Assignment #1

Problem Statement

link

Query Format

GROUP BY TEMPLATE:

```
SELECT <COLUMNS>, FUNC(COLUMN1)
FROM <TABLE>
WHERE <COLUMN1> = Y
GROUP BY <COLUMNS>
HAVING FUNC(COLUMN1)>X
--Here FUNC can be COUNT, MAX, MIN, SUM
```

Input Data Format

```
Id:   0
ASIN: 0771044445
  discontinued product

Id:   1
ASIN: 0827229534
  title: Patterns of Preaching: A Sermon Sampler
  group: Book
  salesrank: 396585
  similar: 5  0804215715  156101074X  0687023955  0687074231  082721619X
  categories: 2
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Christianity[12290]|Clergy[12360]|Preaching[12368]
   |Books[283155]|Subjects[1000]|Religion & Spirituality[22]|Christianity[12290]|Clergy[12360]|Sermons[12370]
  reviews: total: 2  downloaded: 2  avg rating: 5
    2000-7-28  cutomer: A2JW67OY8U6HHK  rating: 5  votes:  10  helpful:   9
    2003-12-14  cutomer: A2VE83MZF98ITY  rating: 5  votes:   6  helpful:   5
```

Approach

- converted given unstructured data into 4 tables named `product`, `similar`, `category`, and `review`, using a script which reads given file line by line and using regular expressions parse it and creates csv files. Read script here
- Some entries from the tables

- Product

```
id,asin,title,group,salesrank,average_rating,downloaded
1,0827229534,Patterns of Preaching: A Sermon Sampler,Book,396585,5,2
2,0738700797,Candlemas: Feast of Flames,Book,168596,4.5,12
3,0486287785,World War II Allied Fighter Planes Trading Cards,Book,1270652,5,1
4,0842328327,Life Application Bible Commentary: 1 and 2 Timothy and Titus,Book,631289,4,1
5,1577943082,Prayers That Avail Much for Business: Executive,Book,455160,0,0
6,0486220125,How the Other Half Lives: Studies Among the Tenements of New York,Book,188784,4,17
7,B00000AU3R,Batik,Music,5392,4.5,3
8,0231118597,Losing Matt Shepard,Book,277409,4.5,15
9,1859677800,Making Bread: The Taste of Traditional Home-Baking,Book,949166,0,0
```

- Similar

```
product_id,similar_product_id
0827229534,0804215715
0827229534,156101074X
0827229534,0687023955
0827229534,0687074231
0827229534,082721619X
0738700797,0738700827
0738700797,1567184960
0738700797,1567182836
0738700797,0738700525
```

- Category

```
category_name,category_code
Books,283155
Subjects,1000
Religion & Spirituality,22
Christianity,12290
Clergy,12360
Preaching,12368
Sermons,12370
Earth-Based Religions,12472
Wicca,12484
```

    - Review

```
customer_id,product_id,rating,votes,helpful,date
A2JW67OY8U6HHK,0827229534,5,10,9,2000-7-28
A2VE83MZF98ITY,0827229534,5,6,5,2003-12-14
A11NCO6YTE4BTJ,0738700797,5,5,4,2001-12-16
A9CQ3PLRNIR83,0738700797,4,5,5,2002-1-7
A13SG9ACZ9O5IM,0738700797,5,8,8,2002-1-24
A1BDAI6VEYMAZA,0738700797,5,4,4,2002-1-28
A2P6KAWXJ16234,0738700797,4,16,16,2002-2-6
AMACWC3M7PQFR,0738700797,4,5,5,2002-2-14
A3GO7UV9XX14D8,0738700797,4,6,6,2002-3-23
```

- Then for hadoop program
    - SQL query is parsed and a json object is constructed out of it which is accessed by mapper and reducer
    - For query `select category_name, count(category_code) from category where category_name = 'Books' group by category_name having count(category_code) > 0`, JSON generated is

```
{'aggregate': {'field': 'category_code', 'function': 'count'},
 'columns': ['category_name'],
 'groupByColumns': ['category_name'],
 'having': {'column': 'category_code',
            'function': 'count',
            'operator': '>',
            'value': '0'},
 'table': 'category',
 'where': {'field': 'category_name', 'value': "'books'"}}
```

    - The mapper uses this JSON to identify which rows satisfy WHERE condition, and it outputs all rows satisfying WHERE conditions as it is with key as the concatenation of values of columns present in group by clause
    - Then reducer calculates aggregate condition and checks if that collection of rows satisfy that aggregate condition or not, if they do, it outputs all columns present in selct clause and aggregate column value, else it ignores it.

- For Spark program
    - the query is parsed and SparkSQL is used to run the query

- Sample outputs of both programs for SQL query

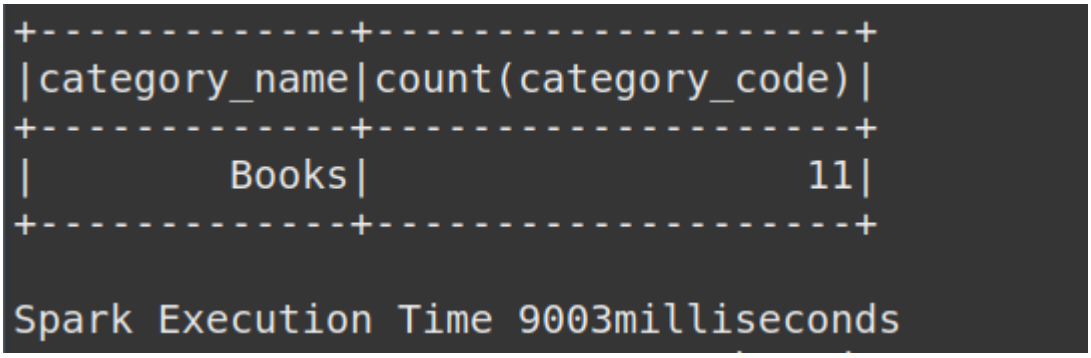
```
    select category_name, count(category_code) from category where
category_name = Books group by category_name having count(category_code) >
0
```

- Hadoop


```
Hadoop Execution Time 1283milliseconds
Mapper input -> <LongWritable, Text> (offset, line of file)
Mapper output -> <Text, Text> (columns in group by separated by _, row satisfying where condition)
Reducer input -> <Text, Text> (columns in group by separated by _, row)
Reducer output -> <Text, Text> (columns in group by separated by _, row containing required columns and satisfying having cor
```

```
    select category_name, count(category_code) from category where
category_name = 'Books' group by category_name having count(category_code)
> 0
```

- Spark



- Some Sample queries to try

```
select product_id, count(similar_product_id) from similar where product_id
= 1559362022 group by product_id having count(similar_product_id) > 0
select category_name, count(category_code) from category where
category_name = Books group by category_name having count(category_code) >
0
select asin, title, sum(downloaded) from product where asin = 0827229534
group by asin, title having sum(downloaded) > 0
select customer_id, max(rating) from review where customer_id =
A2JW67OY8U6HHK group by customer_id having max(rating) > 0
```

Instructions/Assumptions

- For Hadoop, do not enclose strings in `''`, but do it in spark
- Before running the job, delete `data/output` folder everytime