

ASSIGNMENT 11.1

Problem Statement:

Perform and explain the code flow and the associated result for the below tasks. Candidates should create and use their own employee dataset for the same. Share the screenshot of the commands used and its associated result.

- Transfer data between MySQL and HDFS (Import and Export) using Sqoop.
- Transfer data between MySQL and Hive (Import and Export only selected columns) using Sqoop.

Solution:

1. Import data from MySQL to HDFS using Sqoop:

Step 1: Start Hadoop system processes like NameNode, DataNode, etc and MySQL service.

Here is the command to start Hadoop system processes:

```
$ /usr/local/hadoop-2.6.0/sbin/start-all.sh
```

We can verify that they are running by using 'jps' command.

Here is the command to start MySQL service:

```
$ sudo service mysqld start
```

This command will ask for password of super user. Type 'acadgild' and hit Enter.

```
[acadgild@localhost bin]$ jps
2657 NameNode
8673 NodeManager
3092 ResourceManager
2758 DataNode
2887 SecondaryNameNode
8840 JobHistoryServer
18298 Jps
[acadgild@localhost bin]$ sudo service mysqld start
[sudo] password for acadgild:
Starting mysqld: [ OK ]
[acadgild@localhost bin]$ █
```

Step 2: Enter into MySQL prompt and create a new database.

We can use the following command to get into MySQL prompt:

```
$ sudo mysql
```

Once we get MySQL prompt, we can see a list existing databases via following command:

```
mysql> SHOW DATABASES;
```

Let's create a database named 'sample_db' with the following command:

```
mysql> CREATE DATABASE sample_db;
```

```
[acadgild@localhost bin]$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7
Server version: 5.1.73 Source distribution

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| b1 |
| metastore |
| mysql |
| test |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> CREATE DATABASE sample_db;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| b1 |
| metastore |
| mysql |
| sample_db |
+-----+
```

I

Step 3: Switch to the database created. Create a table by name ‘employee’ and insert few records into it.

Here is the command to switch to the database that we create in the previous step:

```
mysql> use sample_db;
```

Let’s create a table by name ‘employee’ with following command:

```
mysql> CREATE TABLE employee
(emp_id int,
emp_name varchar(100),
emp_salary int,
years_of_exp int
);
```

The ‘employee’ schema has four fields: employee id, employee name, employee salary and years of experience.

```
mysql> use sample_db;
Database changed
mysql> CREATE TABLE employee
-> (emp_id int,
-> emp_name varchar(100),
-> emp_salary int,
-> years_of_exp int
-> );
Query OK, 0 rows affected (0.01 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_sample_db |
+-----+
| employee             |
+-----+
1 row in set (0.00 sec)

mysql> DESCRIBE employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id     | int(11)       | YES  |     | NULL    |       |
| emp_name   | varchar(100)  | YES  |     | NULL    |       |
| emp_salary | int(11)       | YES  |     | NULL    |       |
| years_of_exp | int(11)      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> █
```

Now let's insert few records into this table using INSERT command:

```
mysql> INSERT INTO employee VALUES(101,'Amitabh',20000,1);
```

```
mysql> INSERT INTO employee VALUES(102,'Shahrukh',10000,2);
```

```
mysql> INSERT INTO employee VALUES(103,'Akshay',11000,3);
```

```
mysql> INSERT INTO employee VALUES(104,'Anubhav',5000,4);
```

```
mysql> INSERT INTO employee VALUES(105,'Pawan',2500,5);
```

```
mysql> INSERT INTO employee VALUES(106,'Aamir',25000,1);
```

```
mysql> INSERT INTO employee VALUES(107,'Salman',17500,2);
```

We have inserted seven records with employee id in sequential manner starting from value '101'.

We can see the result by using SELECT command on this table:

```
SELECT * FROM employee;
```

```
mysql> INSERT INTO employee VALUES(101,'Amitabh',20000,1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(102,Shahrukh,10000,2);
ERROR 1054 (42S22): Unknown column 'Shahrukh' in 'field list'
mysql> INSERT INTO employee VALUES(102,'Shahrukh',10000,2);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(103,'Akshay',11000,3);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(104,'Anubhav',5000,4);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(105,'Pawan',2500,5);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(106,'Aamir',25000,1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee VALUES(107,'Salman',17500,2);
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM employee;
+-----+-----+-----+-----+
| emp_id | emp_name | emp_salary | years_of_exp |
+-----+-----+-----+-----+
| 101 | Amitabh | 20000 | 1 |
| 102 | Shahrukh | 10000 | 2 |
| 103 | Akshay | 11000 | 3 |
| 104 | Anubhav | 5000 | 4 |
| 105 | Pawan | 2500 | 5 |
| 106 | Aamir | 25000 | 1 |
| 107 | Salman | 17500 | 2 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> █
```

Step 4: Grant privileges to access tables outside of MySQL and commit all the operations performed till now.

Here are few commands to grant access to this table and other related commands.

```
mysql> grant all on *.* to 'root'@'localhost' with grant option;
```

```
mysql> flush privileges;
```

```
mysql> commit;
```

```
mysql> grant all on *.* to 'root'@'localhost' with grant option;
Query OK, 0 rows affected (0.00 sec)

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

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

We can exit from MySQL now:

```
mysql> exit;
```

Step 5: Use Sqoop import command to import data from MySQL table into HDFS.

Here is the import command to achieve this:

```
$ sqoop import --connect jdbc:mysql://localhost/sample_db --table employee --username root -P --split-by 'years_of_exp' --target-dir '/sqoop_output' -m 2;
```

In the above command, connect parameter takes a JDBC URL to connect to a specific database, in this case 'sample_db'. We can specify login credentials (user name and password to MySQL) and table name subsequently. An interesting parameter here is '**split-by**'. Sqoop automatically splits the table data by primary key column which will split total number of records based on that column's data and performs transfer operation in parallel by assigning the task to several mappers. If primary key is not present for a table, we have to use split-by option followed by a column name for Sqoop to consider that field while splitting data.

In this example, we have mentioned number of mappers as 2 and we are splitting data based on the column 'years_of_exp'. If we observe the data inserted into employee table, we have four similar values in this column (1 and 2 in repeated manner). So one mapper will get these four data records and another mapper gets remaining records in the execution of the above command.

Note: We haven't set any password for MySQL. We can hit 'Enter' when it prompts for the same.

```
[acadgild@localhost bin]$ sqoop import --connect jdbc:mysql://localhost/sample_db --table employee --username root -P --split-by 'years_of_exp' --target-dir '/sqoop_output' -m 2;
Warning: /usr/local/sqoop/./hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/./accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/./zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-23 02:20:39,614 INFO [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2018-01-23 02:20:41,479 INFO [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-23 02:20:41,480 INFO [main] tool.CodeGenTool: Beginning code generation
2018-01-23 02:20:42,157 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2018-01-23 02:20:42,239 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2018-01-23 02:20:42,285 INFO [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
```

Let's check the output by examining the contents of the directory /sqoop_output in HDFS.

```
$ Hadoop fs -ls /sqoop_output
```

```
[acadgild@localhost ~]$ hadoop fs -ls /sqoop_output
18/01/23 02:25:10 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 3 items
-rw-r--r--  1 acadgild supergroup          0 2018-01-23 02:22 /sqoop_output/_SUCCESS
-rw-r--r--  1 acadgild supergroup       78 2018-01-23 02:22 /sqoop_output/part-m-00000
-rw-r--r--  1 acadgild supergroup       55 2018-01-23 02:22 /sqoop_output/part-m-00001
```

As we can see there are two files one from each mapper. Let's see the contents of these two files:

```
$ hadoop fs -cat /scoop_output/part-m-00000
```

```
$ hadoop fs -cat /scoop_output/part-m-00001
```

```
[acadgild@localhost ~]$ hadoop fs -cat /scoop_output/part-m-00000
18/01/23 02:25:37 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
101,Amitabh,20000,1
102,Shahrukh,10000,2
106,Aamir,25000,1
107,Salman,17500,2
[acadgild@localhost ~]$ hadoop fs -cat /scoop_output/part-m-00001
18/01/23 02:25:53 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
103,Akshay,11000,3
104,Anubhav,5000,4
105,Pawan,2500,5
[acadgild@localhost ~]$ █
```

All records from MySQL table 'employee' have been imported to HDFS successfully.

2. Export data from HDFS to MySQL using Sqoop:

Step 1: create a table by name 'employee' in MySQL.

For simplicity, we will use the same table as the one we created in previous task and delete all records from that table.

```
mysql> DELETE FROM employee;
Query OK, 7 rows affected (0.00 sec)

mysql> SELECT * FROM employee;
Empty set (0.00 sec)

mysql> DESCRIBE employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id     | int(11)   | YES  |     | NULL    |       |
| emp_name   | varchar(100) | YES  |     | NULL    |       |
| emp_salary | int(11)   | YES  |     | NULL    |       |
| years_of_exp | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Step 2: Use Sqoop export command to export data from HDFS to MySQL.

Here is the command we have used to achieve this:

```
$ sqoop export --connect jdbc:mysql://localhost/sample_db --username 'root' -P --table 'employee'
--export-dir '/scoop_output' --input-fields-terminated-by ',' -m 1 --columns emp_id, emp_name,
emp_salary, years_of_exp
```

```
[acadgild@localhost ~]$ sqoop export --connect jdbc:mysql://localhost/sample_db --username 'root' -P --table 'employee' --export-dir '/sqoop_output' --input-fields-terminated-by ',' -m 1 --columns emp_id,emp_name,emp_salary,years_of_exp
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-23 03:27:26,217 INFO [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2018-01-23 03:27:30,623 INFO [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-23 03:27:30,623 INFO [main] tool.CodeGenTool: Beginning code generation
2018-01-23 03:27:31,306 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2018-01-23 03:27:31,417 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2018-01-23 03:27:31,442 INFO [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0

2018-01-23 03:27:44,369 INFO [main] Configuration.deprecation: mapred.working.dir is deprecated. Instead, use mapreduce.job.working.dir
2018-01-23 03:27:44,369 INFO [main] Configuration.deprecation: mapred.mapoutput.value.class is deprecated. Instead, use mapreduce.map.output.value.class
2018-01-23 03:27:44,371 INFO [main] Configuration.deprecation: mapred.mapoutput.key.class is deprecated. Instead, use mapreduce.map.output.key.class
2018-01-23 03:27:44,371 INFO [main] Configuration.deprecation: mapred.job.classpath.files is deprecated. Instead, use mapreduce.job.classpath.files
2018-01-23 03:27:44,371 INFO [main] Configuration.deprecation: user.name is deprecated. Instead, use mapreduce.job.user.name
2018-01-23 03:27:44,371 INFO [main] Configuration.deprecation: mapred.reduce.tasks is deprecated. Instead, use mapreduce.job.reduce.tasks
2018-01-23 03:27:44,373 INFO [main] Configuration.deprecation: mapred.cache.files.filesizes is deprecated. Instead, use mapreduce.job.cache.files.filesizes
2018-01-23 03:27:45,154 INFO [main] mapreduce.JobSubmitter: Submitting tokens for job: job_1516637375581_0011
2018-01-23 03:27:45,746 INFO [main] impl.YarnClientImpl: Submitted application application_1516637375581_0011 to ResourceManager at /0.0.0.0:8032
2018-01-23 03:27:45,832 INFO [main] mapreduce.Job: The url to track the job: http://http://localhost:8088/proxy/application_1516637375581_0011/
2018-01-23 03:27:45,833 INFO [main] mapreduce.Job: Running job: job_1516637375581_0011
2018-01-23 03:28:41,955 INFO [main] mapreduce.Job: Job job_1516637375581_0011 running in uber mode : false
2018-01-23 03:28:41,992 INFO [main] mapreduce.Job: map 0% reduce 0%
2018-01-23 03:29:23,769 INFO [main] mapreduce.Job: map 100% reduce 0%
2018-01-23 03:29:28,610 INFO [main] mapreduce.Job: Job job_1516637375581_0011 completed successfully
```

Let's examine the contents of MySQL table 'employee'

mysql> SELECT * FROM employee;

mysql> SELECT * FROM employee;
Empty set (0.00 sec)

mysql> DESCRIBE employee;

Field	Type	Null	Key	Default	Extra
emp_id	int(11)	YES		NULL	
emp_name	varchar(100)	YES		NULL	
emp_salary	int(11)	YES		NULL	
years_of_exp	int(11)	YES		NULL	

4 rows in set (0.00 sec)

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

```
mysql> SELECT * FROM employee;
+----+-----+-----+-----+
| emp_id | emp_name | emp_salary | years_of_exp |
+----+-----+-----+-----+
| 101 | Amitabh | 20000 | 1 |
| 102 | Shahrukh | 10000 | 2 |
| 106 | Aamir | 25000 | 1 |
| 107 | Salman | 17500 | 2 |
| 103 | Akshay | 11000 | 3 |
| 104 | Anubhav | 5000 | 4 |
| 105 | Pawan | 2500 | 5 |
+----+-----+-----+-----+
7 rows in set (0.01 sec)
```

We can see the difference in the dataset of employee table before and after Sqoop export operation.

3. Import data from MySQL to Hive using Sqoop:

Step 1: Check 'employee' table in MySQL for its existence. Insert few records into it.

```
mysql> DESCRIBE employee;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id     | int(11)   | YES  |     | NULL    |       |
| emp_name   | varchar(100) | YES  |     | NULL    |       |
| emp_salary | int(11)   | YES  |     | NULL    |       |
| years_of_exp | int(11)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)

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

mysql> SELECT * FROM employee;
+-----+-----+-----+-----+
| emp_id | emp_name | emp_salary | years_of_exp |
+-----+-----+-----+-----+
| 101    | Amitabh | 20000      | 1            |
| 102    | Shahrukh | 10000      | 2            |
| 106    | Aamir   | 25000      | 1            |
| 107    | Salman  | 17500      | 2            |
| 103    | Akshay  | 11000      | 3            |
| 104    | Anubhav | 5000       | 4            |
| 105    | Pawan   | 2500       | 5            |
+-----+-----+-----+-----+
7 rows in set (0.01 sec)
```

Step 2: Use Sqoop import command to import data from MySQL table into Hive.

Here is the command to achieve this:

```
$sqoop import --connect jdbc:mysql://localhost/sample_db --table 'employee' --username 'root' -P --split-by 'years_of_exp' --columns emp_id, emp_name, years_of_exp --target-dir '/hive_output' --hive-import --create-hive-table --hive-table my_db.emp_info -m 1
```

There are few extra parameters here compared to HDFS import command. Let's focus on those. Here **we are interested in importing data of only three columns** out of four from employee table: employee id, employee name and years of experience. All fields are terminated by ',' in the file stored in HDFS. We have to use hive-import parameter to specify hive table information. We can also create table with create-hive-table parameter before starting the data transfer operation.

```
[acadgild@localhost bin]$ sqoop import --connect jdbc:mysql://localhost/sample_db --table 'employee' --username 'root' -P --split-by 'years_of_exp' --columns emp_id,emp_name,years_of_exp --fields-terminated-by ',' --target-dir '/hive_output' --hive-import --create-hive-table --hive-table my_db.emp_info -m 1
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCAT_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-23 05:27:10,890 INFO [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2018-01-23 05:27:15,842 INFO [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-23 05:27:15,850 INFO [main] tool.CodeGenTool: Beginning code generation
2018-01-23 05:27:16,306 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2018-01-23 05:27:16,389 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2018-01-23 05:27:16,405 INFO [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
```



```

2018-01-23 04:04:28,718 INFO [main] mapreduce.JobSubmitter: Submitting tokens for job: job_1516637375581_0013
2018-01-23 04:04:29,911 INFO [main] impl.YarnClientImpl: Submitted application application_1516637375581_0013 to ResourceMan
ager at /0.0.0.0:8032
2018-01-23 04:04:30,203 INFO [main] mapreduce.Job: The url to track the job: http://http://localhost:8088/proxy/application_
1516637375581_0013/
2018-01-23 04:04:30,206 INFO [main] mapreduce.Job: Running job: job_1516637375581_0013
2018-01-23 04:04:48,284 INFO [main] mapreduce.Job: Job job_1516637375581_0013 running in uber mode : false
2018-01-23 04:04:48,317 INFO [main] mapreduce.Job: map 0% reduce 0%
2018-01-23 04:05:10,123 INFO [main] mapreduce.Job: map 100% reduce 0%
2018-01-23 04:05:13,765 INFO [main] mapreduce.Job: Job job_1516637375581_0013 completed successfully
2018-01-23 04:05:14,240 ERROR [main] tool.ImportTool: Imported Failed: No enum constant org.apache.hadoop.mapreduce.JobCounte
r.MB_MILLIS_MAPS

```

Let's verify the output by examining the contents of table created in Hive.

```

hive> SELECT * FROM emp_info;
OK
101    Amitabh 1
102    Shahrukh    2
106    Aamir    1
107    Salman    2
103    Akshay    3
104    Anubhav    4
105    Pawan    5
Time taken: 0.106 seconds, Fetched: 7 row(s)
hive> █

```

As we can see the table 'emp_info' has been created and all records from MySQL table have been inserted into this Hive table successfully.

4. Export data from Hive to MySQL using Sqoop:

Step 1: Create a table in MySQL by name 'employee' with fewer columns than the existing one.

Let's recreate employee table dropping the salary field from it matching the Hive table structure.

```
mysql> CREATE TABLE emp_info
```

```
(emp_id int,
```

```
emp_name varchar(100),
```

```
years_of_exp int);
```

```

mysql> CREATE TABLE emp_info
-> (emp_id int,
-> emp_name varchar(100),
-> years_of_exp int
-> );
Query OK, 0 rows affected (0.06 sec)

```

```

mysql> SHOW TABLES;
+-----+
| Tables_in_sample_db |
+-----+
| emp_info             |
| employee             |
+-----+
2 rows in set (0.00 sec)

```

```

mysql> DESCRIBE emp_info;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(11) | YES | | NULL | |
| emp_name | varchar(100) | YES | | NULL | |
| years_of_exp | int(11) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

```

mysql> SELECT * FROM emp_info;
Empty set (0.00 sec)

```

Step 2: Use Sqoop export command to export data from Hive to MySQL.

Every Hive table will be allocated with physical space in HDFS to store input data files. Here in this case the data concerned with 'emp_info' table will be stored in path: /user/hive/warehouse/my_db.db/emp_info. We can see the contents of this directory with following command:

```
$ hadoop fs -ls /user/hive/warehouse/my_db.db/emp_info
```

```
[acadgild@localhost ~]$ hadoop fs -ls /user/hive/warehouse/my_db.db/emp_info
18/01/23 06:00:06 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r-- 1 acadgild supergroup          93 2018-01-23 05:59 /user/hive/warehouse/my_db.db/emp_info/part-m-00000
[acadgild@localhost ~]$
```

Here is the command used to achieve the export operation task completion:

```
sqoop export --connect jdbc:mysql://localhost/sample_db --table emp_info -username root -P --export-dir '/user/hive/warehouse/my_db.db/emp_info' --input-fields-terminated-by ',' --columns emp_id,emp_name,years_of_exp
```

```
[acadgild@localhost bin]$ sqoop export --connect jdbc:mysql://localhost/sample_db --table emp_info -username root -P --export-dir '/user/hive/warehouse/my_db.db/emp_info' --input-fields-terminated-by ',' --columns emp_id,emp_name,years_of_exp
Warning: /usr/local/sqoop/../hcatalog does not exist! HCatalog jobs will fail.
Please set $HCATALOG_HOME to the root of your HCatalog installation.
Warning: /usr/local/sqoop/../accumulo does not exist! Accumulo imports will fail.
Please set $ACCUMULO_HOME to the root of your Accumulo installation.
Warning: /usr/local/sqoop/../zookeeper does not exist! Accumulo imports will fail.
Please set $ZOOKEEPER_HOME to the root of your Zookeeper installation.
2018-01-23 06:18:13,255 INFO [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2018-01-23 06:18:16,986 INFO [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2018-01-23 06:18:16,987 INFO [main] tool.CodeGenTool: Beginning code generation
2018-01-23 06:18:17,448 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp_info` AS t LIMIT 1
2018-01-23 06:18:17,522 INFO [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `emp_info` AS t LIMIT 1
2018-01-23 06:18:17,537 INFO [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
2018-01-23 06:18:35,006 INFO [main] mapreduce.Job: Running job: job_1516637375581_0023
2018-01-23 06:19:30,448 INFO [main] mapreduce.Job: Job job_1516637375581_0023 running in uber mode : false
2018-01-23 06:19:30,491 INFO [main] mapreduce.Job: map 0% reduce 0%
2018-01-23 06:21:28,713 INFO [main] mapreduce.Job: map 25% reduce 0%
2018-01-23 06:21:30,230 INFO [main] mapreduce.Job: map 100% reduce 0%
2018-01-23 06:22:19,734 INFO [main] mapreduce.Job: Job job_1516637375581_0023 completed successfully
2018-01-23 06:22:23,043 ERROR [main] sqoop.Sqoop: Got exception running sqoop: java.lang.IllegalArgumentException: No enum co
```

Let's verify the output by fetching records of emp_info table in MySQL:

```
mysql> SELECT * FROM emp_info;
Empty set (0.00 sec)

mysql> SELECT * FROM emp_info;
+-----+-----+-----+
| emp_id | emp_name | years_of_exp |
+-----+-----+-----+
| 105 | Pawan | 5 |
| 101 | Amitabh | 1 |
| 102 | Shahrukh | 2 |
| 106 | Aamir | 1 |
| 107 | Salman | 2 |
| 103 | Akshay | 3 |
| 104 | Anubhav | 4 |
+-----+-----+-----+
7 rows in set (0.05 sec)
```

We can see the difference in the dataset of emp_info table before and after Sqoop export operation.