POORNIMA

Date :-

Experiment No. 07

Aim: Install and run Pig then write Pig Latin scripts to sort group, join, project and filter your data.

7(a): Install & run the pig on word count.

Phase - I Installation of Pig

1) Open Cloudera on virtual box
2) Open browser
3) Click on hive - web application developed by cloudera
4) Login with user id & password as cloudera
5) Click on Manage HDFS
6) By default in user / cloudera folder
7) Create folder as sec - B / Pig
8) Create file word count.txt save it
9) open terminal - write pig on command prompt to start pig.

Phase - II

1) Working with Grunt shell
2) Create word count application.
3) Execute word count application.
4) Accessing HDFS from grunt shell.

POORNIMA

Step 1: Start Grunt shell
Open terminal & type pig

Create a file at /user/cloudera/Sec-B/Pig/wordcount.txt with following content:

I am learning Pig using Hadoop Exam
I am learning spark using Hadoop Exam
I am learning Java using Hadoop Exam.
I am learning Hadoop using Hadoop Exam.

Step 2: Now load the file stored in HDFS (Space separated file)

input1 = load /user/cloudera/Sec-B/Pig/wordcount.txt as (F1: chararray);

Dump input1;
I am learning pig using Hadoop Exam.
I am spark using Hadoop Exam.
I am learning Java using Hadoop Exam.

Step 3: Flatten the words in each line

Words In Each line = FOR EACH input1 GENERATE flatten (TOKEN (F1)) as word;
DUMP words In Each Line;

POORNIMA

Step 4) Group the same words

Grouped words = group words /n Each line by word;

dump grouped words;

describe grouped words;

Step 5) Now do the word count

Counted words = For each grouped words generate group COUNT C words /n Each line);

dump Counted words.

Pig Latin

Pig scripts can be linear workflows.

Pig scripts can have flattening like multiple data input arajoined (normalizing) and data splitting.

Grunt: It is shell where we have been writing pig scripts. Generally production code will be written in a separate file. But while writing we want to test our scripts with test data, hence we will be using Grunt shell for prototyping our scripts.

Accessing HDFS: You can use HDFS Commands inside grunt shells as >fs-ls

Output

(21, {(4, Preethi Agarwal, 21, 9848022330, Pune), (1, Rajiv, Reddy, 21, 9840223104, Hyderabad)})
(22, {(3, Rajesh Khanna, 22, 9848022339, Delhi); (3, Sidarth Bhatacharya, 22 9848022339, Kolkata)})
(22, {6, Archana Mishra, 23, 9840 22335, Chennai)

Output:

(21, Pune) {(4, Preethi Agarwal, 21, 9848022330, Pune)})
(21, Hyderabad), {(1, Rajiv Reddy, 21, 9840223337, Hyderabad)})

(22, Delhi), {(3, Rajesh Khanna, 22, 9840802231 Kolkata)})

Output:

(1, Ramesh, 32, Ahmedabad, 2000, 1, Ramesh, 32, Ahmedabad, 2000)
(2, Khilan, 25, Delhi, 1500, 2, Khilan, 25, Delhi, 1500)
(3, Kaushik, 23, Kota, 2000, 3, Kaushik, 25, Kota, 2000)
(4, Charatai, 25, Mumbai, 6500, 4, Chaitali, 25, mumbai, 6500)

POORNIMA

Date:

Experiment NO 7 (b)

Objective: Write Pig latin scripts to sort, group, join, project and filter your data

Apache Pig - Groups operator

Student_details = LOAD '/user/cloudera/sec-B/Pig/Student_details.txt'

using PigStorage(',') as (id: int, first name: chararray, last name, chararray, age: int, phone: chararray, City: chararray); group_data = GROUP student_details by age; dump group_data;

Grouping by multiple columns

group_multiple = GROUP Student_details by age, city; dump group_multiple;

JOIN:

Customers = LOAD '/user/cloudera/sec-B/Pig/Customer.txt' using PigStorage(',') as (id: int, Name: chararray, age: int, address: chararray, salary: int);

Output

(2, Khilan, 25, Delhi, 1500, 101, 2009-11-20, 00:00:00, 2, 1560)
(3, Kaushik, 23, Kota, 2000, 100, 2009-10-08, 00:00:00, 3, 1500)
(3, Kaushik, 23, Kota, 2000, 102, 2009-10-08 00:00:00, 3, 3000)

Output:

(1, Ram, 32, Ahmedabad, 2000, 1111)
(2, Krishna, 31, Vidavan, 1500, 101, 2009-11-20 00:00:00, 2, 1560)
(3, Kaushik, 23, Kota, 2000, 102, 2009-10-08 00:00:00, 3, 3000)
(5, Hitesh, 27, Bhopal, 8500, 111),

Output:

(2, Khilan, 25, Delhi, 1500, 101, 2009-11-20 00:00:00, 2, 1560)
(3, Kaushik, 23, Kota, 2000, 100, 2009-10-08 00:00:00, 3, 1500)
(4, Kautilya, 25, Mumbai, 6500, 103, 2008-05-20 00:00:00, 4, 2060)

Output

(1, Ram, 32, Ahmedabad, 2000, 111)
(2, Khilan, 25, Delhi, 1500, 101, 2009-11-20 00:00:00 2, 1560)
(3, Kaushik, 23, Kota, 2000, 102, 2009-10-08 00:00:00, 3, 3000)
(5, Krishna, 24, Vrindavan, 5000, 105, 2009-10-07 00:00:00, 4, 4600)

POORNIMA

Orders = LOAD '/user/Cloudera/Sec-B/Pig/orders.txt' using Pig storage (',') as (Oid: int, Data: chararray, Customer_id: int amount: int);

Self Join:

Customer1 = LOAD '/user/cloudera/Sec-B/Pig/customer.txt' using pig storage (',') as (id: int, name: chararray, age: int, address: chararray, subarray, salary: int).

Customer2 = LOAD '/user/cloudera/Sec-B/Pig/Customer.txt' using Pig storage (',') as (id: int, name: chararray, age: int; address: chararray, salary: int);

Customer3 = JOIN Customer1 By id, Customer2 By id;

dump customer3;

Inner Join:

Customer_orders = JOIN Customers By id, Orders By customer_id;

dump customer_orders;

Output:

(8, Bharatti, Namidiyar, 24, 98480, 22333, Chennai)

Outer JOIN:

Left - Outer JOIN

Outer - Left = JOIN Customer By id LEFT Outer, Orders By customer_id;

Dump Outer_Left;

Right Outer JOIN:

Outer - Right = JOIN Customers By id RIGHT, Order By Customer_id;

Dump Outer - right.

Full Outer JOIN

Outer - Full = JOIN Customer By id Full outer, Order By customer_id;

Dump outer_full;

Apache - Pig - Filter operator

Student_details = LOAD '/user/cloudera/sec-B/pig/Student_details.txt' using PigStorage(',') as id: int, firstname: chararray, lastname: Chararray, age: int, phone: chararray, city: Chararray); filter_data = FILTER Student_details By City = 'Chennai';

Dump filter_data;

POORNIMA

Date:

Experiment No. 08

Objective: Install and Run Hive then use hive to create, alter & drop database, tasks, Views, function and indexes

Hive is pre-installed in cloudera framework

1) open terminal - run command hive -- enter
2) In hive mode new
   hive >
3) To create database
   hive > CREATE DATABASE USER;
   OR
   Time Taken : 4.478 seconds

4) Show database in hive
   hive > show database;
   OR
   default
   user
   Time Taken : 0.77 seconds, Fetched : 2 rows(s).

5) Drop database :-
   hive > Drop database if exists users;
   OR
   Time Taken : 0.761 seconds.

POORNIMA

6) Create Table:-

hive > CREATE TABLE IF NOT EXISTS employee
(id: int, name: String, salary: String, distination,
String).

> COMMENT 'EMPLOYEE_details'
> ROW FORMAT DELIMITED
> FIELDS TERMINATED By '\t'
> LINES TERMINATED By '\n'
> STORED AS TEXTFILE ;

OK
Time Taken : 2.316 seconds

7) Show Tables:-

hive > show table ;

OK
emp
Time Taken : 0.77 seconds, Fetched : 1 row(s)

8) Create txt file in /home/cloudera/hive/sample.txt
- write data in file.

9) Load data local IN PATH '/Home/cloudera/hive/
Sample.txt' OVER WRITE INTO TABLE emp;

POORNIMA

loading data to table default.emp

Table default.emp stats: [num files = 1, num rows = 0, Total size = 20, raw data size = 0]

OK

Time Taken : 1.409 seconds

10) Hive — Alter Table

The Statement takes any of the following syntax based to what attributes we wish to modify in a txt file:

ALTER TABLE name RENAME To new-name

ALTER TABLE name ADD COLUMNS (col-spaec [col-Spac..])

ALTER TABLE name DROP [COLUMN] Column_name

ALTER TABLE name CHANGE Column name new-name new type]

ALTER TABLE name REPLACE COLUMNS (col-spac [col-spac...])

11) The following query rename the table from employee to emp

hive > ALTER TABLE employee RENAME to emp ;

OR

Time Taken : 0.23 seconds

hive > show tables ;

OR

OK
P
Patients
Prescription
Time Taken : 0.05 seconds, fetched : 4 rows (s)

(12) Change Statement

Change Column name :-
hive > ALTER TABLE emp CHANGE name pname : string ;
OK

(13) Change type of Column :

hive > ALTER TABLE emp CHANGE salary salary double ;
OK
Time Taken : 0.269 seconds

(14) To add new Column in table :

This query to add a Column named dept to the emp table
hive > ALTER TABLE emp ADD COLUMNS (dept STRING COMMENT 'Department name') ;
OK
Time Taken : 0.281 seconds

POORNIMA

(15.) Replace Statement:

This query deletes all the columns from the emp table & replaces it with emp and name columns:

hive > ALTER TABLE emp RERLACE COLUMNS (eid :int, empid : int, ename ; STRING, name : STRING);

(16.) Drop Table Statement :-

This query drops a table named "employee".

hive > DROP TABLE IF EXISTS emp ;