**PigStorage() in Apache Pig**

PigStorage() is the **default** **data loader and storage function** in Apache Pig. It is used to **read and write** data in a structured format using a **delimiter** (like ,, \t, |, etc.).

John,25,New York

Alice,30,Los Angeles

Bob,17,Chicago

Eve,40,New York

Tom,15,San Francisco

Pig Script

-- Load the data from the text file

employees = LOAD 'employees.txt' USING PigStorage(',') AS (name:chararray, age:int, city:chararray);

-- Filter employees who are 18 or older

adults = FILTER employees BY age >= 18;

-- Group by city

grouped\_by\_city = GROUP adults BY city;

-- Count the number of employees in each city

city\_count = FOREACH grouped\_by\_city GENERATE group AS city, COUNT(adults) AS total\_people;

-- Store the result in an output file

STORE city\_count INTO 'output' USING PigStorage(',');

>pig -x local employee\_analysis.pig

Sort Employees by Age

employees = LOAD 'employees.txt' USING PigStorage(',') AS (name:chararray, age:int, city:chararray);

sorted\_employees = ORDER employees BY age DESC;

STORE sorted\_employees INTO 'output' USING PigStorage(',');

1,John,85

2,Alice,90

3,Bob,75

4,Eve,95

Pig Script to Sort by Marks in Descending Order

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Sort the data based on marks in descending order

sorted\_students = ORDER students BY marks DESC;

-- Store or display the sorted output

DUMP sorted\_students;

**the LIMIT**

In **Apache Pig**, the LIMIT command is used to restrict the number of records in the output. It is often used after sorting to get the **top N records**.

Example: Get Top 3 Students with Highest Marks

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Sort the data based on marks in descending order

sorted\_students = ORDER students BY marks DESC;

-- Get the top 3 students

top\_3\_students = LIMIT sorted\_students 3;

-- Display the output

DUMP top\_3\_students;

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Sort the data based on marks in ascending order

sorted\_students = ORDER students BY marks ASC;

-- Display the output

DUMP sorted\_students;

**SPLIT Command in Apache Pig**

The SPLIT command in **Apache Pig** is used to **divide a dataset into multiple subsets** based on specified conditions.

### ****Example: Split Students into Pass and Fail Groups****

#### ****Dataset (students.txt)****

1,John,85

2,Alice,90

3,Bob,35

4,Eve,95

5,Tom,40

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Split students into pass and fail groups (Pass if marks ≥ 50)

SPLIT students INTO pass\_students IF marks >= 50,

fail\_students IF marks < 50;

-- Display pass students

DUMP pass\_students;

-- Display fail students

DUMP fail\_students;

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Filter students who have exactly 90 marks

students\_with\_90 = FILTER students BY marks == 90;

-- Display the result

DUMP students\_with\_90;

-- Students who have 90 marks AND name is 'Alice'

alice\_90 = FILTER students BY marks == 90 AND name == 'Alice';

-- Students who have either 90 or 85 marks

marks\_90\_or\_85 = FILTER students BY marks == 90 OR marks == 85;

DUMP alice\_90;

DUMP marks\_90\_or\_85;

**What FOREACH Can Do**

✅ Select specific columns  
✅ Perform computations (arithmetic, string operations, conditions)  
✅ Create new derived columns

**FOREACH Command in Apache Pig**

The FOREACH command in **Apache Pig** is used to **apply transformations** on each row of a dataset. It is commonly used with GENERATE to **select specific columns**, **modify values**, or **create new computed fields**.\

alias = FOREACH input\_data GENERATE expression1, expression2, ...;

### ****Example 1: Selecting Specific Columns****

#### ****Dataset (students.txt)****

1,John,85

2,Alice,90

3,Bob,75

4,Eve,95

5,Tom,40

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Select only Name and Marks

selected\_data = FOREACH students GENERATE name, marks;

-- Display the result

DUMP selected\_data;

(John,85)

(Alice,90)

(Bob,75)

(Eve,95)

(Tom,40)

2. Creating a New Column

result = FOREACH students GENERATE name, marks, marks + 5 AS updated\_marks;

3. Using Conditional Expression (? :)

result = FOREACH students GENERATE name, marks,

(marks >= 50 ? 'Pass' : 'Fail') AS result;

**Example 3: Performing Arithmetic Operations**

Let's add 5 bonus marks to each student.

bonus\_marks = FOREACH students GENERATE

name, marks + 5 AS updated\_marks;

DUMP bonus\_marks;

**4.Applying FOREACH After FILTER**

-- Load the dataset

students = LOAD 'students.txt' USING PigStorage(',')

AS (id:int, name:chararray, marks:int);

-- Select only Name and Marks

selected\_data = FOREACH students GENERATE name, marks;

-- Filter students who scored more than 80

high\_scorers = FILTER selected\_data BY marks > 80;

-- Display the result

DUMP high\_scorers;