**Problem Statement:-**

Explain Brief of the following in brief

● Hive UDF

● Hive UDAF

● Hive UDTF

● Thrift server

# HIVE UDF

User Defined Functions (UDFs) in hive are used to plug in 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.

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.**

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

**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.

● HIVE 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.

Hive allows us to define our own UDAFs.

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

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

**Step2:** We need to overwrite five methods called **init(), iterate(), terminatePartial(), merge() and terminate()**in our class.

**Step3 to Step5:** It is same as explained in UDF.

● HIVE 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.**

**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 to Step5:** It is same as explained in UDF.

● 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.