**ASS-29.3**

**Explain Brief of the following in brief**

**● Hive UDF**

UDF works on a single row in a table and produces a single row as output. It has 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: Extend a base Class UDF to write 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'.

**● 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. Let’s 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.

**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: 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'.

**● 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. 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 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 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'.

**● Thrift server** 1. Thrift is an interface definition language and binary communication protocol that is used to define and create services for numerous languages.

2. It is used as a remote procedure call (RPC) framework and was developed at Facebook for "scalable cross-language services development".

3. It combines a software stack with a code generation engine to build cross-platform services that can connect applications written in a variety of languages and frameworks, including ActionScript, C, C++,C#, Cappuccino, Cocoa, Delphi, Erlang, Go, Haskell, Java, Node.js, Objective-C, OCaml, Perl, PHP, Python, Ruby and Smalltalk.

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

5. 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.

6. 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.

7. The current implementation, based on Thrift RPC, is an improved version of HiveServer and supports multi-client concurrency and authentication.