-dir \
--null-non-string "-1"
```

The screenshot shows two terminal windows side-by-side. Both windows are titled "cloudera@quickstart:~". The left window displays the original data from the MySQL database, where null values are represented by empty strings. The right window shows the same data after applying the `--null-non-string` option, where null values are replaced by the specified value, in this case, "-1". Red boxes highlight the null entries in both columns for comparison.

Original Data (Left Window)	Data with --null-non-string (Right Window)
16917,-1,5931,COMPLETE	16917,null,5931,COMPLETE
16918,-1,6360,COMPLETE	16918,null,6360,COMPLETE
16919,-1,1725,PENDING PAYMENT	16919,null,1725,PENDING PAYMENT
16920,-1,4234,PENDING PAYMENT	16920,null,4234,PENDING PAYMENT
16921,-1,3126,COMPLETE	16921,null,3126,COMPLETE
16922,-1,4929,COMPLETE	16922,null,4929,COMPLETE
16923,-1,10474,CANCELED	16923,null,10474,CANCELED
16924,-1,1925,PENDING PAYMENT	16924,null,1925,PENDING PAYMENT
16925,-1,11275,COMPLETE	16925,null,11275,COMPLETE
16926,-1,5731,	

nulls are replaced with
non-null string value



5 Star Google Rated
Big Data Course

LEARN FROM THE EXPERT



9108179578

Call for more details

Follow US

Trainer Mr. Sumit Mittal

LinkedIn <https://www.linkedin.com/in/bigdatabysumit/>

Website <https://trendytech.in/courses/big-data-online-training/>

Phone 9108179578

Email trendytech.sumit@gmail.com

Youtube TrendyTech

Twitter @BigdataBySumit

Instagram bigdatabysumit

Facebook <https://www.facebook.com/trendytech.in/>

