**Assignment 29.3**

**Explain Brief of the following in brief**

**● Hive UDF**

**● Hive UDAF**

**● Hive UDTF**

**● Thrift server**

**HIVE UDF:**

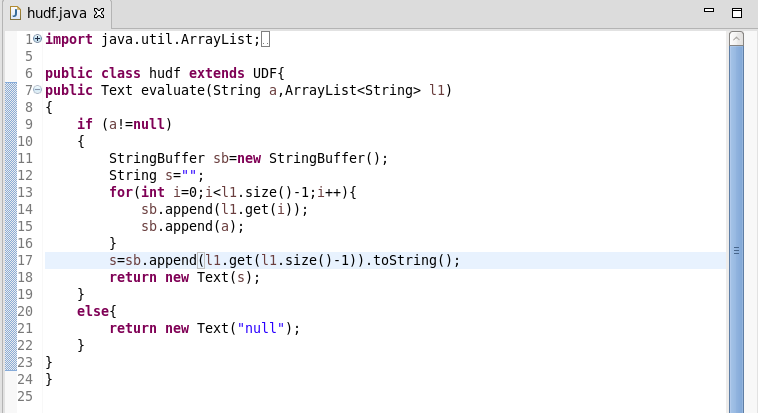
Hive has a rich set of functions that can be used to perform the analysis. But, sometimes there may come scenarios where our requirements cannot be met by simply using the built-in functions. In such a scenario, users need to implement custom User Defined Functions (UDF) and feed it into the Hive query.

User Defined Functions (UDFs) provides us a way to:

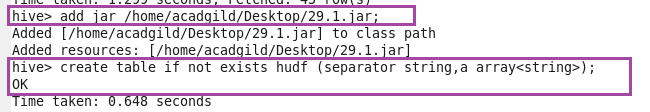
* Extend the functionality of Hive by writing functions that can be evaluated in Hive QL.
* Custom serializers and/or deserializer (“serdes”), which provide a way of either deserializing a custom file format stored on [HDFS](https://acadgild.com/blog/beginners-guide-for-hdfs/) to a POJO (plain old Java object), or serializing a POJO to a custom file format (or both).
* Custom mappers/reducers, which allow you to add a custom map or reduce steps into your Hive query. These map/reduce steps can be written in any programming language, and not just in Java.

Since the Hadoop framework is written in Java, naturally most of the Hadoop developers prefer Java to write the UDFs.

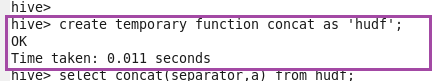
**Hive UDF to return all elements of the array separated by separator.**

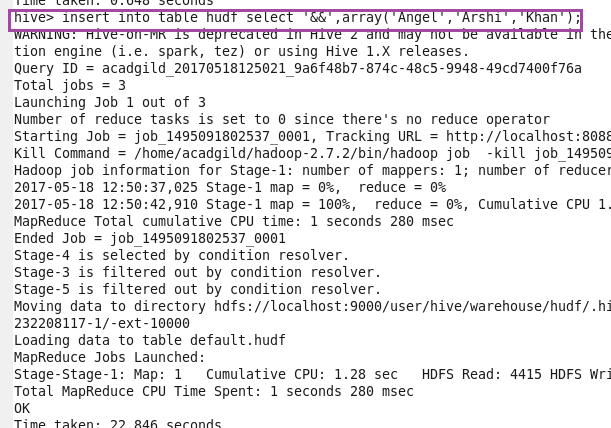
****

**Here, adding the jar file and creating the table named as ‘hudf’**

****

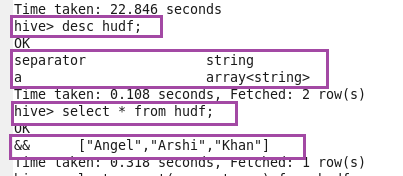
**Creating temporary function ‘concat’ then inserting the data into the created table ‘hudf’ using insert function.**

****

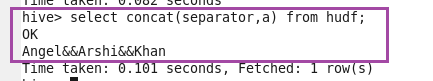
****

**Describing the table.**

**Performing basic select operation.**

****

**Retrieving data using temporary function.**

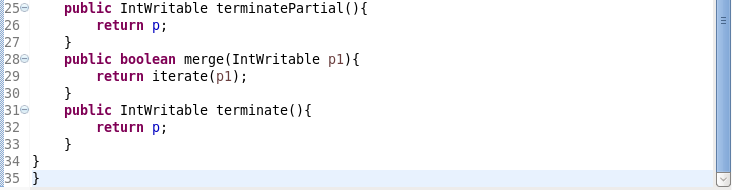
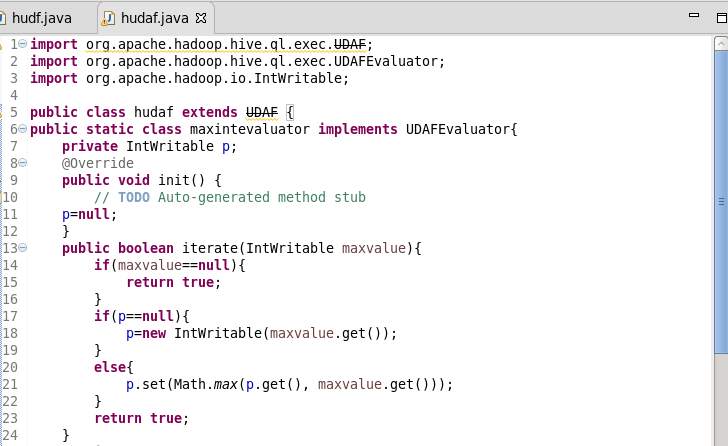
****

**UDAF :**

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.

**Here, I am showing hive UDAF to find the largest integer in a column.**

****

The init() method is called exactly once in an applet's life, when the applet is first loaded. It is used to set up the user interface. Here it initializes the evaluator function.

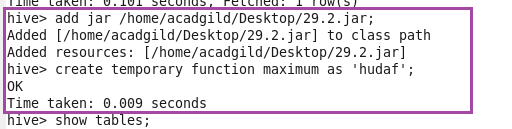
Then, a loop will run i.e, iterate() method for each value.

First, it will store the value and then using Math.max function it will compare all the values with each other and then get the maximum value out of it.

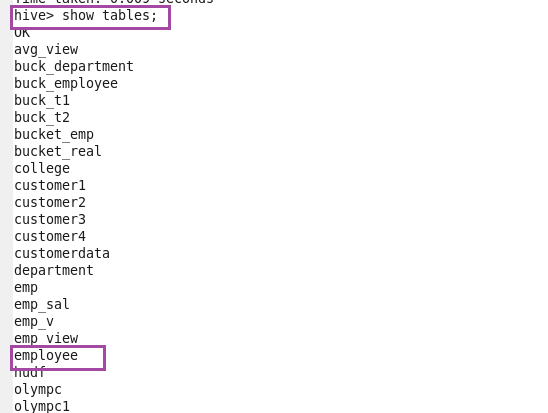
TerminalPartial() method is used to terminate the process terminally. Then using merge method to combine one partial aggregation with another.

Finally terminate method takes place and will return the largest integer stored in the private variable p.

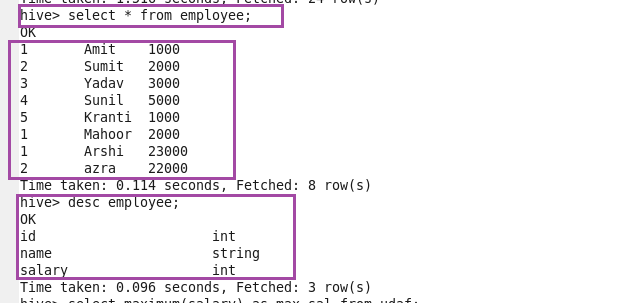
**Adding jar to hive :**

****

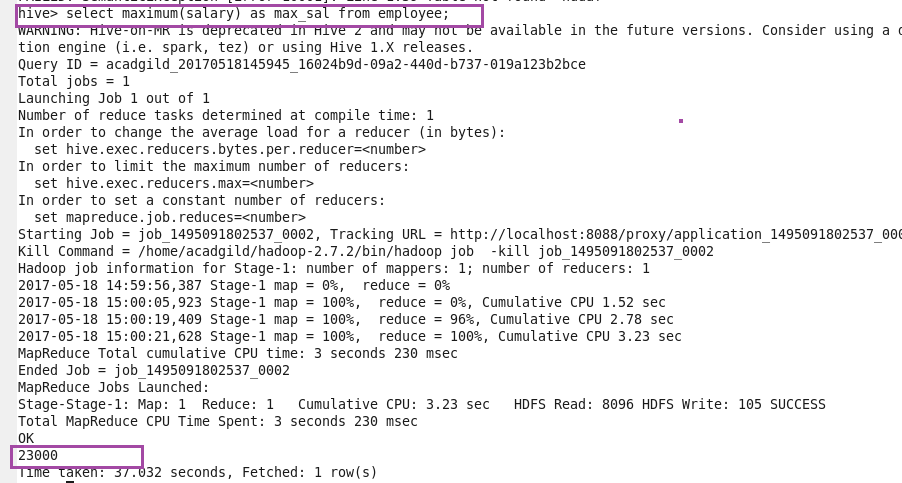
**Showing tables and I will use employee table**

****

**Displaying and describing all the contents of table employee**

****

**Using temporary function displaying the largest integer in the salary column.**

****

**UDTF:**

UDTF is a User Defined Table Generating Function that operates on a single row and produces multiple rows a table as output.

**Steps to Create & Execute UDTFs in Java:**

**Step1:**We have to extend a base Class Generic UDTF to write our business logic in Java.

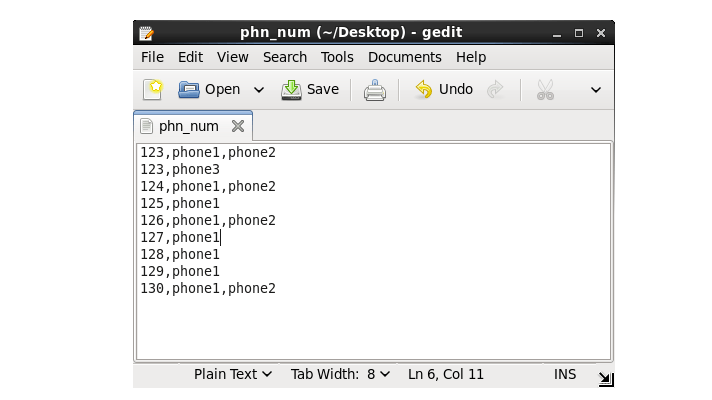
**Step2:**We need to override 3 methods namely **initialize(), process() and close()** in our class.

**Step3**: We need to export the JAR files to HDFS where hive is running.

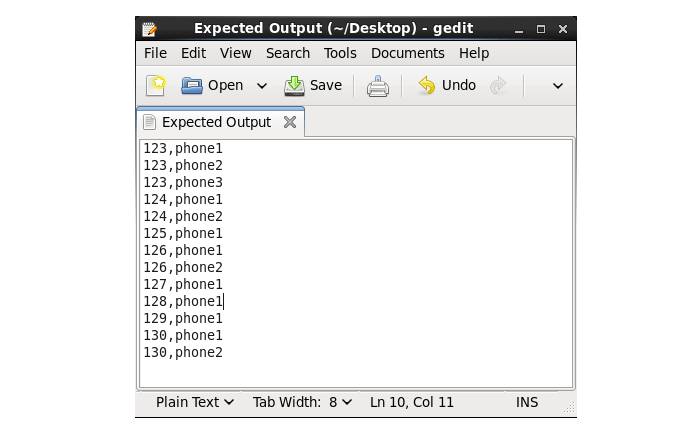
**Step4:**Add the exported JAR file to hive classpath.

**Step5:** Create a temporary function.

For ex : This is the dataset before using UDTF.

****

This is the output after using UDTF

****

Here, we can see that we have populated a single column, which contains multiple values to its primary id.

**● Thrift server**

[Apache Hive](https://acadgild.com/big-data/apache-spark-training-certification) is a data warehouse software that facilitates reading, writing and managing large data sets residing in distributed storage using SQL.

Let’s consider a scenario, where the user is looking forward to performing an operation on Hive server, and the [Hadoop cluster](https://acadgild.com/big-data/big-data-development-training-certification) or Hive software setup is not installed in his/her system. The solution for the above scenario is that the user can write codes in other languages and access Hive server using Apache Thrift interface.

Apache Thrift is a software framework for scalable cross-language services development, which combines a software stack with a code generation engine to build services that work efficiently and seamlessly between C++, Java, Python, PHP, Ruby, Perl, C#, JavaScript, Node.js and other languages.

Thrift can be used when developing a web service that uses a service developed in one language access that is in another language.

HiveServer is a service that allows a remote client to submit requests to Hive, using a variety of programming languages, and retrieve results. It is built on Apache Thrift, therefore it is sometimes called as the Thrift server.

In the context of Hive, Java language can be used to access Hive server. The Thrift interface acts as a bridge, allowing other languages to access Hive, using a Thrift server that interacts with the Java client.