Big Data And Hadoop

Session 20 – Assignment 1

Problem Statement:

Perform UPSERT in Sqoop export.
Read a file from HDFS and based on the id field, perform UPSERT in MySQL table.
In UPSERT, if the field exists, then it is updated else it is inserted.

Solution:

1. We first start mysql service using the command **sudo service mysqld start** as follows:

```
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
[acadgild@localhost ~]$ sudo service mysqld start
[sudo] password for acadgild:
Starting mysqld:                                           [  OK  ]
[acadgild@localhost ~]$
```

2. Next, we start the command line interface for mysql as follows:

```
acadgild@localhost:~
File  Edit  View  Search  Terminal  Help
[acadgild@localhost ~]$ mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
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>
```

3. As we have to place the data from HDFS to MySQL, we should first have the table created already in MySQL. We will create the table in a database named **db1.** So we will first create the database using the command **create database db1;** as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
mysql> create database db1;
Query OK, 1 row affected (0.00 sec)

mysql>
```

4. We specify which database to work in using the command **use db1;**

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
mysql> use db1;
Database changed
mysql>
```

5. We now create the table named **employee** using the command
   Create table employee
   (
   id int,
   name varchar(1000),
   salary int,
   primary key (id)
   );
   Here, we have made **id** to be the primary key of the table.

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
mysql> create table employee
    -> (
    -> id int,
    -> name varchar(1000),
    -> salary int,
    -> primary key (id)
    -> );
Query OK, 0 rows affected (0.01 sec)

mysql>
Abhilasha - Fil...  *Sqoopcomma...  acadgild@local...  acadgild@local...  Browsing HDF...  acadgild@local...
```

6. We can see the schema of the table created using **describe** command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
mysql> describe employee;
+--------+---------------+------+-----+---------+-------+
| Field  | Type          | Null | Key | Default | Extra |
+--------+---------------+------+-----+---------+-------+
| id     | int(11)       | NO   | PRI | 0       |       |
| name   | varchar(1000) | YES  |     | NULL    |       |
| salary | int(11)       | YES  |     | NULL    |       |
+--------+---------------+------+-----+---------+-------+
3 rows in set (0.01 sec)

mysql>
Abhilasha - Fil...  *Sqoopcomma...  acadgild@local...  acadgild@local...  Browsing HDF...  acadgild@local...
```

7. We have the input file present locally at **/home/acadgild/Abhilasha.** Name of the file is **employee.txt**. It can be list using **ls** command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ pwd
/home/acadgild/Abhilasha
[acadgild@localhost Abhilasha]$ ls -l
total 8
-rw-rw-r--. 1 acadgild acadgild   39 Nov  5 13:04 employee.txt
drwxrwxr-x. 7 acadgild acadgild 4096 Nov  4 16:04 Project2.2
[acadgild@localhost Abhilasha]$
```

8. We put this file on HDFS at **/abhilasha/sqoop** using the **put** command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ hadoop fs -put /home/acadgild/Abhilasha/employee.txt /abhilasha/sqoop
17/11/05 13:10:49 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
[acadgild@localhost Abhilasha]$
```

9. Listing of files at HDFS shows **employee.txt** placed successfully.

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ hadoop fs -ls /abhilasha/sqoop
17/11/05 13:11:30 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
Found 1 items
-rw-r--r--   1 acadgild supergroup         39 2017-11-05 13:10 /abhilasha/sqoop/employee.txt
[acadgild@localhost Abhilasha]$
```

10. The content of the file is shown using **cat** command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ hadoop fs -cat /abhilasha/sqoop/employee.txt
17/11/05 13:12:11 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
1,ABCD,10000
2,EFGH,20000
3,IGKL,30000
[acadgild@localhost Abhilasha]$
```

The file has three records.

11. Now we perform the sqoop operation to put data from HDFS into MySQL as follows:
    The input parameters are:
    a. Connect: This specifies the mysql system to be connected to. Here its value is **jdbc:mysql//localhost/db1**
       Here, **db1** is name of the database to be used.

b. Username: This is to specify the username to be used to access MySQL. Here its value is **root.**
c. P: This is for password. Using P doesn't need to type the password in the sqoop command, instead it takes it from standard input stream.
d. Table: This is to specify the table name, our input is **employee**.
e. Export-dir: This is the path of export directory in HDFS. Its value is **/abhilasha/sqoop/**
f. Update-key: This is the column name that will be used for update. Our column for update is **id**.
g. Update-mode: We need the command to perform UPSERT that is if a new record comes, if should update the previous one else it should insert the record. Hence, we have used the mode to be **allowinsert**.
h. M: This is to specify number of mappers to use. We have used **1** in our case.

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ sqoop export --connect jdbc:mysql://localhost/db1 --username 'root' -P --table 'employee' --e
xport-dir '/abhilasha/sqoop/' --update-key id --update-mode allowinsert -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.
2017-11-05 13:38:34,288 INFO  [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2017-11-05 13:38:35,846 INFO  [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2017-11-05 13:38:35,846 INFO  [main] tool.CodeGenTool: Beginning code generation
2017-11-05 13:38:36,259 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2017-11-05 13:38:36,301 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2017-11-05 13:38:36,310 INFO  [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
Note: /tmp/sqoop-acadgild/compile/cf31d07f6f37c4591b7f5eac069de946/employee.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
2017-11-05 13:38:38,476 INFO  [main] orm.CompilationManager: Writing jar file: /tmp/sqoop-acadgild/compile/cf31d07f6f37c4591b
7f5eac069de946/employee.jar
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: MySQL Connector upsert functionality is using INSERT ON
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: DUPLICATE KEY UPDATE clause that relies on table's unique key.
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: Insert/update distinction is therefore independent on column
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: names specified in --update-key parameter. Please see MySQL
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: documentation for additional limitations.
2017-11-05 13:38:38,495 INFO  [main] mapreduce.ExportJobBase: Beginning export of employee
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-11-05 13:38:38,954 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
2017-11-05 13:38:38,967 INFO  [main] Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
```

12. The output of this command execution results in the **employee** table getting populated in MySQL as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
mysql> select * from employee;
+------+------+--------+
| id   | name | salary |
+------+------+--------+
|    1 | ABCD |  10000 |
|    2 | EFGH |  20000 |
|    3 | IGKL |  30000 |
+------+------+--------+
3 rows in set (0.00 sec)

mysql>
```

13. Now to verify that the UPSERT works, we will modify the input file locally and place it on HDFS and re-run the sqoop command. The updated input file on the local file system is as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ pwd
/home/acadgild/Abhilasha
[acadgild@localhost Abhilasha]$ cat employee.txt
1,ABCD,10000
2,EFGH,50000
3,IGKL,30000
4,MNOP,40000
[acadgild@localhost Abhilasha]$
```

Please note that here, we have modified the salary of employee with **id 2.** Earlier, its salary was 20000, now we have made it to 50000. Also we have added a new entry for employee id **4.**

14. Before we place this file on HDFS, we remove the old file using **rm** command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ hadoop fs -rm /abhilasha/sqoop/employee.txt
17/11/05 13:28:18 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
17/11/05 13:28:19 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval =
 0 minutes.
Deleted /abhilasha/sqoop/employee.txt
[acadgild@localhost Abhilasha]$
```

15. Now, we place the updated file on HDFS as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ hadoop fs -put /home/acadgild/Abhilasha/employee.txt /abhilasha/sqoop
17/11/05 13:33:22 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
[acadgild@localhost Abhilasha]$
```

16. The content of file on HDFS is seen using **cat** command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ hadoop fs -cat /abhilasha/sqoop/employee.txt
17/11/05 13:33:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java cl
asses where applicable
1,ABCD,10000
2,EFGH,50000
3,IGKL,30000
4,MNOP,40000
[acadgild@localhost Abhilasha]$
```

17. We now re-run the sqoop command as follows:

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
[acadgild@localhost Abhilasha]$ sqoop export --connect jdbc:mysql://localhost/db1 --username 'root' -P --table 'employee' --e
xport-dir '/abhilasha/sqoop/' --update-key id --update-mode allowinsert -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.
2017-11-05 13:38:34,288 INFO  [main] sqoop.Sqoop: Running Sqoop version: 1.4.5
Enter password:
2017-11-05 13:38:35,846 INFO  [main] manager.MySQLManager: Preparing to use a MySQL streaming resultset.
2017-11-05 13:38:35,846 INFO  [main] tool.CodeGenTool: Beginning code generation
2017-11-05 13:38:36,259 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2017-11-05 13:38:36,301 INFO  [main] manager.SqlManager: Executing SQL statement: SELECT t.* FROM `employee` AS t LIMIT 1
2017-11-05 13:38:36,310 INFO  [main] orm.CompilationManager: HADOOP_MAPRED_HOME is /usr/local/hadoop-2.6.0
Note: /tmp/sqoop-acadgild/compile/cf31d07f6f37c4591b7f5eac069de946/employee.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
2017-11-05 13:38:38,476 INFO  [main] orm.CompilationManager: Writing jar file: /tmp/sqoop-acadgild/compile/cf31d07f6f37c4591b
7f5eac069de946/employee.jar
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: MySQL Connector upsert functionality is using INSERT ON
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: DUPLICATE KEY UPDATE clause that relies on table's unique key.
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: Insert/update distinction is therefore independent on column
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: names specified in --update-key parameter. Please see MySQL
2017-11-05 13:38:38,484 WARN  [main] manager.MySQLManager: documentation for additional limitations.
2017-11-05 13:38:38,495 INFO  [main] mapreduce.ExportJobBase: Beginning export of employee
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
ticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2017-11-05 13:38:38,954 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
2017-11-05 13:38:38,967 INFO  [main] Configuration.deprecation: mapred.jar is deprecated. Instead, use mapreduce.job.jar
```

18. On running the query **select * from employee** in MySQL, we see the data below.

```
acadgild@localhost:~/Abhilasha
File  Edit  View  Search  Terminal  Help
mysql> select * from employee;
+----+------+--------+
| id | name | salary |
+----+------+--------+
|  1 | ABCD |  10000 |
|  2 | EFGH |  50000 |
|  3 | IGKL |  30000 |
|  4 | MNOP |  40000 |
+----+------+--------+
4 rows in set (0.00 sec)

mysql>
Abhilasha - Fil...  *Sqoopcomma...  acadgild@local...  acadgild@local...  Browsing HDF...  acadgild@local...
```

Here, we observe that the records with ids **1** and **3** have not been re-inserted.

The record with id **2** has its salary updated, hence, update operation is performed.

Record with id **4** is a new record, was not present earlier, and now has been inserted.

This shows that UPSERT is performed.