EXPERIMENT 11

LAB PROGRAM ON APACHE PIG – AGRICULTURE DATASET ANALYSIS

Aim:

To analyze an agricultural dataset using **Apache PIG** in **local mode**, performing various operations such as **grouping, filtering, aggregation, and statistical analysis**.

Procedure:

1. **Download the Dataset**
    - The dataset contains **agricultural data** related to **crops, production, and regions**.
    - It has **7 columns**:
        - **State_Name** (chararray)
        - **District_Name** (chararray)
        - **Crop_Year** (int)
        - **Season** (chararray)
        - **Crop** (chararray)
        - **Area** (int)
        - **Production** (int)
    - The dataset has **246092 rows** and **7 columns**.

2. **Enter PIG Local Mode**

   pig -x local

   This opens the **grunt shell** for executing **PIG** commands.

3. **Load the Dataset**

   agriculture = LOAD 'F:/csv files/crop_production.csv' USING PigStorage(',')

   AS (State_Name:chararray , District_Name:chararray , Crop_Year:int ,

   Season:chararray , Crop:chararray , Area:int , Production:int);

   This command loads the dataset into **PIG**.

4. **Dump and Describe the Dataset**

   DUMP agriculture;

   DESCRIBE agriculture;

   - **DUMP** displays the data.
   - **DESCRIBE** provides the schema of the dataset.

5. **Explaining the Dataset**

   EXPLAIN agriculture;

   - **EXPLAIN** provides an execution plan, showing how **PIG** processes the dataset.

Executing PIG Queries

1. Grouping Records by State

statewisecrop = GROUP agriculture BY State_Name;

DUMP statewisecrop;

DESCRIBE statewisecrop;

- Groups all records based on **State_Name**.

2. Filtering Data for a Specific Year (e.g., 2015)

agriculture_2015 = FILTER agriculture BY Crop_Year == 2015;

DUMP agriculture_2015;

- Retrieves all **crop records** from the year **2015**.

3. Filtering Data for a Specific Crop (e.g., Rice)

rice_data = FILTER agriculture BY Crop == 'Rice';

DUMP rice_data;

- Extracts records where the **Crop** is **Rice**.

4. Finding the Total Production per State

```
statewise_production = FOREACH (GROUP agriculture BY State_Name)
GENERATE group AS State, SUM(agriculture.Production) AS Total_Production;
DUMP statewise_production;
```

- Calculates **total production** per **state**.

5. Finding the Maximum Production for Each Crop

```
max_production = FOREACH (GROUP agriculture BY Crop)
GENERATE group AS Crop_Name, MAX(agriculture.Production) AS Max_Production;
DUMP max_production;
```

- Retrieves the **maximum production** for each **crop**.

6. Calculating the Average Area for Each Crop

```
avg_area = FOREACH (GROUP agriculture BY Crop)
GENERATE group AS Crop_Name, AVG(agriculture.Area) AS Avg_Area;
DUMP avg_area;
```

- Computes **average cultivated area** for each **crop**.

7. Counting the Number of Records Per Season

```
season_count = FOREACH (GROUP agriculture BY Season)
GENERATE group AS Season, COUNT(agriculture) AS Record_Count;
DUMP season_count;
```

- Counts the **number of records** per **season**.

8. Explaining a Query Execution Plan

```
EXPLAIN statewise_production;
```

- **EXPLAIN** shows how **PIG** processes the **statewise_production** query internally.

Result:

The dataset was successfully loaded into Apache PIG in local mode, and various operations were performed to analyze agricultural data. Grouping, filtering, aggregation, and statistical calculations were executed to gain insights into state-wise production, crop-wise maximum production, and seasonal records.