Greg Cousin: 18204188
gregoire.cousin@ucdconnect.ie
11/09/18

Practical 5

1. How can I print the results in Pig Latin script? Use the primitive DUMP <bag of tuples>

a.

```
1. Log = LOAD 'animalLog' USING PigStorage();
2. animalLog = GROUP Log BY animKey;
3. DUMP Log
```

2. Print the log file sample grouped by movie.

a.

```
Log_sample = LOAD 'movie_log_file' USING PigStorage();
Movie_logs = GROUP log_sample BY movieID;
DUMP log_sample
```

3. What is the primitive “DESCRIBE” in Pig Latin?

a. this would give you the information of a relation in a db = schema {}

4. Extend the previous script to process the clickstream data into user sessions.

a.

```
  UserID = "which user has set up in the session"
  sessionID = duration of time from start to end of like... "being on a session"
  timestamp = "time when the event happened"
  clickDataSteam = "logging a line for each time a user clicks on something"

  b = FOREACH a GENERATE mytimestamp, user_id ; // generate each time stamp
  c = FOREACH (GROUP b BY user_id) { // group each (data) by user ID
  ordered = ORDER b BY mytimestamp ; // sordering each by their timestamp
  GENERATE FLATTEN(Sessionize(ordered)) AS (mytimestamp, user_id, session_id); // flatten because you have one record per user but need
one record PER EVENT
  }; // (oposite than group by)

  d = ORDER c BY user_id, mytimestamp; // ordering timestamp and user ID
  DKUSTORE d INTO 'toy_data_sessionized'; // storing the output (DSS) => one format... you can use (pig storage ())
```

a.

```
<!--
Log = LOAD 'animal_log' USING PigStorage();
Users = FILTER Log BY (timestamp, sessionID, UserID, clickDataSteam)
Session = FOREACH Users GENERATE user, (clickDataStream) as userSession Dump Session
-->
```

5. How can I use FOREACH statement in Pig Script?

a.

foreach operator is utilized to engender designated data iterations predicated on each of the the column data.
Example: iteration == FOREACH relation GENERATE (our data)

6. Select only the clicks which correspond to starting, browsing, completing, or purchasing movies.

a.

```
Users = FILTER Log BY (timestamp, sessionID, UserID, clickDataSteam)
MovieData = FILTER Log AS startMovie, browseMovie, completedMovie, purchaseMovie
Session = FOREACH Users GENERATE ourUser, (MovieData) as userSession Dump Session
```

Exercise 2 => Let “students.csv” is a file that contains students data. We assume that the data values are separated by “comma”.

1. Create a pig script to load students.csv data.

a. studentLog = LOAD 'studentLog' USING PigStorage(); // from studentLog fold

2. Create a pig script to filter out the first row of the data.

a.

```
1. WithoutFistRow = FILTER studentLog BY $0 != 'FirstHeaderInCsvFile*.*'
```

4. Write a pig script to assign names to the data fields of the students.csv data. The output file should be called “students_details”.

a.

```
allStudents = LOAD 'studentLog' USING PigStorage();
studentDetails = FOREACH allStudents GENERATE $0 as nameofStudent, $1 studentPhone  >>>> and so on
```

```
<!-- ===================================================================== -->
<!-- Assume that we have another file recording the students’ attendance; “students_attendance.csv”. -->
<!-- ===================================================================== -->
```

4. Perform the previous 3 operations on the file “students_attendance.csv”. The output of the 3rd operation in this case should be called “SA_details”.

a.

```
studentAttendance = LOAD 'students_attendance' USING PigStorage(); // students_attendance = means file // after the filtering of the first row
offFirst = studentAttendance;
newStu = FILTER offFirst BY $0 != 'FirstHeaderInCsvFile*.*';
SA_Details = FOREACH newStu GENERATE $0 as day, $1 as hour
```

5. Extend your script, if necessary, to filter the data (all hours attended for each student).

a.

```
studentAttendance = LOAD 'students_attendance' USING PigStorage();
// ===> after the filtering of the first row -->
offFirst = studentAttendance;
withMaxHrs = FOREACH (GROUP offFirst BY student_id) {
  hrs = ORDER offFirst BY student_id, hour
  SUM(CASE hrs
      WHEN hour THEN 1
      ELSE 0 END)
}
newStu = FILTER withMaxHrs BY $1 = 10;
```

6. Write a script to find the sum of hours attended by each student.

a.

```
FOREACH (GROUP offFirst BY student_id) {
hrs = ORDER offFirst BY student_id, hour
SUM(CASE hrs
      WHEN hour THEN 1
      ELSE 0 END)
}
```

8. Write a script to join StudentID, Name with the hours attended.

a.

```
joined = JOIN studen_name BY student_id, attendance BY student_id
```

10. Print the results on the screen.

a.

```
DUMP joined
```