READ ME

SCHEMA

CATEGORIES:

[ASIN(STRING) : PID(INT) : CID(INT):CATNAME(STRING)]

SIMILAR :

[ASIN(STRING): PID(INT) :SIMILAR(STRING)]

PRODUCT:

[PID(INT) : ASIN(INT) : TITLE(STRING) : GROUP(STRING) : SALESRANK(INT)]

REVIEWS: [ASIN(STRING):TIME(STRING):PID(INT):USERID(STRING):RATING(INT):VOTES(INT):HELPFUL(INT)]

INPUT GIVEN THE FOLLOWING FORM :

AS A STRING QUERY IN FLASK API FORM

PREPROCESS amazon-meta.txt

{

TRANSFORM THE DATA TO DIFFERENT TABLES AND SAVES IT IN TXT FORMAT

}

PARSER

{

1.DRIVER PROGRAM PARSE THE SQL QUERY AND EXTRACTS INDIVIDUAL COMPONENTS OF SQL STATEMENTS

2.CALL WRAPPER FUNCTIONS FOR MAPREDUCE AND SPARK

3.CALCULATES TIME TAKEN BY MAPREDUCE AND SPARK AND SENDS OVERALL QUERY REPLY BY SENDING JSON OBJECTS

METHODS:

parse\_query():parse query into individual sql component and save it in mapperit.txt and run the following methods

{

run\_cmd():running hadoop streaming command ,running mapper.py reducer.py and giving total time taken to run

spark():running spark command and pyspark prog and also saving spark transformation operations saved in txt file and returning time taken

Cmd\_output():hadoop result output

spark\_out():spark output

spark()\_trans():return intermediate spark transformation as json object

heade(data,table):return intermediate mapper and reducer outputs

Finally returning the outputs of above methods and jsonify it and return it

}

}

MAP REDUCE LOGIC:

The GROUPBY mapper takes the comma separated values from the input file line by line through stdin. It also gets the parsed query object from the Driver. Based on the query it produces the key value pairs. Here the key is the SELECT columns (which may be one or more) and the value is the aggregate column. It pipes the key-value pair to the reducer.

The reducer takes the key-value pairs generated by the mapper and groups the pairs having the same keys and filters the data according to the having condition specified in the query. It outputs the data in the form of tab separated values.

SPARK LOGIC

The GROUPBY Spark mapper takes the values from the input file line by line through stdin. It also gets the parsed query object from the Driver. Based on the condition the spark mapper starts a spark job and read the file.

As per the "aggregate" function and the "where" condition the spark mapper generates the output data.