Bhasheyam Krishnan

A20380078

Magic number: 55179

Que1:

Comand :

foot\_ratings = LOAD '/user/maria\_dev/foodratings55179.txt' USING PigStorage(',')

AS (name:chararray,f1:int, f2:int, f3:int, f4:int, placeid :int

);

DESCRIBE foot\_ratings;

output:



Que 2:

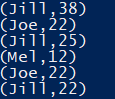
Command:

food\_ratings\_subset = FOREACH foot\_ratings GENERATE name, f4;

STORE food\_ratings\_subset INTO ‘my\_food\_rating\_output’ USING PigStorage(‘\*’);

que2\_output = LIMIT food\_ratings\_subset 6;

dump que2\_output;



Que 3:

Command:

Group\_rating = GROUP foot\_ratings All;

food\_ratings\_profile = FOREACH grop\_ratings GENERATE MIN(foot\_ratings.f2),MAX(foot\_ratings.f2),AVG(foot\_ratings.f2),MIN(foot\_ratings.f3),MAX(foot\_ratings.f3),AVG(foot\_ratings.f3);

dump food\_ratings\_profile;



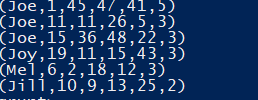
Que 4:

Command:

SPLIT foot\_ratings INTO food\_ratings\_filtered IF (f1 < 20 and f3 > 5) , x IF (f1 > 20 and f3 < 5);

que4\_output =LIMIT food\_ratings\_filtered 6;

dump que4\_output



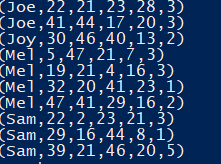
Que 5:

Command:

food\_ratings\_2percent = SAMPLE foot\_ratings 0.02;

ques5\_output = LIMIT foot\_ratings\_2percent 10;

dump qeus5\_output;



Que6

Command:

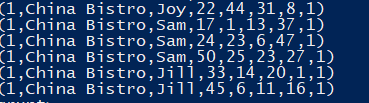
foot\_places = LOAD '/user/maria\_dev/foodplaces55179.txt' USING PigStorage(',') AS (placeid:int,placename:chararray );

DESCRIBE foot\_places;

food\_ratings\_w\_place\_names = JOIN food\_places BY (placeid), foot\_ratings BY (placeid);

ques6\_output = LIMIT food\_ratings\_w\_place\_names 6;





Que 7:

* Spark System(RDD’s) is motivated by complex multi stage application and interactive ad-hoc queries as map-reduce simplified big data analysis but as soon as it got popular , used wanted it perform more.
* To perform such task a good data sharing technique is required which map – reduce lack as it use instance storage which is slow. Instead RDDs use the in -memory which faster to share data.
* To process all data in in-memory is not feasible all the time, so to overcome this, RDDs use the portioned data like filter, map etc.
* Spark is implementation helps to over come the issue faced in map – reduce for these advanced task.
* Spark produce better results in ML, data mining Processes as it can perform better in the parallel and iterative environment.
* a. RDDs would be less suitable for applications that make asynchronous fine-grained updates to shared state, such as a storage system for a web application or an incremental web crawler
* RDDs performance is better as restrictions on RDDs have little impact in many parallel applications.
* RDDs offer an API based on coarse-grained transformations that lets them recover data efficiently using lineage.
* Spark system is tent to work better than Hadoop for the iterative application.