

## Ex No: 5 Implement Pig Latin scripts to sort, group, join, project, and filter your data

### AIM:

To implement Pig Latin scripts to load, filter, project, group, sort, and join datasets using Apache Pig.

### Algorithm :

1. **Load the Data**  
Use LOAD command to read data from CSV files using PigStorage(',').  
Define schema (column names and types).
2. **Filter Operation**  
Use FILTER to select tuples based on a condition (e.g., marks > 60).
3. **Projection Operation**  
Use FOREACH ... GENERATE to select specific columns.
4. **Group Operation**  
Use GROUP to group tuples by a particular field (e.g., department).
5. **Sort Operation**  
Use ORDER BY to sort tuples in ascending or descending order.
6. **Join Operation**  
Use JOIN to combine two datasets on a common key (e.g., department).
7. **Display Results**  
Use DUMP to display intermediate and final results.

### Example Input Files

#### students.csv

```
1,Ravi,CSE,85
2,Anita,IT,55
3,John,CSE,72
4,Kiran,ECE,67
5,Meera,IT,90
```

#### departments.csv

```
CSE,Dr.Sharma
IT,Dr.Verma
ECE,Dr.Rao
```

### Python Implementation

```
!wget https://downloads.apache.org/pig/pig-0.17.0/pig-0.17.0.tar.gz
!tar -xzf pig-0.17.0.tar.gz
!mv pig-0.17.0 /content/pig
```

```

import os
os.environ['PIG_HOME'] = '/content/pig'
os.environ['PATH'] += os.pathsep + os.path.join(os.environ['PIG_HOME'], 'bin')

# =====
# 2. Create Input CSV Files
# =====
students = """1,Ravi,CSE,85
2,Anita,IT,55
3,John,CSE,72
4,Kiran,ECE,67
5,Meera,IT,90
"""
with open("students.csv", "w") as f:
    f.write(students)

departments = """CSE,Dr.Sharma
IT,Dr.Verma
ECE,Dr.Rao
"""
with open("departments.csv", "w") as f:
    f.write(departments)

# =====
# 3. Write the Pig Latin Script
# =====
pig_script = r"""
-- Load student and department data
students = LOAD 'students.csv' USING PigStorage(',')
          AS (id:int, name:chararray, dept:chararray, marks:int);

departments = LOAD 'departments.csv' USING PigStorage(',')
            AS (dept:chararray, hod:chararray);

-- Filter: select students with marks > 60
good_students = FILTER students BY marks > 60;

-- Project: select only name, dept, marks
projected = FOREACH good_students GENERATE name, dept, marks;

-- Group: group by department
grouped = GROUP projected BY dept;

-- Sort: order by marks descending
sorted = ORDER projected BY marks DESC;

-- Join: combine students with department HODs
joined = JOIN projected BY dept, departments BY dept;

-- Dump results
DUMP sorted;
DUMP grouped;

```

```
DUMP joined;
""""
```

```
with open("program.pig", "w") as f:
    f.write(pig_script)
```

```
# =====
# 4. Set Java Environment & Run Pig Script (Local Mode)
# =====
!export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
!export PATH=$JAVA_HOME/bin:$PATH
```

```
os.environ['JAVA_HOME'] = '/usr/lib/jvm/java-11-openjdk-amd64'
os.environ['PATH'] = os.environ['JAVA_HOME'] + '/bin:' + os.environ['PATH']
```

```
!pig -x local program.pig
```

### **Expected Output:**

#### **Sorted Output**

```
(Meera,IT,90)
(Ravi,CSE,85)
(John,CSE,72)
(Kiran,ECE,67)
```

#### **Grouped Output**

```
(CSE, {(Ravi,CSE,85),(John,CSE,72)})
(IT, {(Meera,IT,90)})
(ECE, {(Kiran,ECE,67)})
```

#### **Joined Output**

```
(Ravi,CSE,85,CSE,Dr.Sharma)
(John,CSE,72,CSE,Dr.Sharma)
(Kiran,ECE,67,ECE,Dr.Rao)
(Meera,IT,90,IT,Dr.Verma)
```

### **Result:**

Thus, a Pig Latin script was successfully implemented to sort, group, join, project, and filter data, demonstrating Pig's ability to process structured datasets efficiently.