10. Illustrate the PIG string functions and date and time functions with student database.

student.csv

1,John Doe,1999-05-14,CS
2, Jane Smith ,2000-11-22,IT
3,Rahul Kumar,1998-07-19,EE
4,Anita Sharma,2001-01-09,ME
5,mary-ann,1997-12-31,CS
6,Bob O'Neil,1999-03-02,IT

Upload to HDFS

hdfs dfs -put student.csv /user/cloudera/student.csv

Load the data

grunt> student_data = LOAD '/user/cloudera/student.csv' USING PigStorage(',') AS (id:int, name:chararray, dob:chararray, dept:chararray);

grunt> dump student_data;

Output:

(1,John Doe,1999-05-14,CS)
(2,Jane Smith,2000-11-22,IT)
(3,Rahul Kumar,1998-07-19,EE)
(4,Anita Sharma,2001-01-09,ME)
(5,mary-ann,1997-12-31,CS)
(6,Bob O'Neil,1999-03-02,IT)

String Functions:

Built-in string functions: UPPER, LOWER, TRIM, SUBSTRING, REPLACE, INDEXOF, etc.

(a) UPPER() and LOWER()

grunt> upper_lower = FOREACH student_data GENERATE id, UPPER(name) AS name_up, LOWER(dept) AS dept_low;

grunt> dump upper_lower;

Output:

(1,JOHN DOE,cs)
(2,JANE SMITH,it)
(3,RAHUL KUMAR,ee)
(4,ANITA SHARMA,me)
(5,MARY-ANN,cs)
(6,BOB O'NEIL,it)

(b) TRIM(): It removes leading/trailing spaces.

```
grunt> trimmed = FOREACH student_data GENERATE id, TRIM(name) AS clean_name;

grunt> DUMP trimmed;
```

Output:

```
(1,John Doe)
(2,Jane Smith)
(3,Rahul Kumar)
(4,Anita Sharma)
(5,mary-ann)
(6,Bob O'Neil)
```

(c) SUBSTRING(string, start, end): It returns characters starting at index start (0-based) up to index end-1. Use TRIM() first if input has leading spaces.

```
grunt>first4 = FOREACH student_data GENERATE id, SUBSTRING(TRIM(name), 0, 4) AS first4;

grunt>DUMP first4;
```

Output:

```
(1,John)
(2,Jane)
(3,Rahu)
(4,Anit)
(5,mary)
(6,Bob )
```

(d) REPLACE(string, 'old','new'): It replaces occurrences of a substring.

```
grunt>replaced = FOREACH student_data GENERATE id, REPLACE(TRIM(name), ' ', '_') AS name_underscored;

grunt>DUMP replaced;
```

Output:

```
(1,John_Doe)
(2,Jane_Smith)
(3,Rahul_Kumar)
(4,Anita_Sharma)
(5,mary-ann)
(6,Bob_O'Neil)
```

(e) INDEXOF(string, substring, startIndex): It returns **0-based** index of the first occurrence of the substring (search is case-sensitive). If not found returns -1.

grunt>index_a = FOREACH student_data GENERATE id, TRIM(name) AS name, INDEXOF(TRIM(name), 'a', 0) AS pos_a;

grunt>DUMP index_a;

Output:

(1,John Doe,-1)
(2,Jane Smith,1)
(3,Rahul Kumar,1)
(4,Anita Sharma,4)
(5,mary-ann,1)
(6,Bob O'Neil,-1)

Date & Time Functions:

Pig provides ToDate() to convert strings to DateTime objects and other functions are GetYear, GetMonth, GetDay, AddDuration, CurrentTime, ToString, YearsBetween, etc.

Convert date field with ToDate()

ToDate():

- It converts your plain yyyy-MM-dd string into a **datetime object**.
- Pig stores it in **ISO-8601 format**:

  yyyy-MM-ddTHH:mm:ss.SSS±hh:mm

Example: 1999-05-14T00:00:00.000-07:00

- T separates date & time
- 00:00:00.000 = midnight time
- -07:00 = time zone offset (based on your system/JVM defaults, often PDT or PST on Cloudera VMs).

grunt>student_date = FOREACH student_data GENERATE id, name, ToDate(dob, 'yyyy-MM-dd') AS birth_date, dept;

grunt>dump student_date;

Output:

(1,John Doe,1999-05-14T00:00:00.000-07:00,CS)
(2,Jane Smith,2000-11-22T00:00:00.000-08:00,IT)
(3,Rahul Kumar,1998-07-19T00:00:00.000-07:00,EE)
(4,Anita Sharma,2001-01-09T00:00:00.000-08:00,ME)
(5,mary-ann,1997-12-31T00:00:00.000-08:00,CS)
(6,Bob O'Neil,1999-03-02T00:00:00.000-08:00,IT)

(a) GetYear, GetMonth, GetDay: It extracts numeric year / month / day from the DateTime.

grunt>ymd = FOREACH student_date GENERATE id, name, GetYear(birth_date) AS yyyy, GetMonth(birth_date) AS mm, GetDay(birth_date) AS dd;

grunt>DUMP ymd;

Output:

(1,John Doe,1999,5,14)
(2,Jane Smith,2000,11,22)
(3,Rahul Kumar,1998,7,19)
(4,Anita Sharma,2001,1,9)
(5,mary-ann,1997,12,31)
(6,Bob O'Neil,1999,3,2)

(b) AddDuration(datetime, 'P...') — add ISO-8601 durations: It adds a duration (ISO-8601: P1Y = 1 year, P1M = 1 month, PT5H = 5 hours, etc.).

grunt>plus1 = FOREACH student_date GENERATE id, name, ToString(AddDuration(birth_date, 'P1Y'), 'yyyy-MM-dd') AS plus1yr;

grunt>DUMP plus1;

Output:

(1,John Doe,2000-05-14)
(2,Jane Smith,2001-11-22)
(3,Rahul Kumar,1999-07-19)
(4,Anita Sharma,2002-01-09)
(5,mary-ann,1998-12-31)
(6,Bob O'Neil,2000-03-02)

(c) CurrentTime() and computing ages with YearsBetween()

- CurrentTime() returns a DateTime object for the system time at runtime.
- YearsBetween(d1,d2) returns number of whole years between two DateTime objects.

grunt>ages = FOREACH student_date GENERATE id, name, YearsBetween(birth_date, CurrentTime()) AS age;

grunt>DUMP ages;

Output:

(1,John Doe,-26)
(2,Jane Smith,-24)
(3,Rahul Kumar,-27)
(4,Anita Sharma,-24)
(5,mary-ann,-27)
(6,Bob O'Neil,-26)