**EX : 04(1) REG.NO:210701307 CREATE USER DEFINED FUNCTION(UDF)**

**Aim :**

To create User Define Function in Apache Pig and execute it on map reduce.

**Procedure:**

**Create a sample text file** hadoop@Ubuntu:~/Documents$ nano sample.txt

Paste the below content to sample.txt

1,John

2,Jane

3,Joe

4,Emmahadoop@Ubuntu:~/Documents$ hadoop fs -put sample.txt /home/hadoop/piginput/ **Create PIG File** hadoop@Ubuntu:~/Documents$ nano demo\_pig.pig **paste the below the content to demo\_pig.pig**

-- Load the data from HDFS data = LOAD '/home/hadoop/piginput/sample.txt' USING PigStorage(',') AS (id:int> -- Dump the data to check if it was loaded correctly

DUMP data;

**---------------------------------------------------------------------------------**

**Run the above file** hadoop@Ubuntu:~/Documents$ pig demo\_pig.pig

2024-08-07 12:13:08,791 [main] INFOorg.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1

(1,John)

(2,Jane)

(3,Joe)

(4,Emma)

---------------------------------------------------------------------------------------------------

**Create udf file an save as uppercase\_udf.py** uppercase\_udf.py

--------------------------------------------------------------------------------------------

def uppercase(text): return text.upper() if \_\_name\_\_ == "\_\_main\_\_":

import sys for line in sys.stdin: line = line.strip() result = uppercase(line) print(result)

-------------------------------------------------------------------------------------------

**Create the udfs folder on hadoop** hadoop@Ubuntu:~/Documents$ hadoop fs -mkdir /home/hadoop/udfs **put the upppercase\_udf.py in to the abv folder** hadoop@Ubuntu:~/Documents$ hdfs dfs -put uppercase\_udf.py /home/hadoop/udfs/

**---------------------------------------------------------------------------------**

hadoop@Ubuntu:~/Documents$ nano udf\_example.pig **copy and paste the below content on udf\_example.pig**  -- Register the Python UDF script

REGISTER 'hdfs:///home/hadoop/udfs/uppercase\_udf.py' USING jython AS udf;

-- Load some data data = LOAD 'hdfs:///home/hadoop/sample.txt' AS (text:chararray);

-- Use the Python UDF uppercased\_data = FOREACH data GENERATE udf.uppercase(text) AS uppercase\_text;

-- Store the result

STORE uppercased\_data INTO 'hdfs:///home/hadoop/pig\_output\_data';

--------------------------------------------------------------------------------------------------------------

**place sample.txt file on hadoop** hadoop@Ubuntu:~/Documents$ hadoop fs -put sample.txt /home/hadoop/  **To Run the pig file** hadoop@Ubuntu:~/Documents$ pig -f udf\_example.pig

**finally u get**

Success!

**Job Stats (time in seconds):**

JobId Maps Reduces MaxMapTimeMinMapTime AvgMapTime MedianMapTime

MaxReduceTime MinReduceTime AvgReduceTime MedianReducetime Alias Feature Outputs job\_local1786848041\_0001 1 0 n/a n/a n/a n/a 00 0 0 data,uppercased\_data MAP\_ONLY hdfs:///home/hadoop/pig\_output\_data,

Input(s):

Successfully read 4 records (42778068 bytes) from: "hdfs:///home/hadoop/sample.txt"

**Output(s):**

Successfully stored 4 records (42777870 bytes) in: "hdfs:///home/hadoop/pig\_output\_data"  Counters:

Total records written : 4

Total bytes written : 42777870

Spillable Memory Manager spill count : 0

Total bags proactively spilled: 0 Total records proactively spilled: 0

Job DAG:

job\_local1786848041\_0001 2024-08-07 13:33:04,631 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl -

JobTracker metrics system already initialized! 2024-08-07 13:33:04,639 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl -

JobTracker metrics system already initialized! 2024-08-07 13:33:04,644 [main] WARN org.apache.hadoop.metrics2.impl.MetricsSystemImpl -

JobTracker metrics system already initialized! 2024-08-07 13:33:04,667 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher -

Success!

**Note :**

**If any error check jython package is installed and check the path specified on the above steps are give correctly**

**--------------------------------------------------------------------------------------------------------------**

**To check the output file is created**

hadoop@Ubuntu:~/Documents$ hdfs dfs -ls /home/hadoop/pig\_output\_data

Found 2 items

If you need to examine the files in the output folder, use: **To view the output hadoop@Ubuntu:~/Documents$ hdfs dfs -cat /home/hadoop/pig\_output\_data/part-m**

**00000**

1,JOHN

2,JANE

3,JOE

4,EMMA



**Result:**

Thus the program is executed successfully