**WEEK-6**

**i.Running Pig in Command Line Interface (CLI)**

To start Pig in either Local or MapReduce mode from the terminal:

**MapReduce Mode**

1. SSH into one of the nodes in the Hadoop cluster (often the master node).
2. Run the following command to start Pig in **MapReduce Mode**

pig -x mapreduce

**Local Mode**

If you want to run Pig locally (for testing), use:

pig -x local

**Basic Pig Commands in Cloudera**

Once you’re in the Pig environment (either CLI or Hue), you can start writing and running Pig Latin commands.

**Example Pig Commands**

**Load Data**

data = LOAD '/user/cloudera/input/data.csv' USING PigStorage(',') AS (name:chararray, age:int, city:chararray);

Display Data

DUMP data;

Filter Data

filtered\_data = FILTER data BY age > 25;

DUMP filtered\_data;

Store Data

STORE filtered\_data INTO '/user/cloudera/output/filtered\_data' USING PigStorage(',');

Group Data

grouped\_data = GROUP data BY city;

DUMP grouped\_data;

Aggregate Data

average\_age = FOREACH grouped\_data GENERATE group AS city, AVG(data.age) AS avg\_age;

DUMP average\_age;

ii. Pig Latin scripts demonstrating sorting, grouping, joining, projecting, and filtering data in Apache Pig.

**employees.csv**: Contains employee information.

emp\_id, name, age, department

1, Alice, 30, HR

2, Bob, 35, IT

3, Charlie, 28, IT

4, David, 40, HR

5, Eve, 25, Marketing

**2.departments.csv**: Contains department information.

dept\_id, department, location

1, HR, New York

2, IT, San Francisco

3, Marketing, Boston

**1. Load Data**

First, load both datasets in Pig:

employees = LOAD 'employees.csv' USING PigStorage(',') AS (emp\_id:int, name:chararray, age:int, department:chararray);

departments = LOAD 'departments.csv' USING PigStorage(',') AS (dept\_id:int, department:chararray, location:chararray);

2. Filter Data

Filter employees to include only those older than 30:

older\_employees = FILTER employees BY age > 30;

DUMP older\_employees;

1. Project Specific Fields

Project (select) only the name and age fields from the employees dataset:

name\_age = FOREACH employees GENERATE name, age;

DUMP name\_age;

1. Sort Data

Sort employees by age in descending order:

sorted\_employees = ORDER employees BY age DESC;

DUMP sorted\_employees;

1. Group Data

Group employees by their department:

grouped\_by\_department = GROUP employees BY department;

DUMP grouped\_by\_department;

1. Join Data

Join the employees dataset with the departments dataset on the department name:

joined\_data = JOIN employees BY department, departments BY department;

DUMP joined\_data;

1. Store Data

To save the result of a particular operation to HDFS:

STORE joined\_projected INTO 'output/joined\_projected' USING PigStorage(',');

Complete Example Script

Here’s a complete script file, example.pig, that combines these operations:

-- Load datasets

employees = LOAD 'employees.csv' USING PigStorage(',') AS (emp\_id:int, name:chararray, age:int, department:chararray);

departments = LOAD 'departments.csv' USING PigStorage(',') AS (dept\_id:int, department:chararray, location:chararray);

-- Filter employees older than 30

older\_employees = FILTER employees BY age > 30;

-- Project name and age fields

name\_age = FOREACH employees GENERATE name, age;

-- Sort employees by age descending

sorted\_employees = ORDER employees BY age DESC;

-- Group employees by department and calculate average age

grouped\_by\_department = GROUP employees BY department;

average\_age = FOREACH grouped\_by\_department GENERATE group AS department, AVG(employees.age) AS avg\_age;

-- Join employees with departments on department name

joined\_data = JOIN employees BY department, departments BY department;

-- Project specific fields after join

joined\_projected = FOREACH joined\_data GENERATE employees::name, employees::age, departments::location;

-- Store the final result

STORE joined\_projected INTO 'output/joined\_projected' USING PigStorage(',');

Run this script using:

pig -x mapreduce example.pig