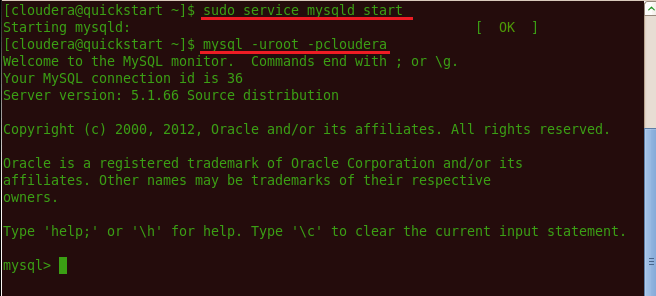
**Incremental Import in Sqoop To Load Data From Mysql To HDFS**

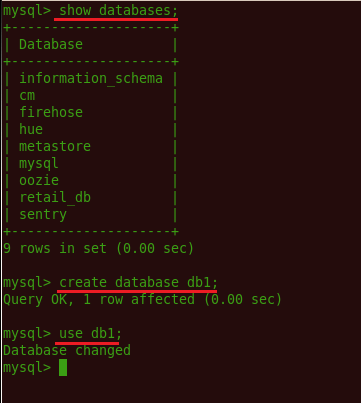
This post covers the advanced topics in Sqoop, beginning with ways to import the recently updated data in MySQL table into HDFS. If you are new to Sqoop.

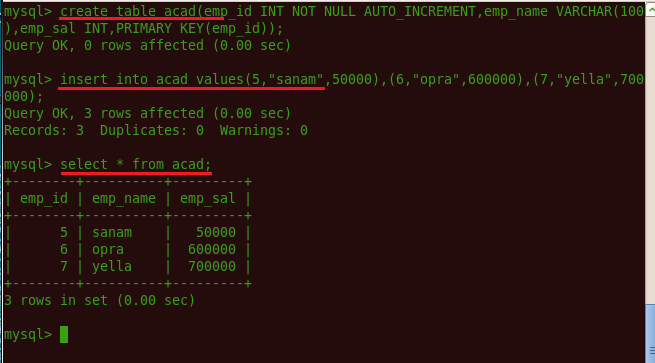
***Note: Make sure your Hadoop daemons are up and running. This real-world practice is done in Cloudera system.***  
Sqoop supports two types of incremental imports: *append* and *lastmodified*. You can use the *–incremental* argument to specify the type of incremental import to perform.  
You should specify the append mode when importing a table, where new rows are continually added with increasing row id values. You must specify the column containing the row’s id with *–check-column*. Sqoop imports rows where the check column has a value greater than the one specified with –last-value.  
An alternate table update strategy supported by Sqoop is called *lastmodified* mode. This should be used when rows of the source table is updated, and each such update will set the value of a last-modified column to the current timestamp. Rows where the check column holds a timestamp more recent than the timestamp specified with *–last-value* are imported.  
At the end of an incremental import, the value which should be specified as *–last-value* for a subsequent import is printed to the screen. When running a subsequent import, you should specify*–last-value* in this way to ensure you import only the new or updated data. This is handled automatically by creating an incremental import as a saved job, which is the preferred mechanism for performing a recurring incremental import.

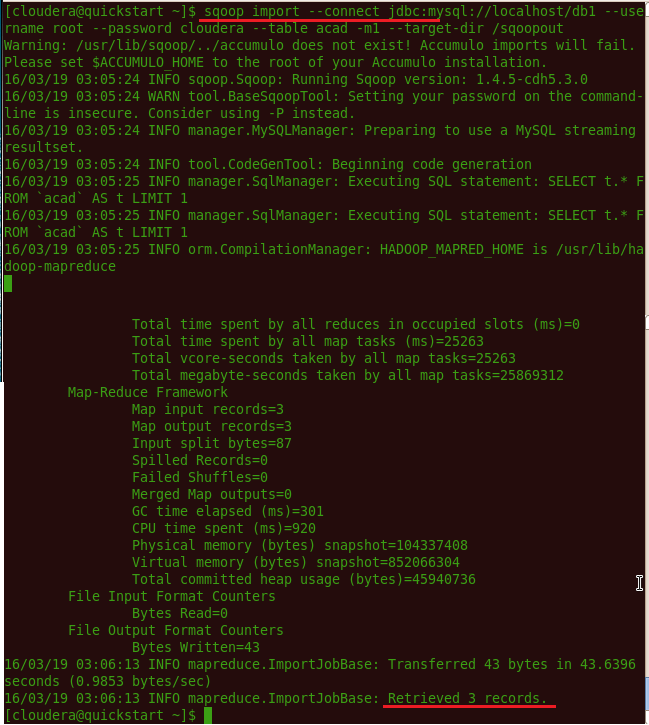
### [Test your command in Linux Here](https://acadgild.com/blog/linux-quiz-beginners/)

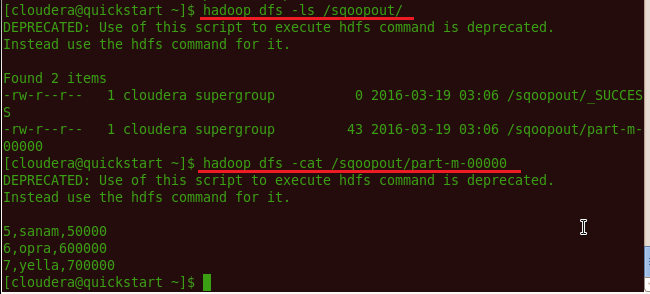
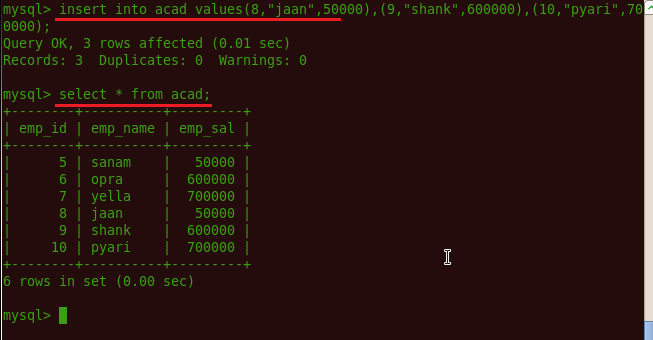
**Let’s see with an example, step by step procedure to perform incremental import from MySQL table.**  
***Step:1***  
**Start the MySQL service with the below command:**  
sudo service mysqld start  
**And enter MySQL shell using the below command:**  
mysql -u root -p cloudera  


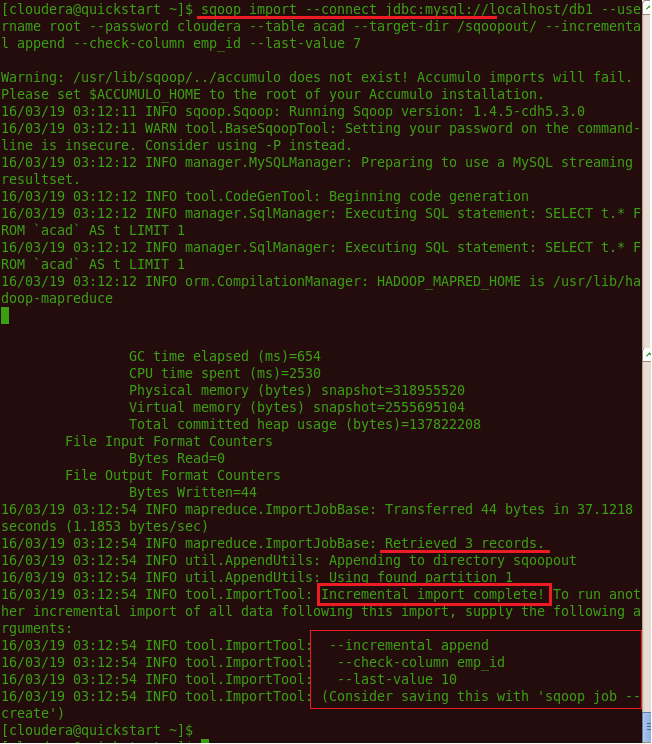
Bottom of Form

***Step:2***  
**Command to list database if already existing:**  
show databases;  
**Command to create a new database:**  
create database db1;  
**Command for using the database:**  
use db1;  


***Step:3***  
**Also creating table, inserting values inside table is done using the following syntax.**  
create table <table name>(column name1, column name 2);  
insert into <table name> values(column1 value1, column2 value1);  
insert into <table name> values(column1 value2, column2 value2);  


***Step:4***  
**Since the data is present in table of MySQL and Sqoop is up and running, we will fetch the data using following command.**  
Sqoop import –connect jdbc:mysql://localhost/db1 –username root –password cloudera –table acad -m1 –tagret-dir /sqoopout  
  
As confirmation of the result, you can see in the image, the comment as Retrieved 3 records.  
***Step:5***  
**Let’s checkout whether any data is stored in HDFS. This can be done by giving the following command in the terminal.**  
**Hadoop dfs -ls /sqoopout/**  
This shows that part file has been created in our target directory. Now, by the following command we view the content inside part file.  
hadoop dfs -cat /sqoopout/part-m-0000

  
This confirms the data inside MySQL has come inside the HDFS. But what if the data inside MySQL is increasing and has more number of rows present now than earlier?  
**The following steps will shed some light on the same.**  
***Step:1***  
Let’s manually insert few extra values in mysql / acad table.  
  
Now, the following command with little few extra syntax will help you feed only the new values in the table**acad.**

***Step:2***  
The following syntax is used for the incremental option in Sqoop import command.  
**–incremental <mode>**  
**–check-column <column name>**  
**–last value <last check column value>**  
  
As you can see in above image, 3 more records have been retrieved and the incremental import is now complete.  
Along with message for next incremental import, you need to give last value as**10.**  
***Step:3***  
Now let’s check and confirm the new data inside HDFS.

