**Hive UDF**

**Task 1: Code your function**

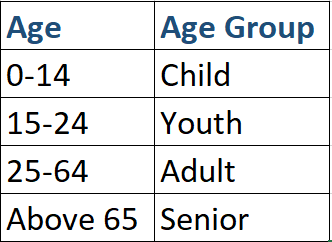
1. Simple UDF

**Purpose:**

One primitive input and one output.

**Example:**

Convert an age to age group information



**Coding Approach:**

Create a class which extends org.apache.hadoop.hive.ql.exec.UDF

Implement evaluate method

**Task 2: Deploy it temporarily**

1. Add JAR to classpath

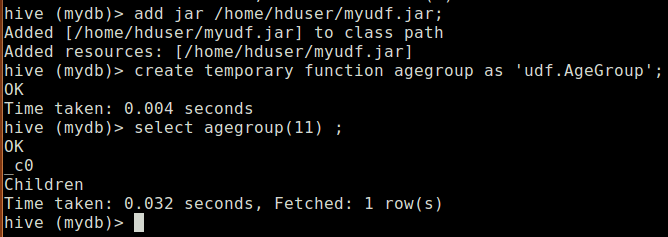
add JAR <Local path of your JAR file>

1. Create temporary function

create temporary function agegroup as ‘udf.AgeGroup”

1. Use it

select agegroup(14)



**Or just append it into .hiverc file**

Following is the content of a sample .hiverc file. It should be within conf directory of hive. If not present then you can create one.

*set hive.cli.print.header=true;*

*set hive.cli.print.current.db=true;*

*add JAR /home/s\_kante/IdeaProjects/HiveUDF/out/artifacts/HiveUDF\_jar/HiveUDF.jar;*

*create temporary function isaccepted as 'udf.IsAcceptedNew';*

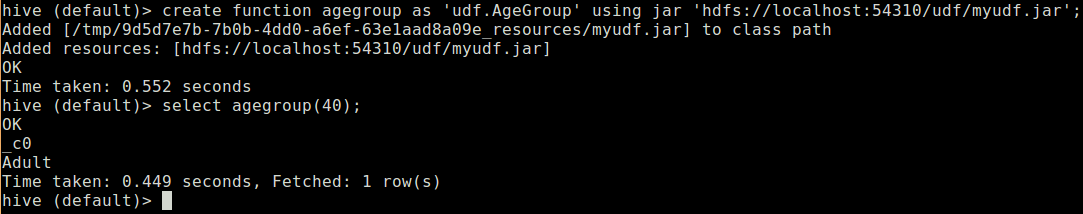
**Task 3: Deploy it permanently**

1. Copy JAR file to hdfs file system

hadoop fs -copyFromLocal myudf.jar /udf/

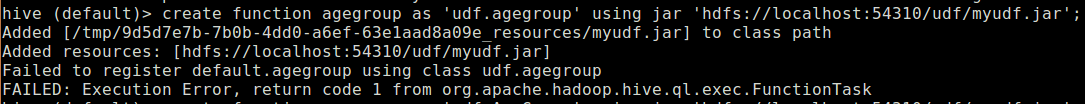
1. Register function

create function agegroup as ‘udf.AgeGroup’ using jar ‘hdfs://localhost:54310/udf/myudf.jar’;

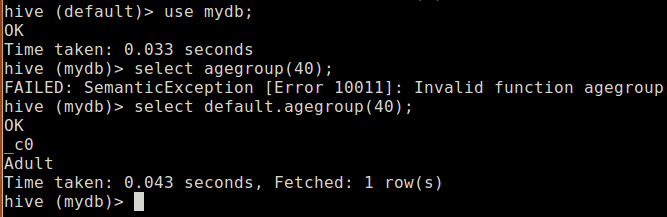


**Possible errors:**

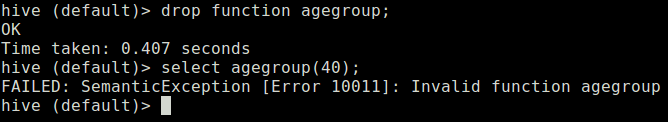
* Case sensitive package and class name



* Using function in another database



**Task 4: Deregister the function**



**Hive UDF for Complex data:**

1. **Array**

*create table result(student\_id int, bands array<double>) row format delimited fields terminated by ‘|’ collection items terminated by ‘,’*

*insert into result select 10 , array(cast(4.5 as double),cast(6.7 as double));*

hive (mydb)> *select \* from result;*

OK

result.student\_id result.bands

10 [4.5,6.7]

Time taken: 0.662 seconds, Fetched: 1 row(s)

hive (mydb)> *select student\_id, bands[0] from result;*

OK

student\_id \_c1

10 4.5

Time taken: 0.821 seconds, Fetched: 1 row(s)

**Hdfs view**

hduser@shyam:/usr/local/hadoop/etc/hadoop$ *hdfs dfs -cat /user/hive/warehouse/mydb.db/result/000000\_0*

10|4.5,6.7

Let’s deploy a UDF IsAccepted that returns True if the student has scored above certain bands in each category and average is also greater than some minimum value.

**Debugging Hive CLI**

hive -hiveconf hive.log.file=debug\_hive\_20180403.log -hiveconf hive.log.dir=/tmp/hivedebug/ -hiveconf hive.root.logger=DEBUG,DRFA

**User Defined Aggregate Function (UDAF)**

**Purpose:**

UDAF takes multiple records with primitive data types as input to generate single record with primitive data type as output.

**Example:**

For a given IELTS bands in denormalized form, decide if student has cleared the exam or not

**Coding Approach:**

Create a class which extends *org.apache.hadoop.hive.ql.exec.UDAF*

Create a subclass within that class which implements *org.apache.hadoop.hive.ql.exec.UDAFEvaluator*

Define methods

**init**: initialize variables

**iterate**: Will be called for each record

**terminatePartial**: how to behave when process completes with partial result on one node

**merge**: to merge two partial results

**terminate**: finally output the result

hive (mydb)> select \* from result2;

OK

result2.id result2.band

1 4.5

1 6.5

1 7.5

2 6.5

2 7.5

Time taken: 0.108 seconds, Fetched: 5 row(s)

hive (mydb)> select id, haspassed(band, cast(6 as double), cast(5 as double)) as result from result2 group by id;

OK

id result

1 NO

2 YES

Time taken: 1.496 seconds, Fetched: 2 row(s)

**User Defined Tabular Function (UDTF)**

**Purpose:**

UDTF takes single record as input and generates multiple records in output.

**Example:**

Generate combination of transaction id and product id for a given transaction with all products flattened in single record.

**Coding Approach:**

Create a class which extends org.apache.hadoop.hive.ql.udf.generic.GenericUDTF

Define methods

**initialize**: will return the structure information of output record

**process**: will be called on each new record

**close**: any cleanup tasks to be carried out

hive (mydb)> select flattrans("1|2,3,4");

OK

trans\_id product\_id

1 2

1 3

1 4

**Reference:**

<https://blog.matthewrathbone.com/2013/08/10/guide-to-writing-hive-udfs.html>

<https://cwiki.apache.org/confluence/display/Hive/GenericUDAFCaseStudy>

<https://cwiki.apache.org/confluence/display/Hive/GenericUDAFCaseStudy#GenericUDAFCaseStudy-WritingGenericUDAFs:ATutorial>

<https://community.hortonworks.com/content/supportkb/150214/how-to-enable-debug-hive-cli-logging.html>

**Hive Serde:**

<https://cwiki.apache.org/confluence/display/Hive/LanguageManual+DDL#LanguageManualDDL-RowFormat,StorageFormat,andSerDe>

<https://cwiki.apache.org/confluence/display/Hive/DeveloperGuide#DeveloperGuide-CodeOrganizationandaBriefArchitecture>

<https://stackoverflow.com/questions/24607685/loading-xml-data-into-hive-table-org-apache-hadoop-hive-ql-metadata-hiveexcepti>

**Fun to Learn:**

<https://stackoverflow.com/questions/20208696/hadoop-restart-datanode-and-tasktracker>