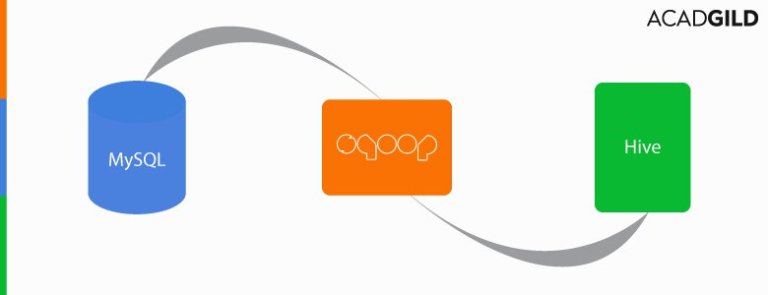
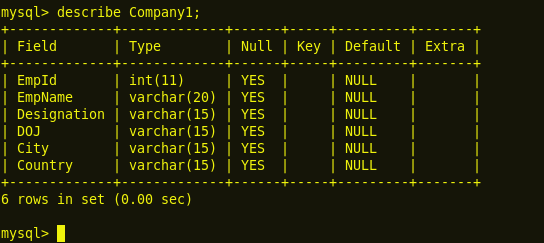
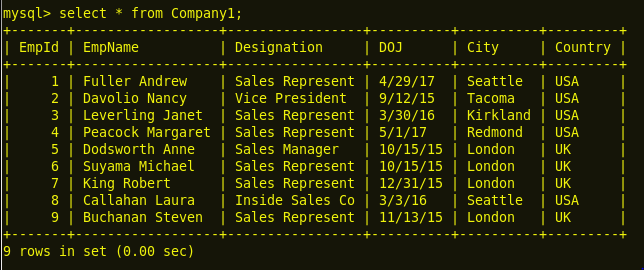
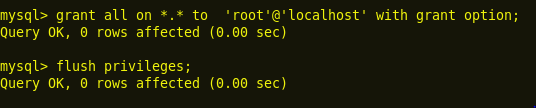
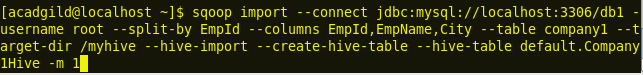
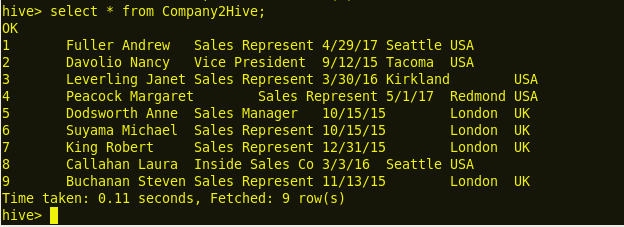
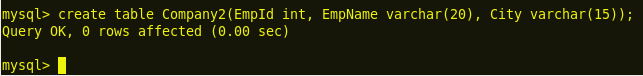
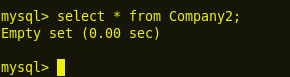
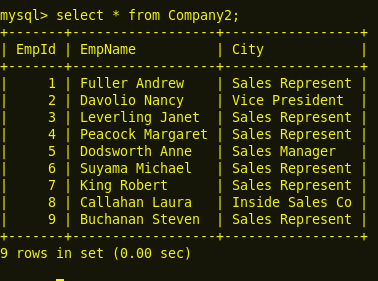
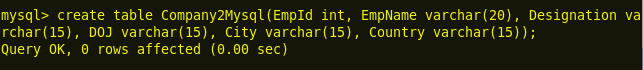
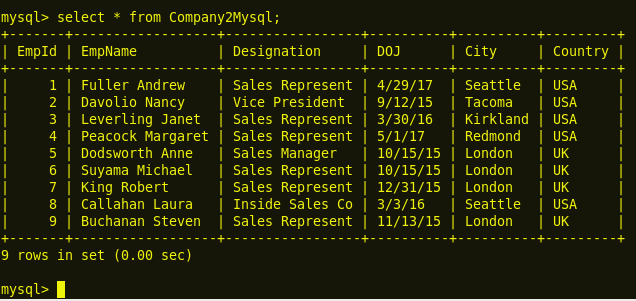
**How to Import Data from MySQL to Hive Using Sqoop**



In this blog, we will discuss how we can efficiently import data from MySQL to Hive using Sqoop. But before we move ahead, we recommend you to take a look at some of the blogs that we put out previously on Sqoop and its functioning.  
[Beginners Guide for Sqoop](https://acadgild.com/blog/beginners-guide-for-sqoop/)  
[Sqoop Tutorial for Incremental Imports](https://acadgild.com/blog/sqoop-tutorial-incremental-imports/)  
[Export Data from Hive to MongoDB](https://acadgild.com/blog/how-to-export-data-from-hive-to-mongodb/)  
[Importing Data from MySQL to HBase](https://acadgild.com/blog/how-to-import-table-from-mysql-to-hbase/)  
In this example, we will be using the table **Company1** which is already present in the MySQL database.  
We can use the **describe** command to see the schema of the **Company1** table.   
**Describing theTable Schema**  
describe Company1;  
  
The DESCRIBE TABLE command lists the following information about each column:

* Column name
* Type schema
* Type name
* Length
* Scale
* Nulls (Yes/No)

**Displaying the Table Contents**  
We can use the following commands to display all the columns present in the table **Company1.**  
select \* from Company1;  
  
**Granting All Permissions to Root and Flush the Privileges**  
We can use the following command to grant a superuser the permission to root.  
grant all on \*.\* to  ‘root’@’localhost’ with grant option;  
flush privileges;  
  
MySQL privileges are critical to the utility of the system as they allow each of their users to access and utilize only those areas that are needed to perform their work functions. This is meant to prevent a user from accidentally accessing an area which they should not have access to.   
Additionally, this adds to the security of the MySQL server.   
Whenever someone connects to a MySQL server, their identities are determined by the host used to connect them and the user name specified. With this information, the server grants privileges based upon the identity determined.  
The above step finishes the MySQL part.  
Now, let us open a new terminal and enter Sqoop commands to import data from MySQL to Hive table.  
**I. A Sqoop command is used to transfer selected columns from MySQL to Hive.**  
Now, use the following command to import selected columns from the MySQL **Company1** table to the Hive **Company1Hive** table.  
sqoop import –connect jdbc:mysql://localhost:3306/db1 -username root –split-by EmpId –columns EmpId,EmpName,City –table **company1** –target-dir /myhive –hive-import –create-hive-table –hive-table **default.Company1Hive** -m 1  
  
The above Sqoop command will create a new table with the name **Company1Hive** in the Hive **default** database and transfer the 3 mentioned column (EmpId, EmpName and City) values from the MySQL table **Company1** to the Hive table **Company1Hive**.  
**Displaying the Contents of the Table Company1Hive**  
Now, let us see the transferred contents in the table **Company1Hive.**  
select \* from Company1Hive;  
  
**II. Sqoop command for transferring a complete table data from MySQL to Hive.**  
In the previous example, we transferred only the 3 selected columns from the MySQL table **Company1** to the Hive default database table **Company1Hive**.   
Now, let us go ahead and transfer the complete table from the table **Company1** to a new Hive table by following the command given here:  
sqoop import –connect jdbc:mysql://localhost:3306/db1 -username root –table Company1 –target-dir /myhive –hive-import –create-hive-table –hive-table default.Company2Hive -m 1  
https://s3.amazonaws.com/acadgildsite/wordpress_images/bigdatadeveloper/Transfer+Data+Between+Mysql+and+Hive+Using+Sqoop/1.7%20Sqoop%20command%20to%20import%20complete%20table%20from%20mysql%20to%20hive.PNG  
The above given Sqoop command will create a new table with the name **Company2Hive** in the Hive **default** database and will transfer all this data from the MySQL table **Company1** to the Hive table **Company2Hive**.  
**In Hive.**  
Now, let us see the transferred contents in the table **Company2Hive.**  
select \* from Company2Hive;  
  
We can observe from the above screenshot that we have successfully transferred these table contents from the MySQL to a Hive table using Sqoop.  
Next, we will do a vice versa job, i.e, we will export table contents from the Hive table to the MySQL table.   
**III.** **Export command for transferring the selected columns from Hive to MySQL.**  
In this example we will transfer the selected columns from Hive to MySQL. For this, we need to create a table before transferring the data from Hive to the MySQL database. We should follow the command given below to create a new table.  
create table Company2(EmpId int, EmpName varchar(20), City varchar(15));  
  
The above command creates a new table named **Company2** in the MySQL database with three columns: **EmpId**, **EmpName**, and **City**.  
Let us use the select statement to see the contents of the table **Company2**.  
Select \* from Company2;  
  
We can observe that in the screenshot shown above, the table contents are empty. Let us use the Sqoop command to load this data from Hive to MySQL.  
sqoop export –connect jdbc:mysql://localhost/db1 -username root –P –columns EmpId,EmpName,City –table Company2 –export-dir /user/hive/warehouse/company2hive –input-fields-terminated-by ‘\001’ -m 1  
https://s3.amazonaws.com/acadgildsite/wordpress_images/bigdatadeveloper/Transfer+Data+Between+Mysql+and+Hive+Using+Sqoop/2.3%20Sqoop%20command%20to%20export%20selected%20columns%20from%20hive%20to%20mysql.PNG  
The Sqoop command given above will transfer the 3 mentioned column (EmpId, EmpName, and City) values from the Hive table **Company2Hive** to the MySQL table **Company2**.  
**Displaying the Contents of the Table Company2**  
Now, let us see the transferred contents in the table **Company2.**  
select \* from Company2;  
  
We can observe from the above image that we have now successfully transferred data from Hive to MySQL.  
**IV. Export command for transferring the complete table data from Hive to MySQL.**  
Now, let us transfer this complete table from the Hive table Company2Hive to a MySQL table by following the command given below:  
create table Company2Mysql(EmpId int, EmpName varchar(20), Designation varchar(15), DOJ varchar(15), City varchar(15), Country varchar(15));  
  
Let us use the select statement to see the contents of the table **Company2Msyql**.  
select \* from Company2Mysql;  
https://i1.wp.com/s3.amazonaws.com/acadgildsite/wordpress_images/bigdatadeveloper/Transfer+Data+Between+Mysql+and+Hive+Using+Sqoop/2.7%20select%20all%20from%20Company2Mysql%20%20before%20sqoop%20export.PNG?zoom=0.9024999886751175&resize=317%2C57&ssl=1  
We observe in the screenshot given above that the table contents are empty. Let us use a Sqoop command to load this data from Hive to MySQL.  
sqoop export –connect jdbc:mysql://localhost/db1 –username root –P  –table Company2Mysql –export-dir /user/hive/warehouse/company2hive –input-fields-terminated-by ‘\001’ -m 1  
https://s3.amazonaws.com/acadgildsite/wordpress_images/bigdatadeveloper/Transfer+Data+Between+Mysql+and+Hive+Using+Sqoop/2.6%20Sqoop%20command%20to%20export%20complete%20table%20from%20hive%20to%20mysql.PNG  
The above given Sqoop command will transfer the complete data from the Hive table **Company2Hive** to the MySQL table **Company2Mysql**.  
**Displaying the Contents of the Table Company2Mysql**  
Now, let us see the transferred contents in the table **Company2Mysql**.  
select \* from Company2Mysql;  
  
We can see here in the screenshot how we have successfully exported table contents from Hive to MySQL. We can follow the above steps to transfer this data between Apache Hive and the structured databases.