**Explain the following in brief**

## ● Hive UDF (User Defined Function)

Generally Hive having some Built-in functions,we can use that Built-in functions for our Hive program with out adding any extra code but some times user requirement is not available in that built-in functions at that time user can write some own custom user defined functions called UDF (user defined function).Here is the simple steps of How To Write Hive UDF Example In Java.

• To write a UDF, start by extending the UDF class and implements and the evaluate() function.

• During query processing, an instance of the class is instantiated for each usage of the function in a query.

• The evaluate() is called for each input row.

• The result of evaluate() is returned to Hive.

• It is legal to overload the evaluate method.

• Hive will pick the method that matches in a similar way to Java method overloading.

• Finally to use UDF, create jar and register the class as temporary function.

ADD JAR<jar\_file\_path> ;

--ADD JAR /home/acadgild/hive/hive-udf.jar;

CREATE TEMPORARY FUNCTION<symbolic\_name> 'as<full\_class\_name>';

--CREATE TEMPORARY FUNCTION up AS 'udf.ToUpper';

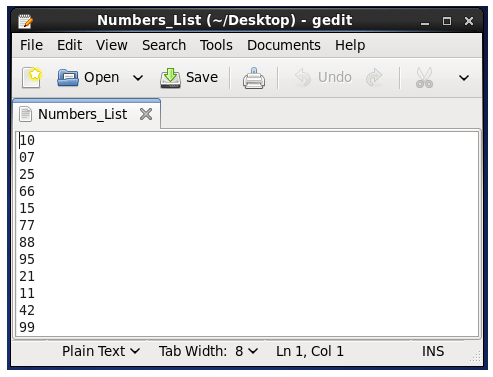
● **Hive UDAF** **(User-Defined Aggregation Functions)**

User-Defined Aggregation Functions (UDAFs) are an exceptional way to integrate advanced data-processing into Hive. Aggregate functions perform a calculation on a set of values and return a single value.

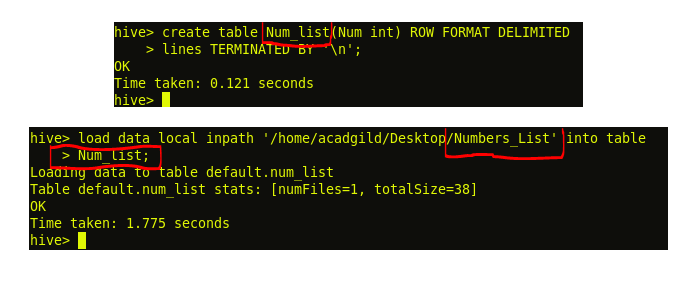
An aggregate function is more difficult to write than a regular UDF. Values are aggregated in chunks (potentially across many tasks), so the implementation has to be capable of combining partial aggregations into a final result.

**Steps for UDAF Execution.**

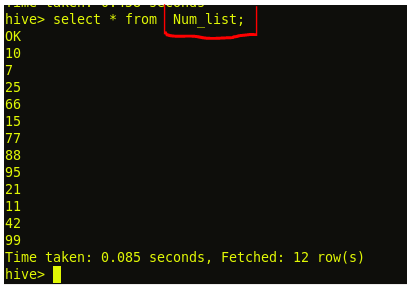
1. Creating a new Input Dataset



1. Create a new table  and load the input dataset



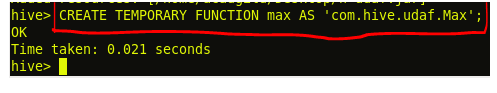
1. Display the contents of table Num\_list to ensure whether the input file have been loaded successfully or not.



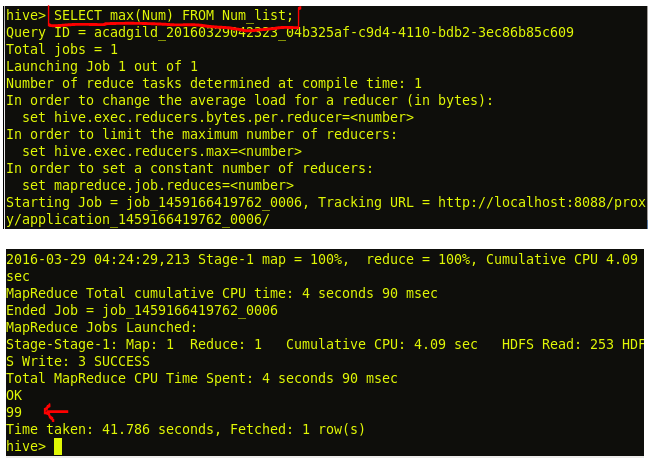
1. Add the Jar file in hive with complete path

C:\Users\613044\Documents\Capture.PNG

1. Create temporary function



1. Use the select statement to find the largest number from the table Num\_List



**● Hive UDTF (User Defined Table Generating Function)**

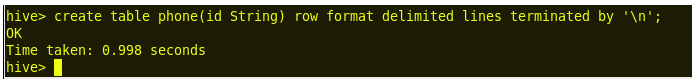
Hive UDTF contains multiple values in a single column based on the primary / unique id. UDTF is a User Defined Table Generating Function that operates on a single row and produces multiple rows a table as output.

**Steps for Executing Hive UDTF:**

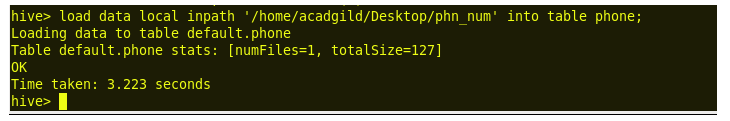
**Step 1:** After writing the code in Eclipse, add the below mentioned jar files in the program and then export it in the Hadoop environment as a jar file.



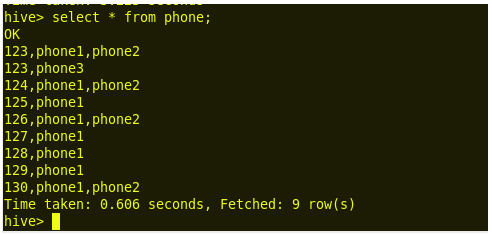
**Step 2:** Create a table  named ‘phone’ with a single column named ‘id’.



**Step 3:** Load the input data set phn\_num contents into the table phone.



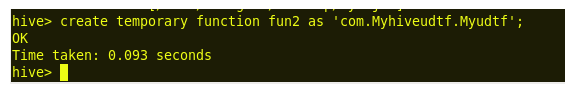
**Step 4:** Check if the data contents are loaded or not, using select statement.



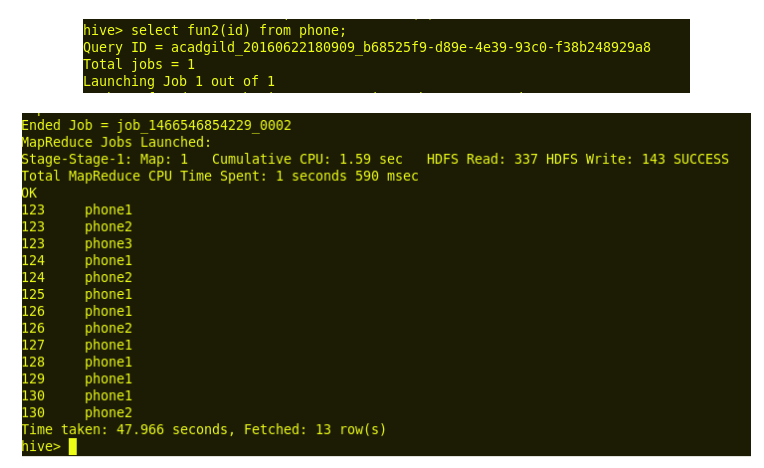
**Step 5:** Add the jar file with the complete path of the jar made as shown above.



**Step 6:** Create a temporary function as shown below.



**Step 7:** Use the select statement to populate the above table of strings with its primary id

****

* **Thrift server**

1. HiveServer2 (HS2) is a service that enables clients to execute queries against Hive. HiveServer2 is the successor to HiveServer1 which has been deprecated. HS2 supports multi-client concurrency and authentication. It is designed to provide better support for open API clients like JDBC and ODBC. You can find more details about hiveserver at <https://cwiki.apache.org/confluence/display/Hive/HiveServer2+Overview>
2. Hive Service is nothing but daemon which runs on your client node which sends requests to Hive Server.
3. Thrift is an RPC framework for building cross-platform services. Its stack consists of 4 layers: Server, Transport, Protocol, and Processor. You can find more details about the layers at<https://thrift.apache.org/docs/concepts>.

Relation between all these:

* The Thrift-based Hive service is the core of HS2 and responsible for servicing the Hive queries (e.g., from Beeline). In simple terms Hive server is based on thrift protocols which sends queries from hive client i.e., your command line interface or from HUE interface to the underlying data which can be in your HDFS or any other data sources.

Usage:

* When you query any hive tables or database, in background automatically your requests is transferred between hive service and hive server
* when you want to create your own service or project you can use thrift protocols which will help you in creating layers, think this as you are creating your user defined functions using libraries, so in that case libraries will be thrift