

**Solution-**

User Defined Functions (UDFs) in hive are used to write our own logic in terms of code into hive when we are not able to get the desired result from hive's built in functions. We can invoke the UDFs from hive query.

* 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 ; --ADD JAR /home/acadgild/hive/hive-udf.jar; CREATE TEMPORARY FUNCTION as ''; --CREATE TEMPORARY FUNCTION up AS 'udf.ToUpper';

There are three kind of UDFs in Hive:

**1.Regular UDF**

**2. User Defined Aggregate Function (UDAF)**

**3.User Defined Tabular Function (UDTF).**

**1. Regular UDF:** UDFs works on a single row in a table and produces a single row as output. Its one to one relationship between input and output of a function.

e.g Hive built in TRIM() function.

Hive allows us to define our own UDFs as well which can be done by following steps-

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

**Step2**: In order to write business logic , we have to overload a method called evaluate**()**inside 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 using below command from hive terminal:

ADD JAR EXPORTED\_FILE\_NAME.jar

**Step5:**In order to apply business logic on top of hive column using our UDF, we need to create a temporary function for the exported jar file.

CREATE temporary function func\_name as 'absolute\_class\_path\_name'

**2. UDAF:** User defined aggregate functions works on more than one row and gives single row as output.

**e.g Hive built in MAX() or COUNT() functions.**

here the relation is many to one. Lets say you have a table with students name, id and total marks, so here if I have 10 rows in the table and if I have to find student who got maximum number then our query need to check each 10 row to find the maximum but ultimately we get only one output which is the maximum.

* 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.

Let’s see now the steps for UDAF Execution.

1. Creating a new Input Dataset- First the input dataset has to be created.
2. Create a new table  and load the input dataset- A new table has to be created and load dataset into the table.
3. Display the contents of table to ensure whether the input file have been loaded successfully or not.

By using select statement command we can see if the contents of the dataset  have been loaded to the table or not.

1. Add the Jar file in hive with complete path (Jar file made from source code need to be added)
2. Create temporary function as shown below
3. Use the select statement to find the largest number from the table

After, successfully following the above steps we can see use the Select statement command to find the particular data from the table.

**3. UDTF:** User defined tabular function works on one row as input and returns multiple rows as output. So here the relation in one to many.

**e.g Hive built in EXPLODE() function.**

Now lets take an array column USER\_IDS as ARRAY10,12,5,45> then SELECT EXPLODE(USER\_IDS) as ID FROM T\_USER. will give 10,12,5,45 as four different rows in output. UDTF can be used to split a column into multiple column as well which we will look in below example. Here alias "AS" clause is mandatory

Steps for Executing Hive UDTF:

**Step 1:** After writing the code in Eclipse, add the needed 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.

Differences Between UDF, UDAF and UDTF:

UDF:

UDF is a user-defined function that takes a single input value and produces a single output value. When used in a query, we can call it once for each row in the result set.

Example:

input.toString().toUpperCase();

input.toString().toLowerCase();

The above methods will convert a string of lowercase to uppercase and vice versa.

UDAF:

UDAF is a user-defined aggregate function (UDAF) that accepts a group of values and returns a single value. Users can implement UDAFs to summarize and condense sets of rows in the same style as the built-in COUNT, MAX(), SUM(), and AVG() functions.

Example:

You can refer below post to know more about Hive UDAF.

Execution of Hive UDAF

UDTF:

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