**PIG UseCases**

**Practicing every usecase in Production Mode only**

**Input File for Few UseCases:**

1201,gopal,45,Male,50000

1202,manisha,40,Female,51000

1203,khaleel,34,Male,30000

1204,prasanth,30,Male,31000

1205,kiran,20,Male,40000

1206,laxmi,25,Female,35000

1207,bhavya,20,Female,15000

1208,reshma,19,Female,14000

1209,kranthi,22,Male,22000

1210,Satish,24,Male,25000

1211,Krishna,25,Male,26000

1212,Arshad,28,Male,20000

1213,lavanya,18,Female,8000

1. UseCase1: ScriptFile

employees = LOAD '/pigBatch89/prac1/input.txt' using PigStorage(',');

empNames = FOREACH employees GENERATE $1;

dump empNames;

2. UseCase2:

employees = LOAD '/pigBatch89/input.txt' using PigStorage(',');

empNames = FOREACH employees GENERATE $1,$3,$4;

STORE empNames INTO '/pigBatch89/prac2/output';

3. Usecase3: filterUsecase.pig

empData = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

filData = FILTER empData BY gender == 'Male';

orderData = ORDER filData BY age;

STORE filData INTO '/pigBatch89/prac4/filData';

STORE orderData INTO '/pigBatch89/prac4/orderData';

4. UseCase4: aliasNameUseCase.pig

empData = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

aliasData = FOREACH empData GENERATE eName, gender, salary+10000 AS salary;

filterData = FILTER aliasData BY salary > 25000;

STORE aliasData into '/pigBatch89/prac5/aliasData';

STORE filterData into '/pigBatch89/prac5/filterData';

5. UseCase5: splitUseCase.pig

empData = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

SPLIT empData INTO maleData IF gender == 'Male', femaleData IF gender == 'Female';

STORE maleData INTO '/pigBatch89/prac5/males';STORE femaleData INTO '/pigBatch89/prac5/females';

UseCase6: groupByUseCase.pig

empData = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

employees = FOREACH empData GENERATE eName, age, gender, eloc;

maleGroups = GROUP employees BY eloc;

--maleGroupsLoc = GROUP employees BY (eloc,eName);

STORE employees INTO '/pigBatch89/prac6/Selectedemployees';STORE maleGroups INTO '/pigBatch89/prac6/malesByLoc';

UseCase7: limitUseCase.pig

empData = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

orderData = ORDER empData BY salary desc;

topThree = LIMIT orderData 3;

groupEmp = GROUP topThree BY gender;

STORE groupEmp INTO '/pigBatch89/prac7/groupedEmp';

UseCase8: setsUseCase.pig

empCTS.txt

1201,gopal,45,Male,50000

1202,manisha,40,Female,51000

1203,khaleel,34,Male,30000

1204,prasanth,30,Male,31000

1205,kiran,20,Male,40000

1206,laxmi,25,Female,35000

1207,bhavya,20,Female,15000

1208,reshma,19,Female,14000

1209,kranthi,22,Male,22000

1210,Satish,24,Male,25000

1211,Krishna,25,Male,26000

1212,Arshad,28,Male,20000

1213,lavanya,18,Female,8000

empInfy.txt

1201,gopal,Male,hyd

2202,manisha,Female,hyd

2203,jagadeesh,Male,chn

1204,prasanth,Male,chn

1205,kiran,Male,hyd

1206,laxmi,Female,blr

1207,bhavya,Female,vlr

2208,rakesh,Female,chn

1209,kranthi,Male,pune

1210,Satish,Male,mum

2211,Krishna,Male,blr

1212,Arshad,Male,chn

2213,lavanya,Female,hyd

1214,ramu,Female,hyd

empCTS = LOAD '/pigBatch89/input/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray, eloc:chararray,salary:int);

empInfy = LOAD '/pigBatch89/input/empInfy.txt' using PigStorage(',') as (Id:int,eName:chararray,Gender:chararray, Loc:chararray);

crossData = CROSS empCTS, empInfy;

unionData = UNION empCTS, empInfy;

STORE crossData INTO '/pigBatch89/prac8/crossOutput';

STORE unionData INTO '/pigBatch89/prac8/unionOutput';

UseCase9: joinsUseCase.pig

empCTS = LOAD '/pigBatch89/input/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray, eloc:chararray,salary:int);

empInfy = LOAD '/pigBatch89/input/empInfy.txt' using PigStorage(',') as (Id:int,eName:chararray,Gender:chararray, Loc:chararray);

innerJoin = JOIN empCTS BY eid, empInfy BY Id;

leftJoin = JOIN empCTS BY gender LEFT, empInfy BY Gender;

rightJoin = JOIN empCTS BY eName RIGHT, empInfy BY eName;

fullJoin = JOIN empCTS BY eid FULL, empInfy BY Id;

STORE innerJoin INTO '/pigBatch89/prac9/innerOutput'; STORE leftJoin INTO '/pigBatch89/prac9/leftOutput'; STORE rightJoin INTO

'/pigBatch89/prac9/rightOutput'; STORE fullJoin INTO '/pigBatch89/prac9/fullOutput';

10. CustomDataUseCase.pig

1201,gopal,45,Male,50000

1202,manisha,40,Female,51000

1203,khaleel,34,Male,30000

1204,prasanth,30,Male,31000

1205,kiran,20,Male,40000

1206,laxmi,25,Female,35000

1207,bhavya,20,Female,15000

1208,reshma,19,Female,14000

1209,kranthi,22,Male,22000

1210,Satish,24,Male,25000

1211,Krishna,25,Male,26000

1212,Arshad,28,Male,20000

1213,lavanya,18,Female,8000

daily = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

toBagData = FOREACH daily GENERATE TOBAG(eName,age,gender,salary);

toTupleData = FOREACH daily GENERATE TOTUPLE(eName,age,gender,salary);

toMapData = FOREACH daily GENERATE TOMAP(eName,eloc);

toMultiMapData = FOREACH daily GENERATE TOMAP(eName,(gender,salary));

STORE toBagData INTO '/pigBatch89/prac11/bagOutput';STORE toTupleData INTO '/pigBatch89/prac11/tupleOutput';STORE toMapData INTO '/pigBatch89/prac11/toMapData';STORE toMultiMapData INTO '/pigBatch89/prac11/toMultiMapData';

11. parallelUseCase.pig

initData = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

parallerReducers = GROUP initData BY gender PARALLEL 6;

STORE parallerReducers INTO '/pigBatch89/prac12';

12. tokenizeUseCase.pig

spaceData = LOAD '/pigBatch89/input\_withspaces.txt' using PigStorage('\n') as (line:chararray);

commaData = LOAD '/pigBatch89/input.txt' using PigStorage('\n') as (line:chararray);

spaceTokens = FOREACH spaceData GENERATE TOKENIZE(line);

commaTokens = FOREACH commaData GENERATE TOKENIZE(line,',');

STORE spaceTokens INTO '/pigBatch89/prac14/spaceTokens';

STORE commaTokens INTO '/pigBatch89/prac14/commaTokens';

13. flattenDataUseCase.pig

data = LOAD '/pigBatch89/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

orderedData = ORDER data BY age;

groupedData = GROUP orderedData BY gender;

flattenData = FOREACH groupedData GENERATE FLATTEN(orderedData);

--flattenData = FOREACH groupedData GENERATE group, FLATTEN(orderedData);

STORE orderedData INTO '/pigBatch89/prac15/orderedData ';

STORE groupedData INTO '/pigBatch89/prac15/groupedData ';

STORE flattenData INTO '/pigBatch89/prac15/flattenData ';

14. nullCheckUseCase.pig

1000 ABC 12000

1001 EFG

1002 XYZ 15000

1003 18000

PQR 20000

daily = LOAD '/pigBatch89/nullData.txt' using PigStorage('\\t') as (cid:int,cName:chararray,cValue:int);

filNullData = FILTER daily BY cName is null;

filNotNullData = FILTER daily BY cName is not null;

STORE filNullData INTO '/pigBatch89/prac17/nullData';STORE filNotNullData INTO '/pigBatch89/prac17/notNullData';

15. wordCountUseCase.pig

hadoop is bigdata hadoop bigdata is

is hadoop bigdata or bigdata is hadoop

hadoop is bigdata or bigdata is hadoop

inputdata = load '/pigBatch89/wordCount.txt' as (line:chararray);

words = FOREACH inputdata GENERATE FLATTEN(TOKENIZE(line)) AS word;

filtered\_words = FILTER words BY word MATCHES '\\w+';

word\_groups = GROUP filtered\_words BY word;

word\_count = FOREACH word\_groups GENERATE group AS word , COUNT(filtered\_words) AS count;

ordered\_word\_count = ORDER word\_count BY count DESC;

STORE ordered\_word\_count INTO '/pigBatch89/prac18/wordCount';

Semi-Structured Data Processing: XML Data Processing:

Input-XML.

xml-Data.xml

<book>

<name>secret</name>

<price>180</price>

<copies>25000</copies>

</book>

<book>

<name>playing-it-my-way</name>

<price>500</price>

<copies>15000</copies>

</book>

<book>

<name>chase-your-dreams</name>

<price>250</price>

<copies>12000</copies>

</book>

<book>

<name>My-experiments-with-truth</name>

<price>12150</price>

<copies>18000</copies>

</book>

<book>

<name>Mahaprastanam</name>

<price>1250</price>

<copies>19000</copies>

</book>

<book>

<name>Rich-Dad-Poor-Dad</name>

<price>2500</price>

<copies>40000</copies>

</book>

16. xmlProcessing.pig

REGISTER /home/ram/PRAC/pig\_UseCases/required-jars/piggybank-0.15.0.jar;

DEFINE XPath org.apache.pig.piggybank.evaluation.xml.XPath();

data = LOAD '/pigBatch89/xml-Data.xml' USING org.apache.pig.piggybank.storage.XMLLoader('book') as (x:chararray);

xmldata = FOREACH data GENERATE (XPath(x,'book/name')), (XPath(x,'book/price')), (XPath(x,'book/copies'));

filtered = FILTER xmldata BY (int)($2) > 15000;

STORE xmldata INTO '/pigBatch89/prac19/xmldata'; STORE filtered INTO '/pigBatch89/prac19/filteredXmldata';

JSON-Data\_Processing:

**Having Problem with HDFS heap size, So that executed json and UDF in local mode.**

17. jsonUseCase.pig

{"food":"Tacos", "person":"Alice", "amount":3}

{"food":"Tomato Soup", "person":"Sarah", "amount":2}

{"food":"Grilled Cheese", "person":"Alex", "amount":5}

jsonData = LOAD '/home/ram/PRAC/pig\_UseCases/JSON-Processing/json-Data.json' USING JsonLoader('food:chararray,person:chararray,amount:int');

foods = FOREACH jsonData GENERATE food, person, amount;

filteredData = FILTER foods BY amount >=10;

STORE foods INTO '/home/ram/PRAC/pig\_UseCases/JSON-Processing/foods';STORE filteredData INTO '/home/ram/PRAC/pig\_UseCases/JSON-Processing/filtered';

**PIG UDF**

18. COUNTRY-APPEND.pig

1201,gopal,45,Male,hyd,50000

1202,manisha,40,Female,hyd,51000

1203,khaleel,34,Male,chn,30000

1204,prasanth,30,Male,chn,31000

1205,kiran,20,Male,hyd,40000

1206,laxmi,25,Female,blr,35000

1207,bhavya,20,Female,vlr,15000

1208,reshma,19,Female,chn,14000

1209,kranthi,22,Male,pune,22000

1210,Satish,24,Male,mum,25000

1211,Krishna,25,Male,blr,26000

1212,Arshad,28,Male,chn,20000

1213,lavanya,18,Female,hyd,8000

1214,lavanya,18,Female,hyd,12000

**package** com.udf.pig;

**import** java.io.IOException;

**import** org.apache.pig.EvalFunc;

**import** org.apache.pig.data.Tuple;

**public** **class** ErrorAppend **extends** EvalFunc<String>{

@Override

**public** String exec(Tuple input) **throws** IOException {

**if**(input == **null** || input.size() == 0)

**return** **null**;

**try** {

String str =(String) input.get(0);

**return** str.concat("--INDIA");

} **catch**(Exception e) {

**throw** **new** IOException("Caught expression input processing row",e);

}

}

}

REGISTER /home/ram/PRAC/pig\_UseCases/Pig-UDF/errors/COUNTRY-APPEND.jar;

orgData = LOAD '/home/ram/PRAC/pig\_UseCases/Pig-UDF/errors/input.txt' using PigStorage(',') as (eid:int,eName:chararray,age:int,gender:chararray,eloc:chararray,salary:int);

concatData = FOREACH orgData GENERATE eName, age, gender,com.udf.pig.ErrorAppend(eloc);

STORE orgData INTO '/home/ram/PRAC/pig\_UseCases/Pig-UDF/errors/orgData';STORE concatData INTO '/home/ram/PRAC/pig\_UseCases/Pig-UDF/errors/concatData';