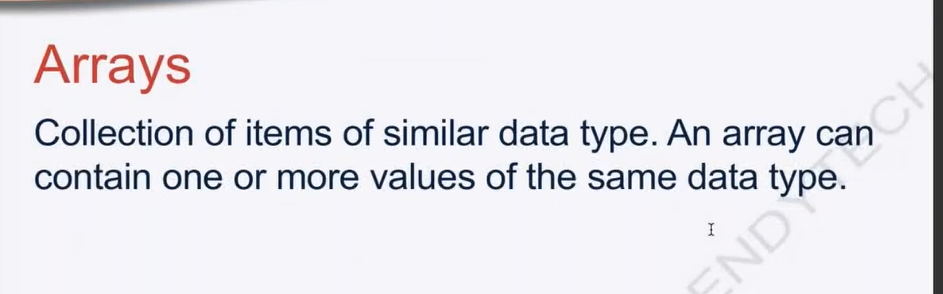
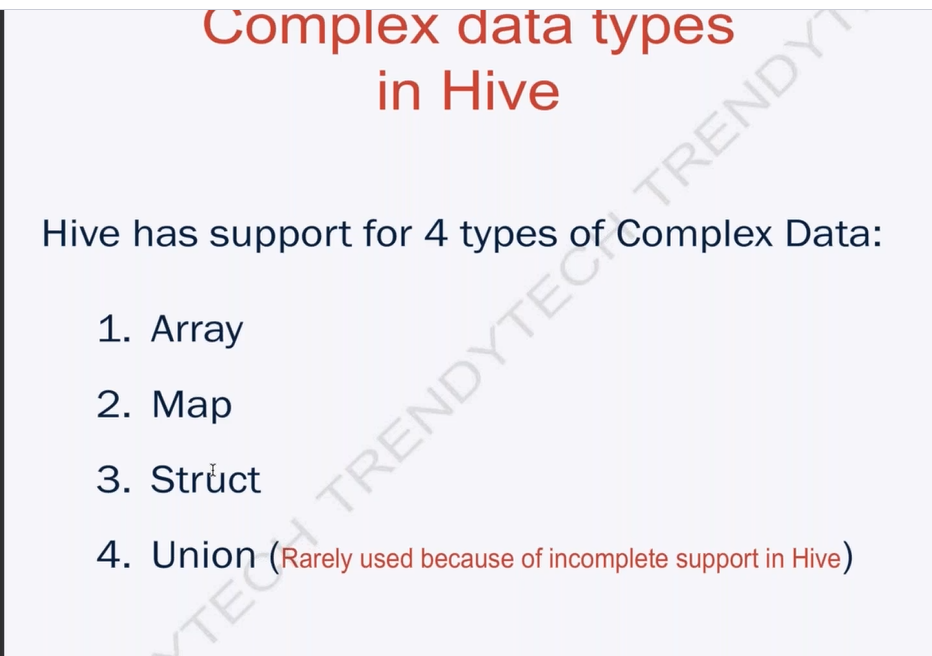
**Complex Data type in Hive**

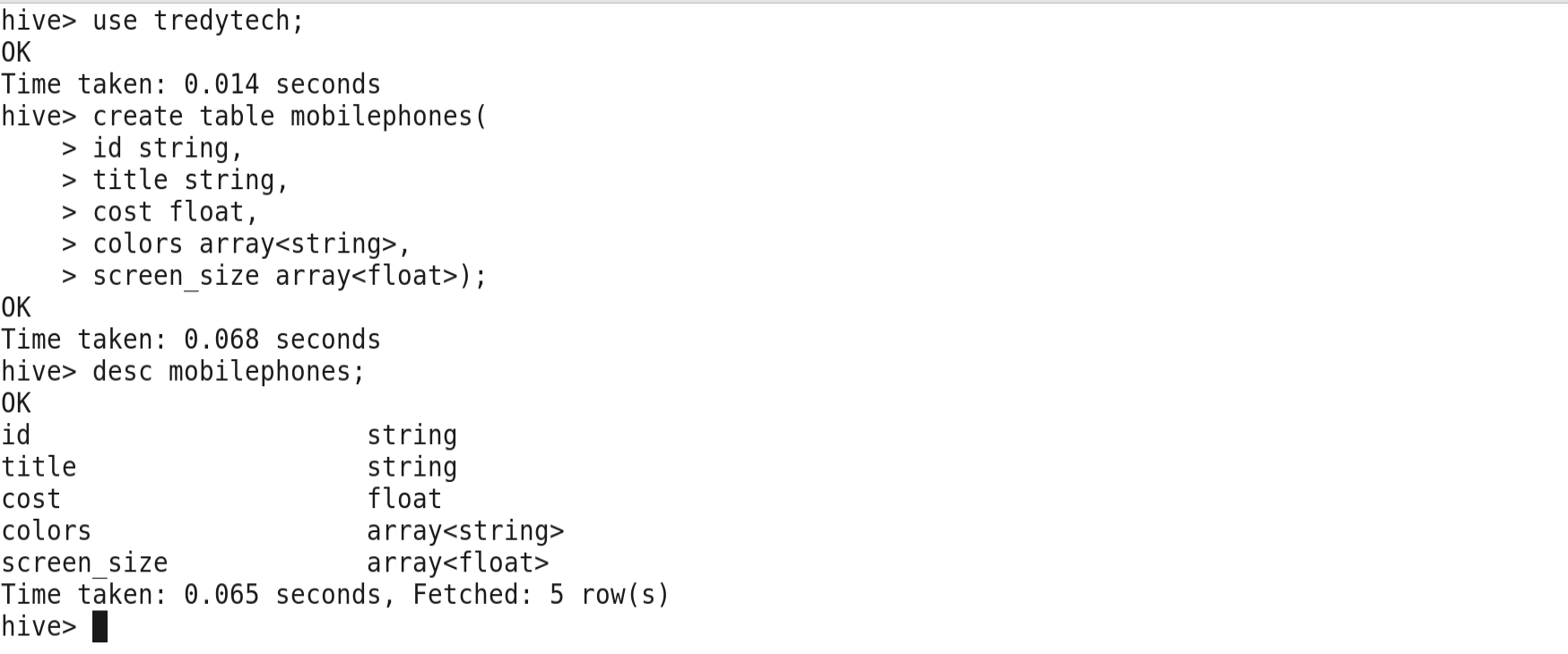
>>Complex data type means holds more than a single value

Ex:- Array

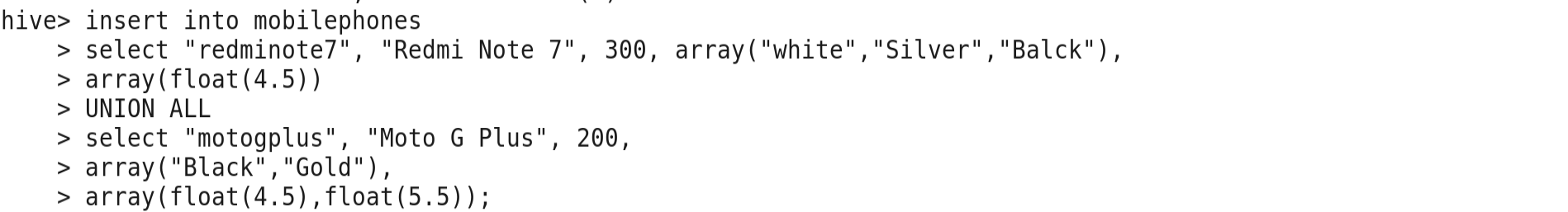


Array contain the Homogenous type of Data

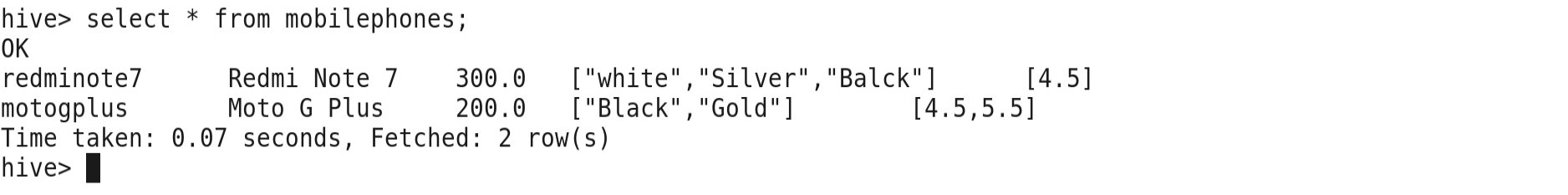
>>Now create a table of the mobile-phones.



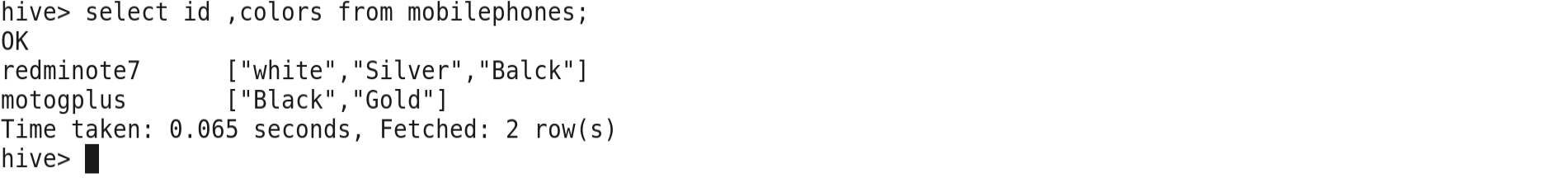
>>Insert data into to the mobilephones table



>>Display data from mobilephones table



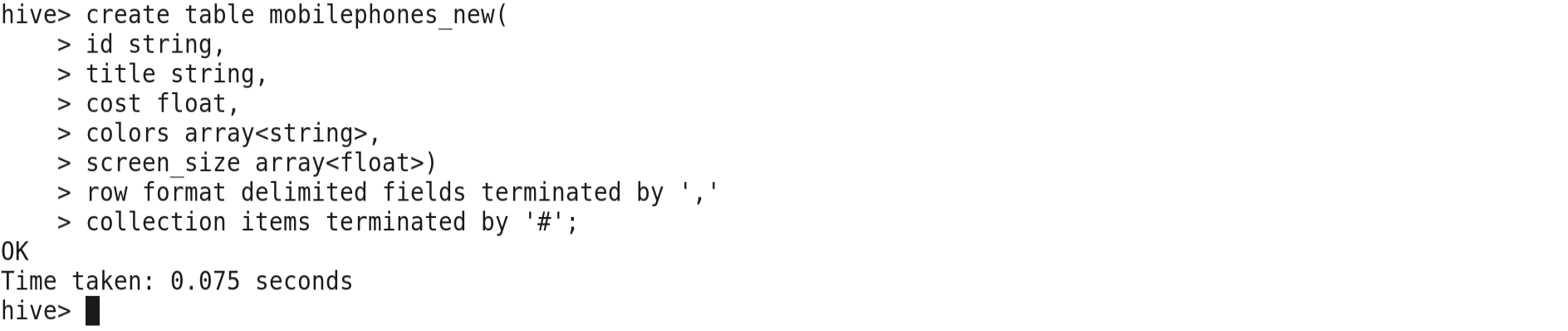
>>Display id and colors of the mobilephone table



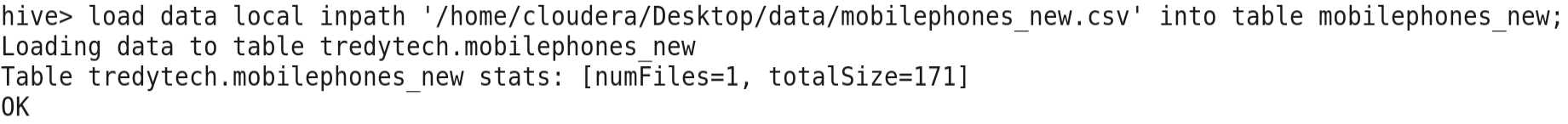
>>Display id and colors [0] of the column One



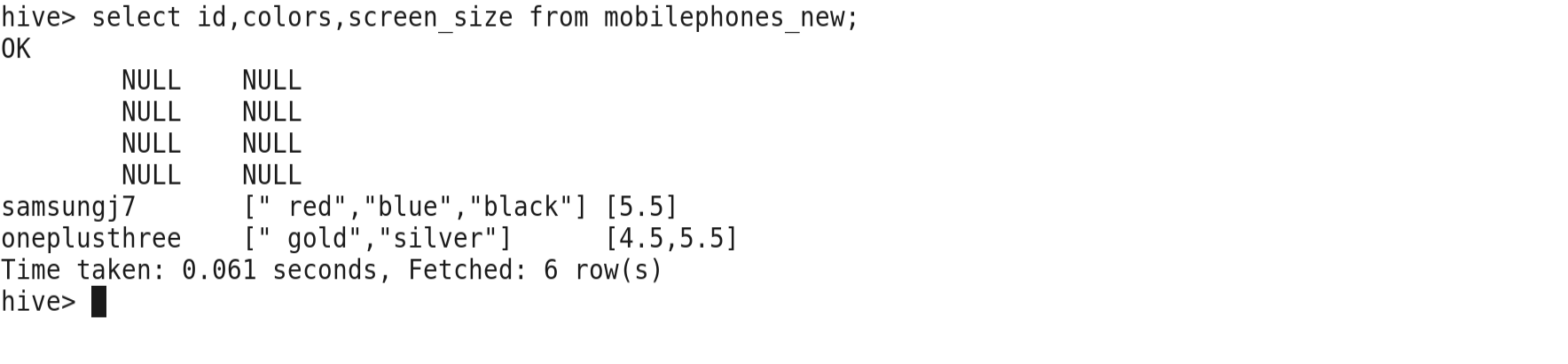
>>**Now create a New Table mobileophones\_new**



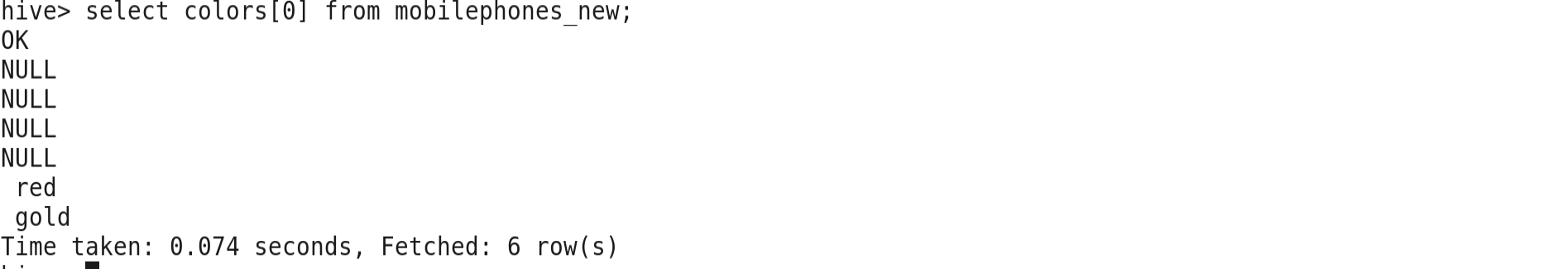
>>load data into the new table mobilephones\_new from mobilephones\_new.csv file



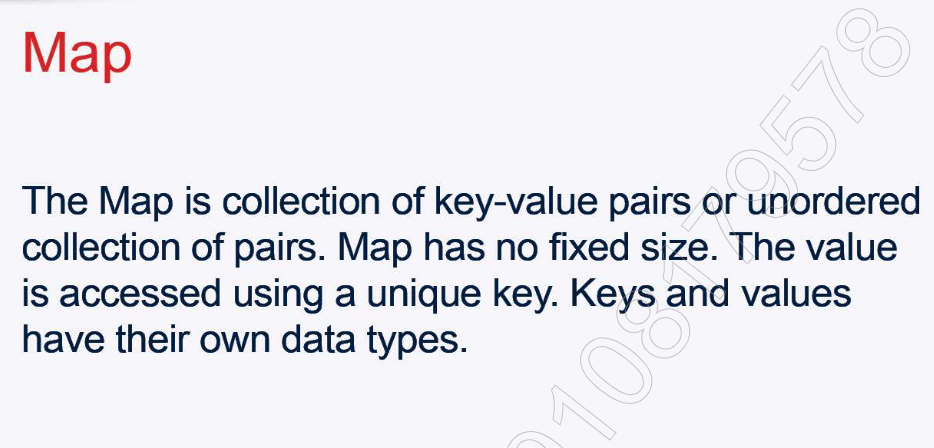
>>Display all Records of the array Element of Mobilephones\_new



>>Display Records of 1st column Element of mobilephones\_new



**Map**

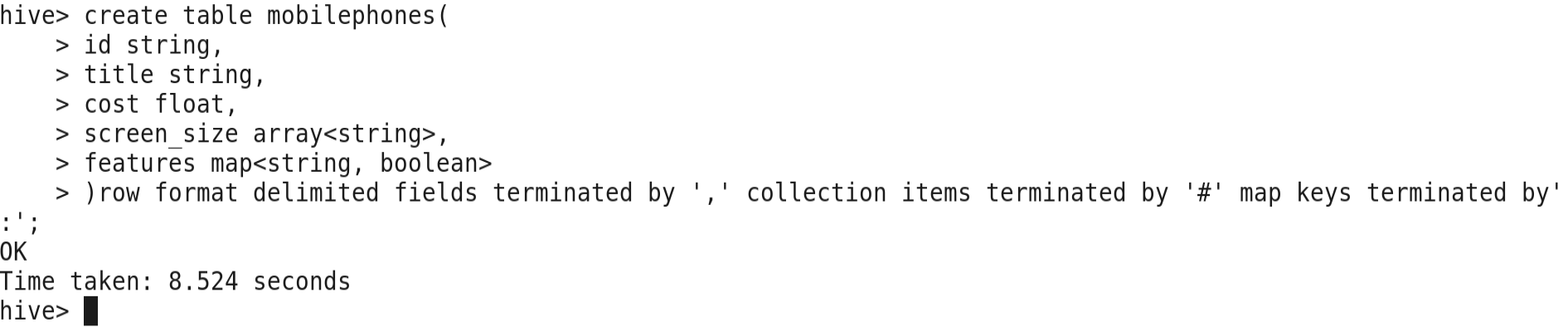


**Key Value**

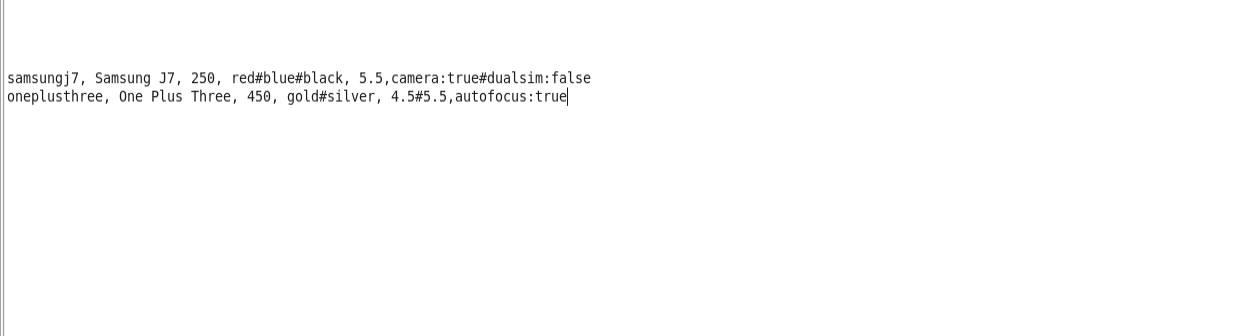
Roll\_no Name

1. Shilpi
2. Sushant

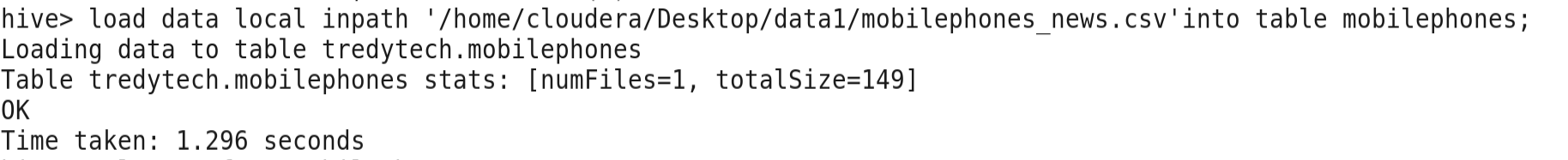
>>Now Create a Map New Table Mobilephones;



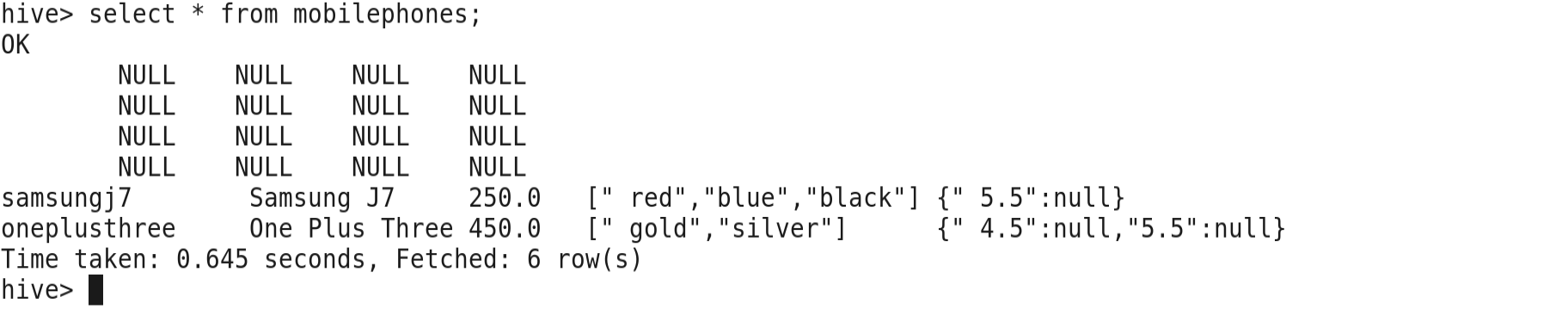
>>First of the Edit the mobilephones\_new.csv file and Load into the mobilephones table of the map.



>>Load data into the Table

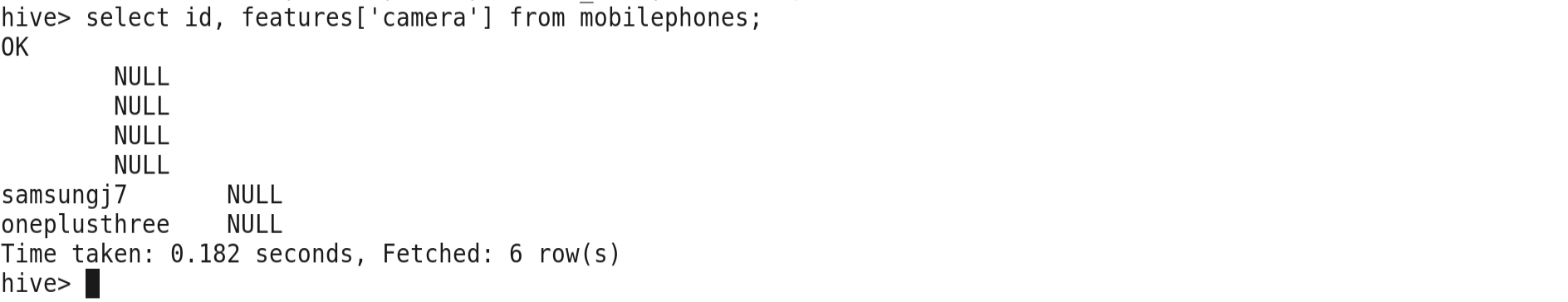


>>Check the all the data of the Tables of Map



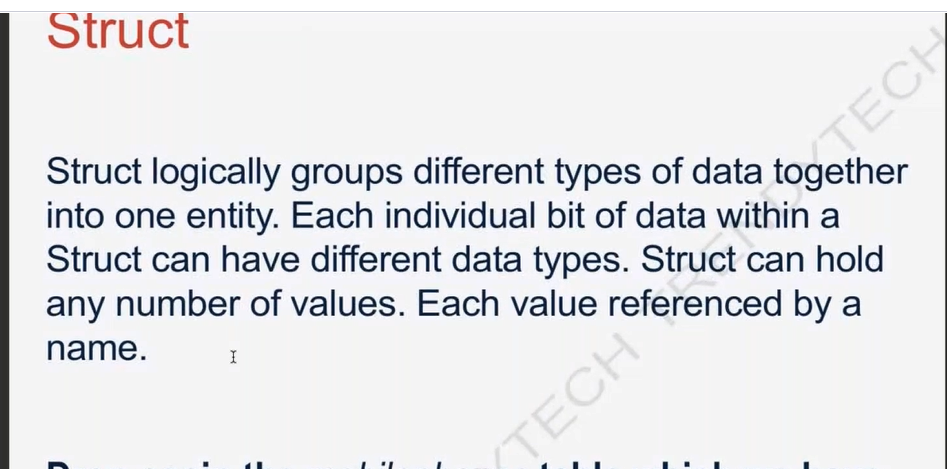
>> Select some features of the table.





**Struct**

>>



Ex:-

Student Struct

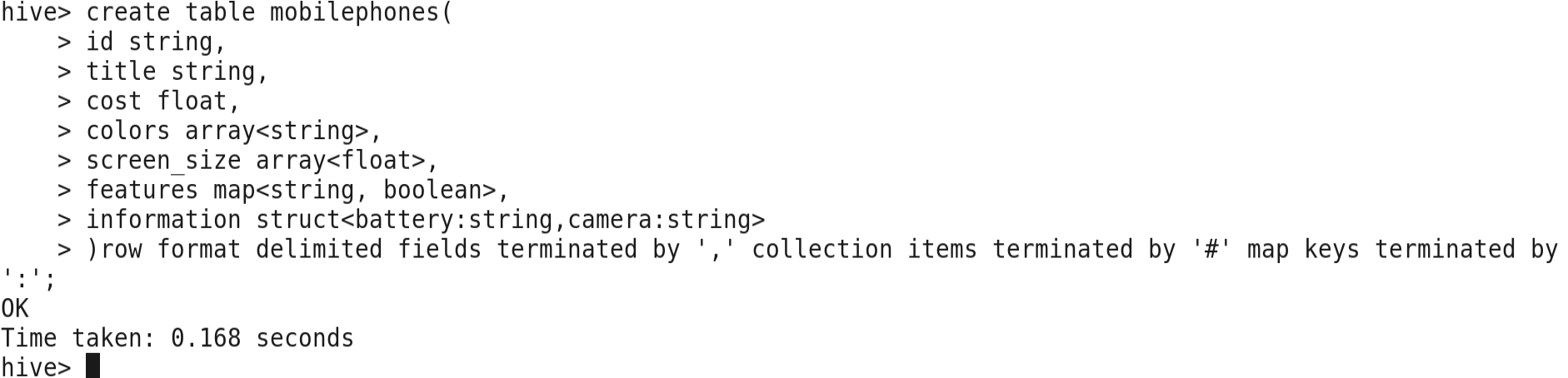
Student\_id 101

Student\_marks 001

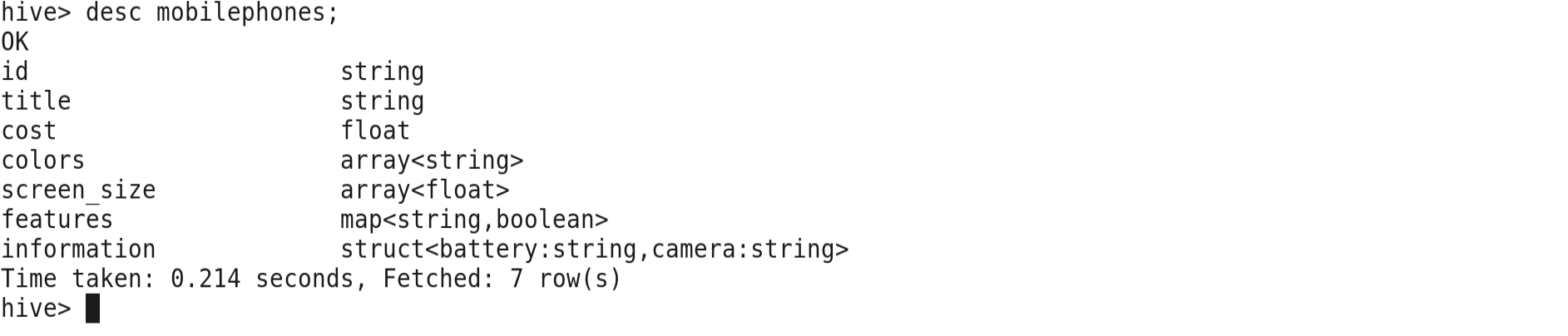
Student\_roll\_no 404

>>First of all drop the Previous table of Mobilephones of the Map

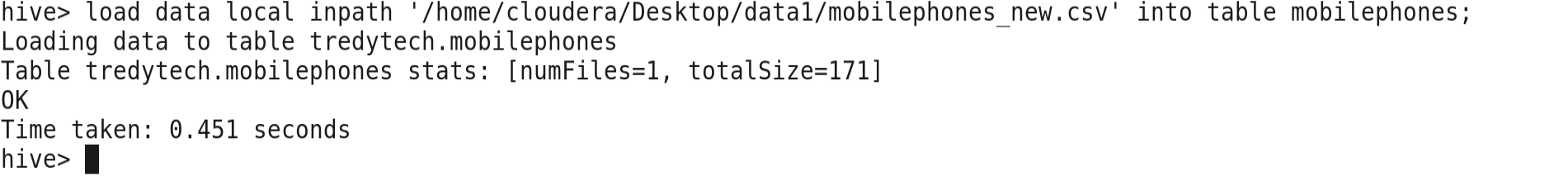
>>**Create a new table Mobilephones with different Struct Data Types.**



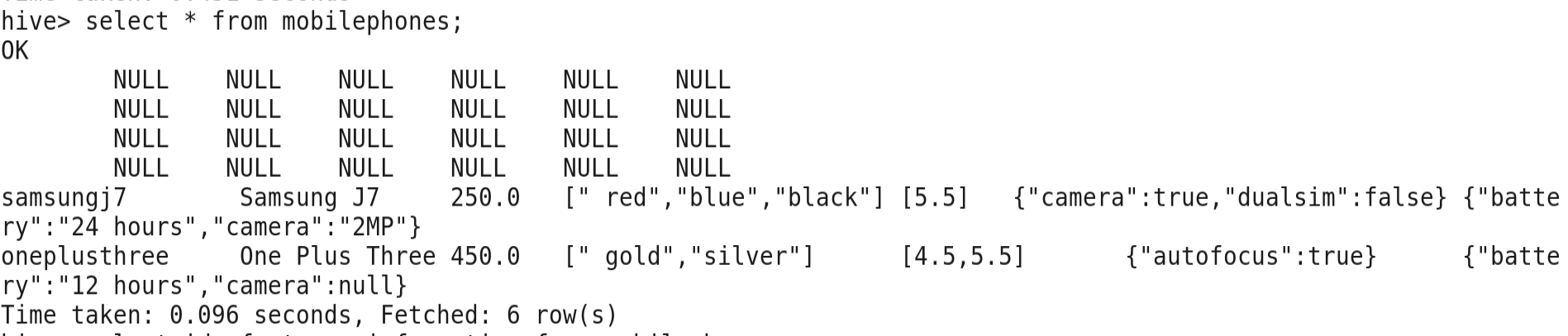
>>desc table of Struct Table



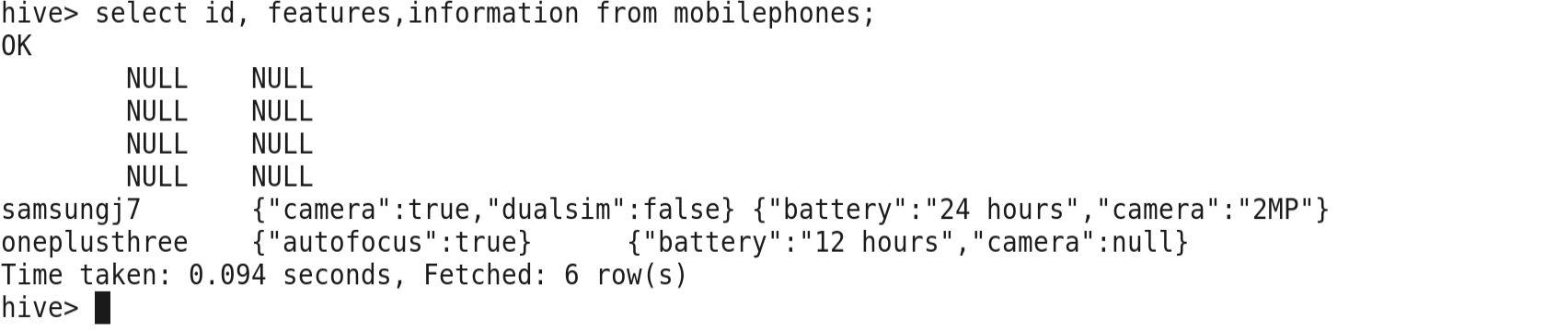
>>Load data into the Mobilephones table of Struct



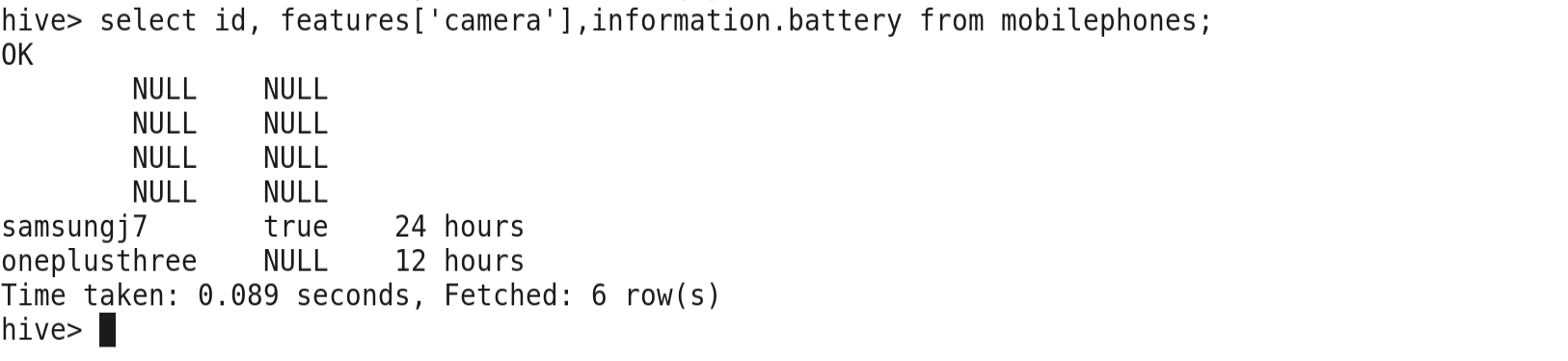
>>select from the table of the struct .



>>select some items of the Struct Table.

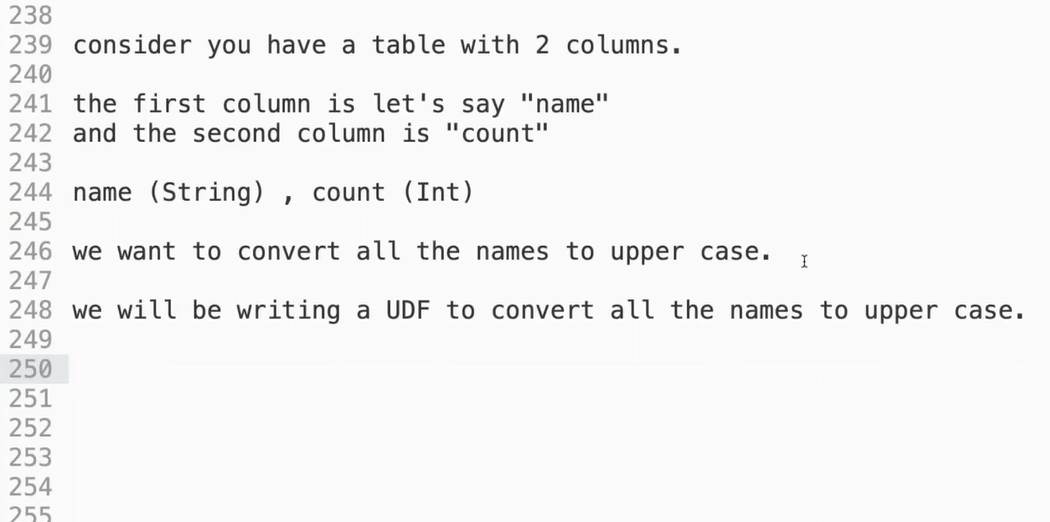


>>Some Items



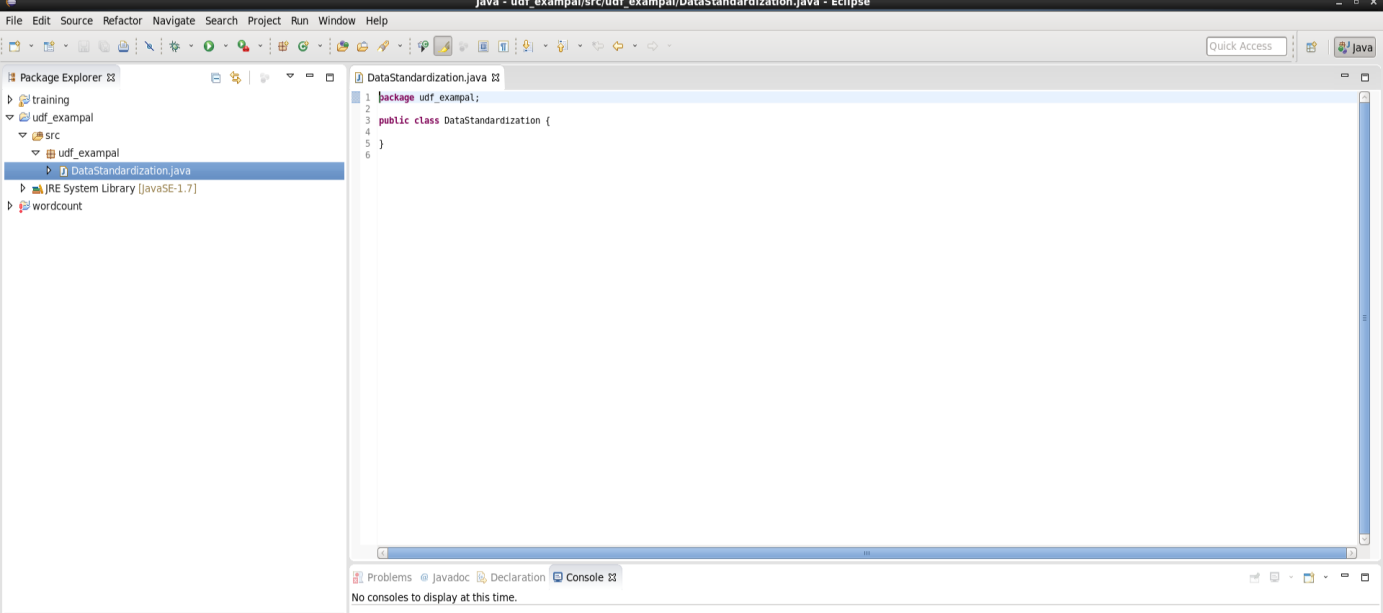
**UDF (User define Function)**

>>When user define function are not fit for our needs.



>>We need to write the Java Code for the convert the UDF function in Upper Case.

>>First of all open the Eclipse



>>Write code in the Eclipse

Package udf\_example;

import org.apache.hadoop.hive.ql.exec.UDF;

public class DataStandardization extends UDF

{

public String evaluate (String input)

{

if (input==null)

{

return null;

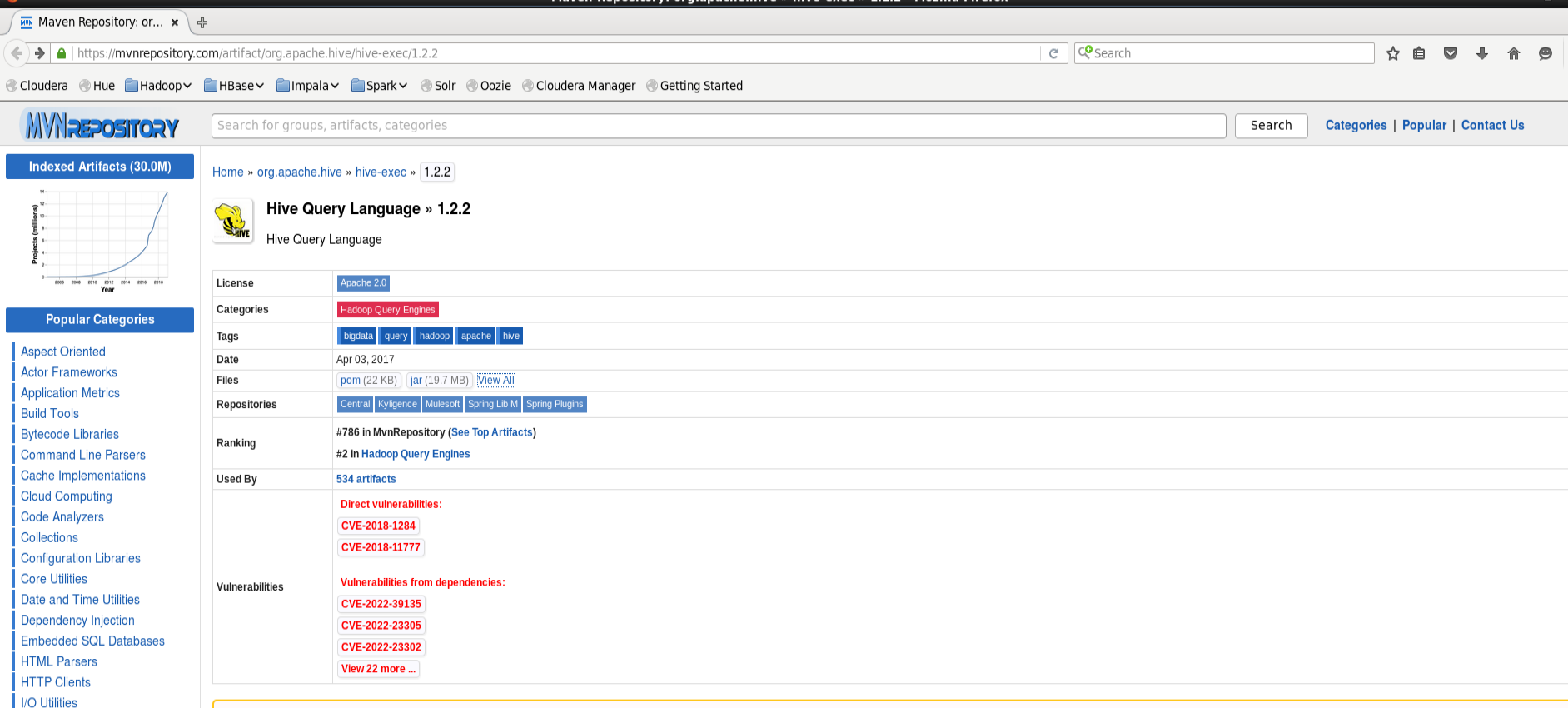
}

return (input.toString ().toUpperCase());

}

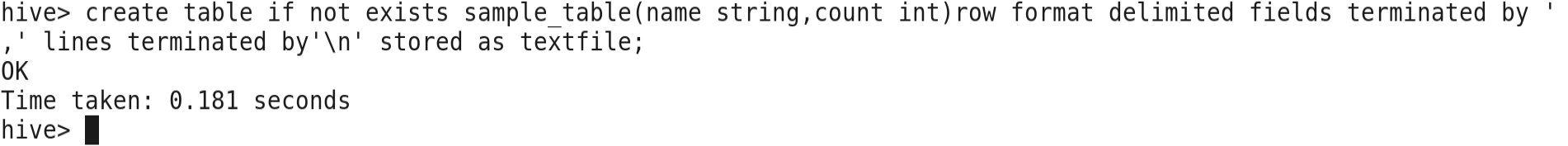
}

>>Now Download the jar file in the Eclipse for Resolve the error in the code.

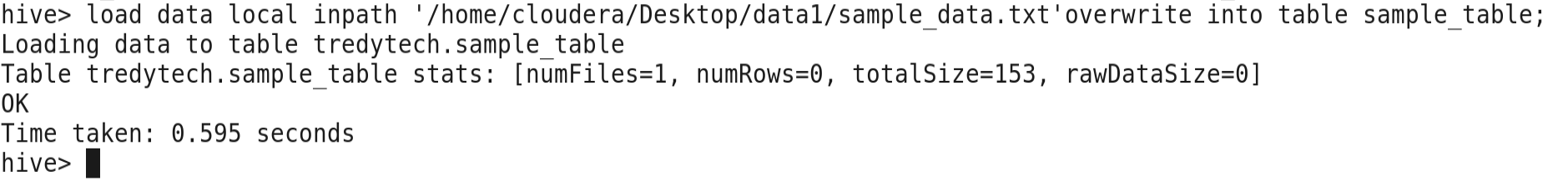


>>Now Download the jar file and Add into the Eclipse file.

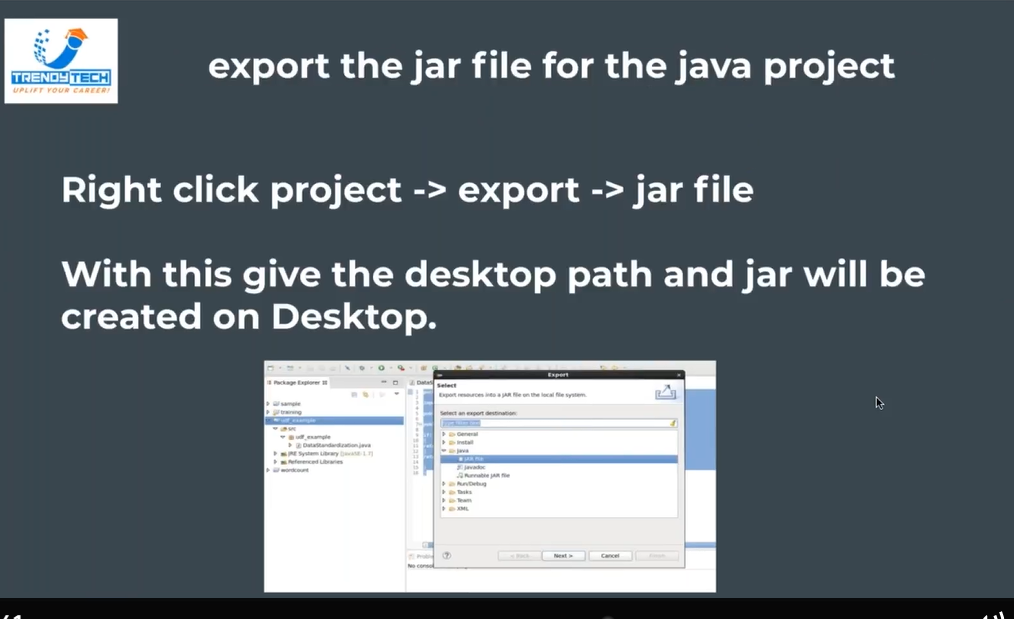
>>Now create a table of the UDF



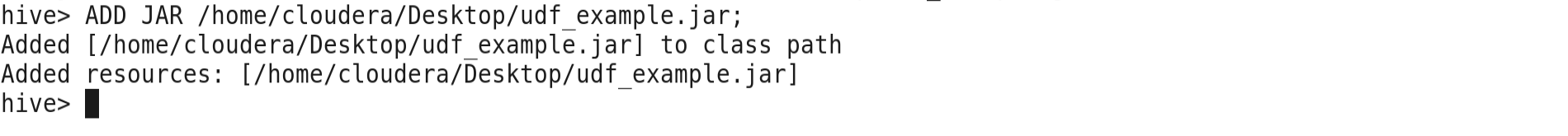
>>Load the data of the Sample.txt into Sample\_table.



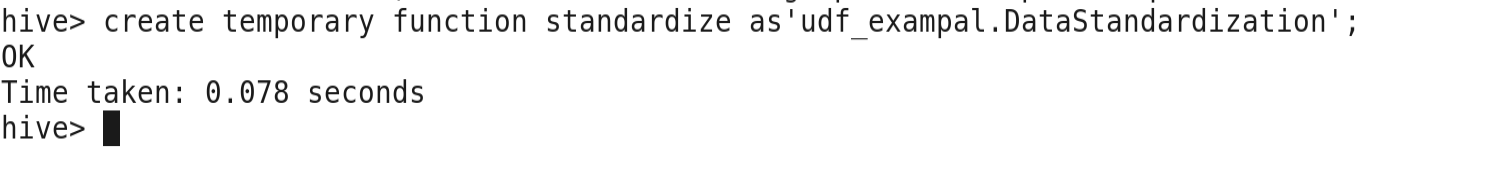
>>Now create a jar file



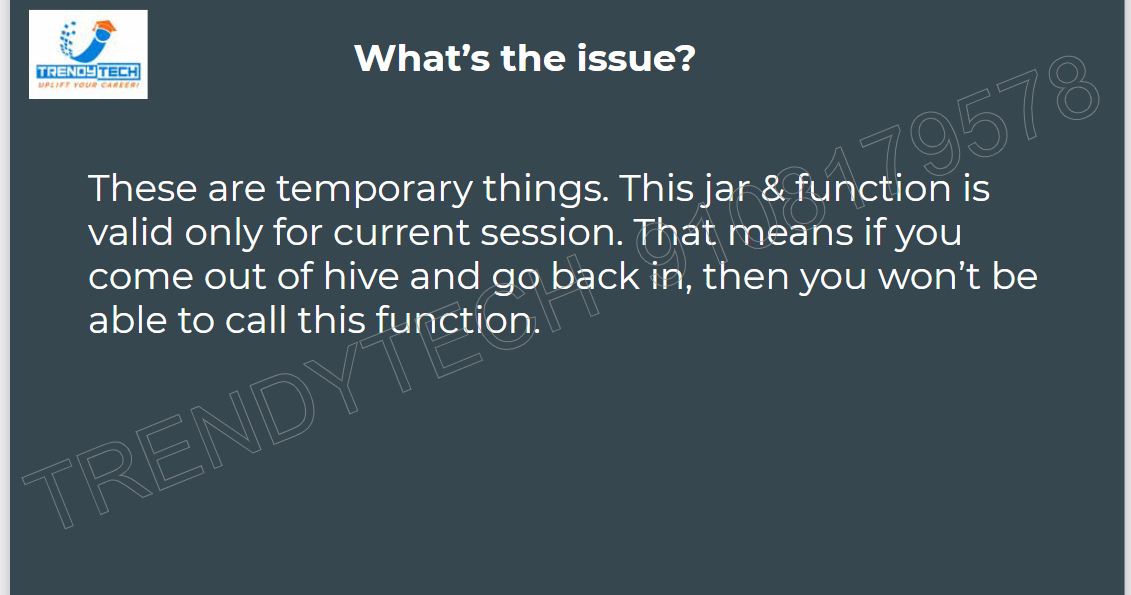
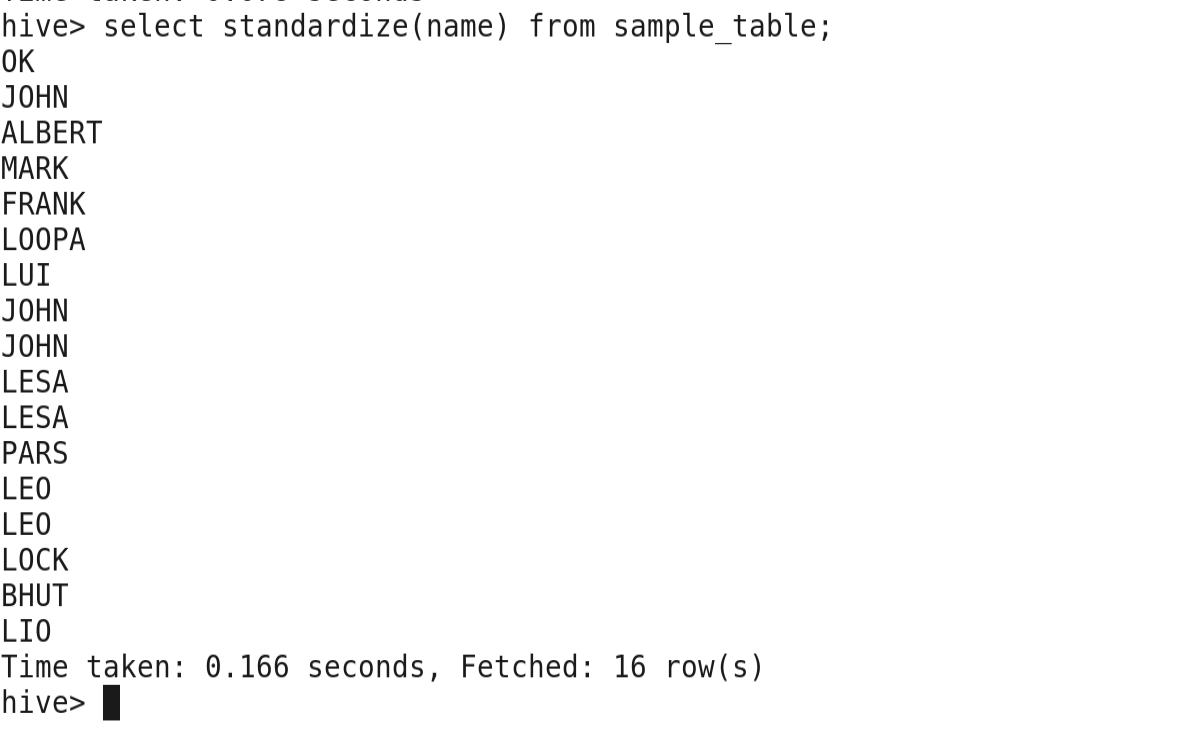
>>Now add the Jar file into the Hive from Desktop of the Cloudera.



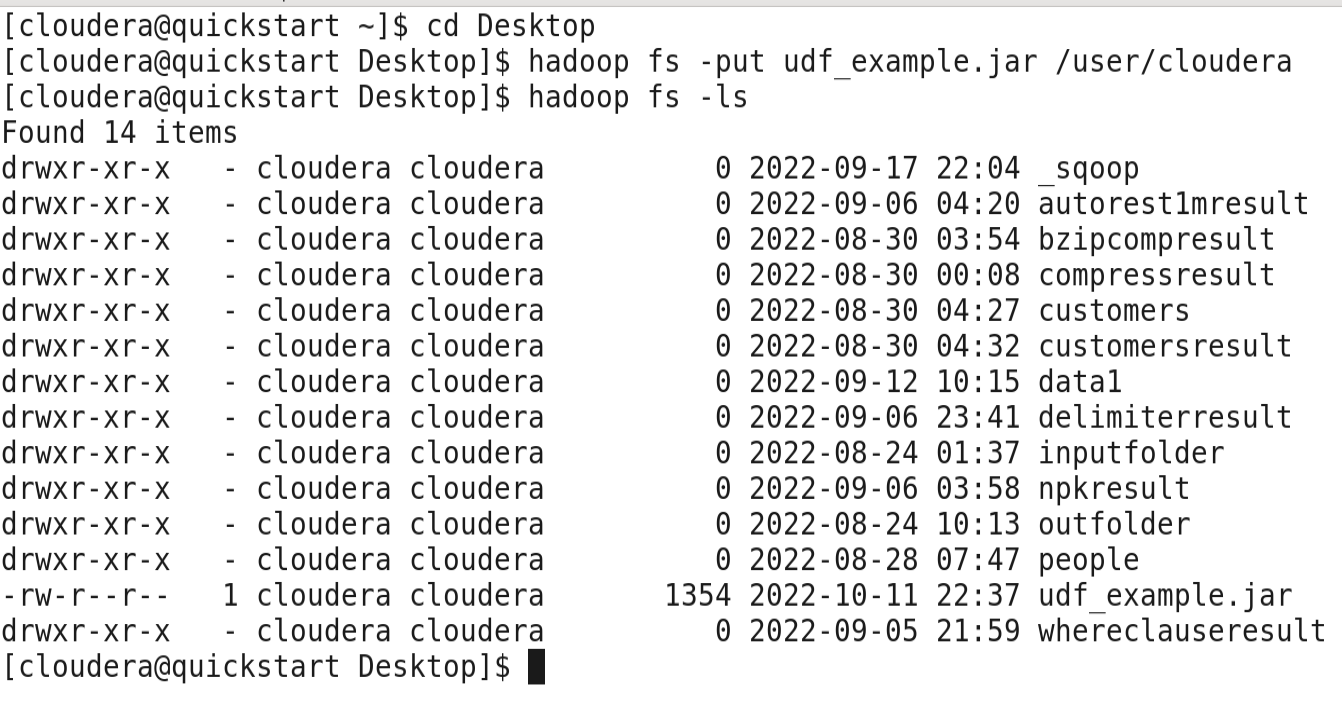
>>Now create a temporary Function



>>Now we are all set, try calling this UDF on a column in a table

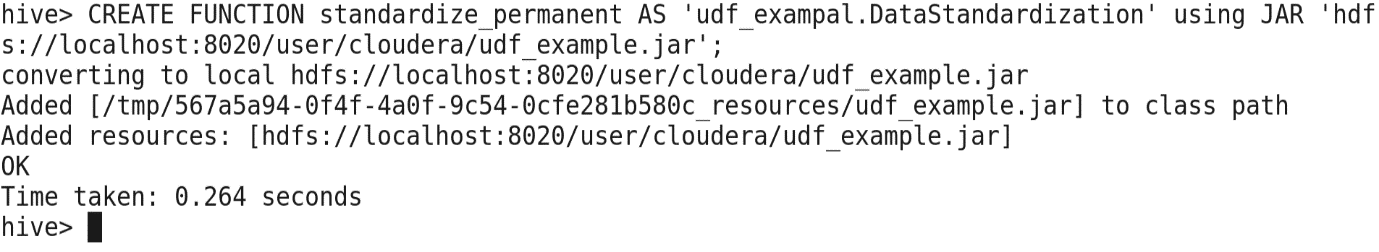


>> To make sure things are permanent, we need to first of all move the jar to HDFS

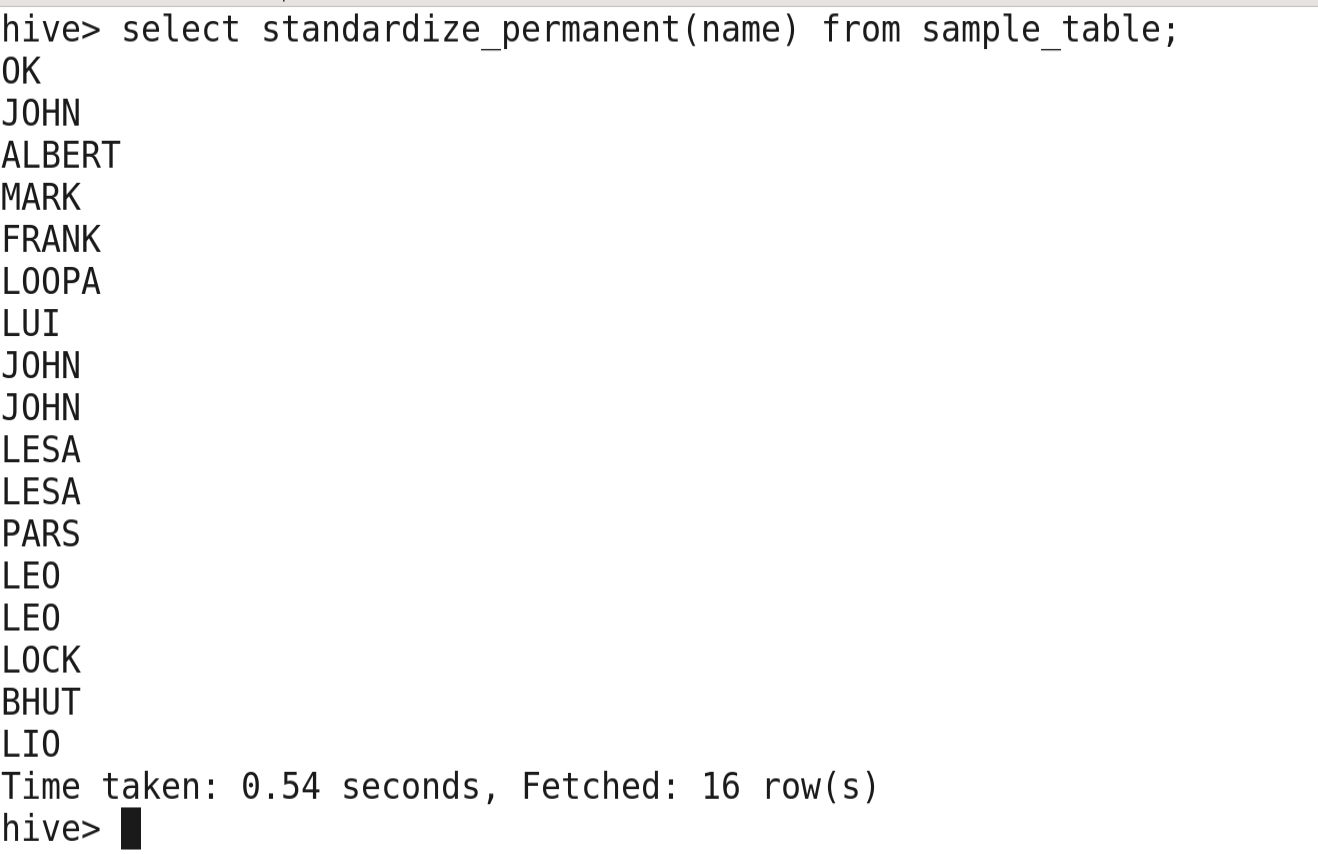


>> Then we need to define a permanent function not a Temporary one.

>>Now we have to create a Permanent function in Hive.



>>Now trying calling the Permanent Function table



>>When we exits from the Hive then we use the Same Permanent Function