Bigdata Systems - Assignment 1 (S1-22\_SEZG522)

## Submitted by

* Noel John K - 2021MT93693
* Pavithra S – 2021MT93542
* Jayanthi Sangita M - 2021MT93337

# Dataset

Dataset source - [*https://www.kaggle.com/datasets/thedevastator/chemicals-in-cosmetics-what-s-really-in-your?resource=download*](https://www.kaggle.com/datasets/thedevastator/chemicals-in-cosmetics-what-s-really-in-your?resource=download)

This dataset is provided by Kaggle, and it contains ***114,297*** records of information on the chemicals used in cosmetics, including the name of the chemicals, the company that manufactures it, the primary category it is used in, and the date it was first reported.

The dataset contains following attributes:

***index,CDPHId,ProductName,CSFId,CSF,CompanyId,CompanyName,BrandName,PrimaryCategoryId,PrimaryCategory,SubCategoryId,SubCategory,CasId,CasNumber,ChemicalId,ChemicalName,InitialDateReported,MostRecentDateReported,DiscontinuedDate,ChemicalCreatedAt,ChemicalUpdatedAt,ChemicalDateRemoved,ChemicalCount***

# Assumptions

The following assumptions are made so that complex logics can be avoided.

* Values containing commas are neglected since the comma is considered as a delimiter.
* Instead of using ids as keys, we are taking name as the primary key. Because only few fields had ids.

# Hadoop Cluster

We were facing issues with BITS remote labs. So, we have spin up a Virtual machine in Azure and followed an article to setup a single node Hadoop Cluster. The reference articles are added to the reference section of this document.

# Execution

The MapReduce jobs can be executed in the cluster we have setup by executing the following command.

*hadoop jar libs/hadoop-streaming-3.3.4.jar -files mapper.py,reducer.py,chemicals-in-cosmetics-3.csv -mapper mapper.py -reducer reducer.py -input chemicals-in-cosmetics-3.csv -output output*

In order to test the application locally, the following command can be used.

*cat .\chemicals-in-cosmetics-3.csv | python .\mapper.py | python .\reducer.py*

# Analysis 1

## Analysis Performed

Finding the unique cosmetic products launched by a company - In this MapReduce program, we must find out the unique products launched by a company irrelevant of its brand name. This will help to identify how much cosmetic products are patented to each company.

## Input & Output Attributes

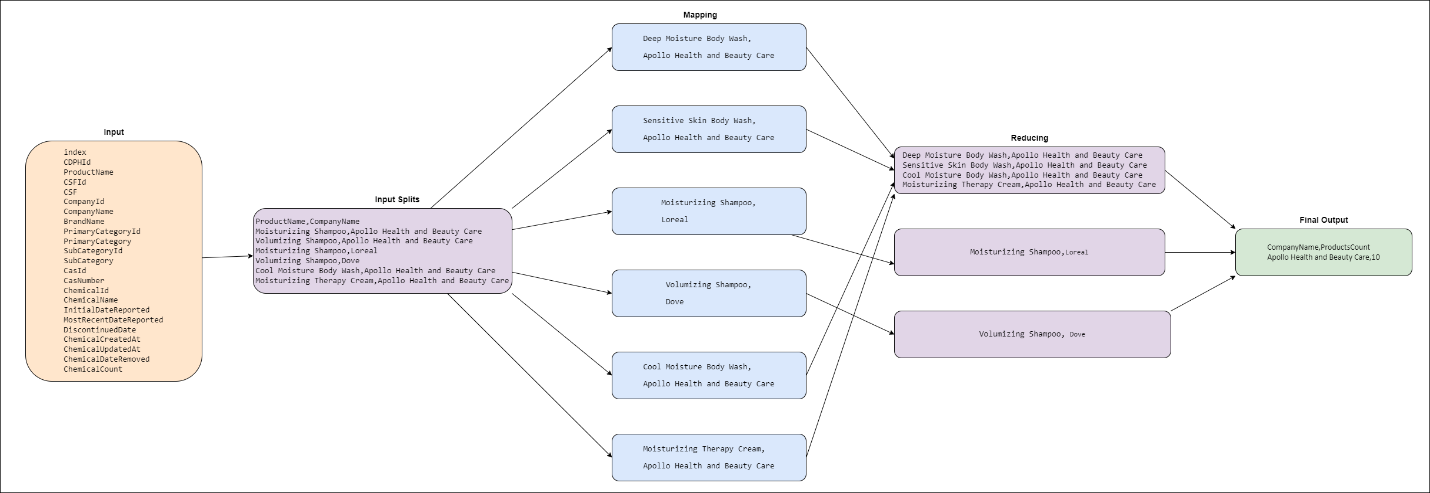
### Input Attributes

*CompanyName*, *ProductName*

### Output Attributes

*CompanyName*, *ProductNameCount*

## MapReduce Diagrams



## Mapper & Reducer Pseudo Codes

### Mapper Pseudo Code

*class Mapper:*

*method map(fullColumns):*

*columns = fullColumns.split(‘,’)*

*if(length(columns) == total\_colums)*

*companyName = columns[companyNamePosition]*

*productName = columns[productNamePosition]*

*write(selected columns)*

### Reducer Pseudo Code

*class Reducer:*

*method reduce(companyName, productName)*

*uniqueProducts = companyName productName #uniqueProducts is a set*

*key = companyName*

*value = count(companyName, productName)*

*dict = {key:value} #key is string, value is a counter where unique products are stored*

*if key not in dict:*

*dict.getValue(key).add(1)*

*else:*

*dict.getValue(key)+1*

*write(dict.key, dict.value)*

## Mapper & Reducer Programs

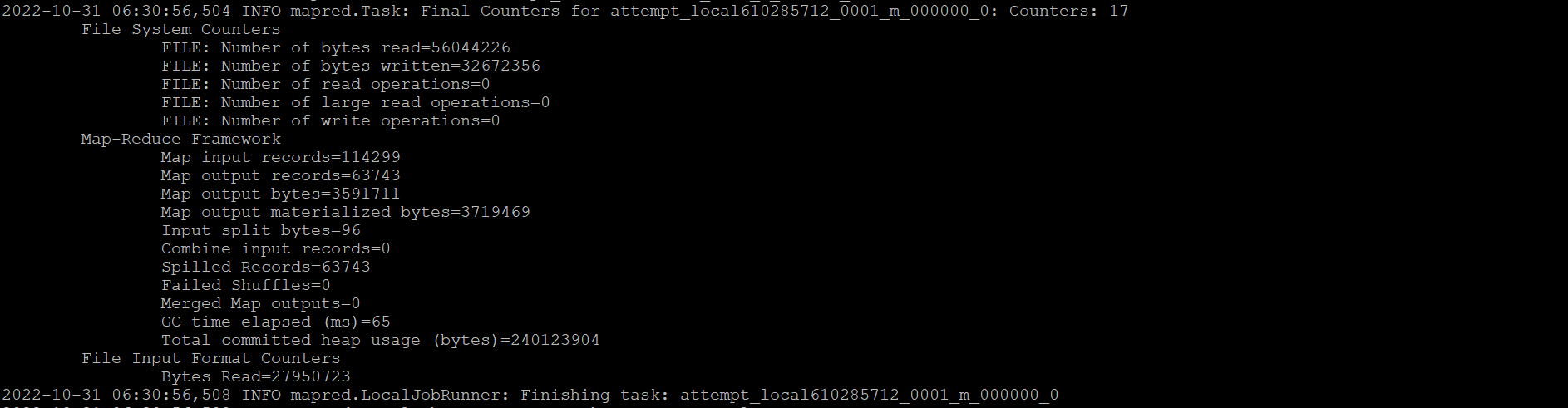
### Mapper Program

*#!/usr/bin/env python3  
import sys  
  
delimiter = ","  
  
def map():  
 for line in sys.stdin:  
 rows = line.strip()  
 columns = rows.split(delimiter)  
  
 if len(columns) == 23:  
 product\_name = columns[2]  
 company\_name = columns[6]  
  
 print(f"{company\_name}{delimiter}{product\_name}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 map()*

### Reducer Program

*#!/usr/bin/env python3  
import sys  
  
delimiter = ","  
total\_products = set()  
unique\_products = {}  
  
  
def reduce():  
 my\_iterator = iter(sys.stdin.readline, "")  
 header = next(my\_iterator)  
 company\_name\_header, product\_name\_header = header.strip().split(delimiter)  
 print(f"{company\_name\_header}{delimiter}ProductsCount")  
  
 for line in sys.stdin:  
 line = line.strip()  
  
 company\_name, product\_name = line.split(delimiter)  
 key = f"{company\_name}{delimiter}{product\_name}"  
 total\_products.add(key)  
  
 for key in total\_products:  
 company\_name, product\_name = key.split(delimiter)  
  
 if company\_name in unique\_products.keys():  
 count = unique\_products[company\_name]  
 unique\_products[company\_name] = count + 1  
 else:  
 unique\_products[company\_name] = 1  
  
 for company\_name in unique\_products.keys():  
 print(f"{company\_name}{delimiter}{unique\_products[company\_name]}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 reduce()*

## Statistics



Text

Description automatically generated

Text

Description automatically generated

## Sample Input & Output data’s

### Input data

index,CDPHId,ProductName,CSFId,CSF,CompanyId,CompanyName,BrandName,PrimaryCategoryId,PrimaryCategory,SubCategoryId,SubCategory,CasId,CasNumber,ChemicalId,ChemicalName,InitialDateReported,MostRecentDateReported,DiscontinuedDate,ChemicalCreatedAt,ChemicalUpdatedAt,ChemicalDateRemoved,ChemicalCount  
33518,11448,Deep Moisture Body Wash,,,475,Apollo Health and Beauty Care,Equate,6,Bath Products,159,Body Washes and Soaps,656,13463-67-7,16729,Titanium dioxide,05/20/2010,07-01-10,,05/20/2010,05/20/2010,,1  
40705,14452,COOL MOISTURE BODY WASH,,,475,Apollo Health and Beauty Care,Equate,6,Bath Products,159,Body Washes and Soaps,656,13463-67-7,22125,Titanium dioxide,07-01-10,07-01-10,,07-01-10,07-01-10,,1  
40708,14454,Sensitive Skin Body Wash,,,475,Apollo Health and Beauty Care,Equate,6,Bath Products,159,Body Washes and Soaps,656,13463-67-7,22127,Titanium dioxide,07-01-10,07-01-10,,07-01-10,07-01-10,,1  
40714,14458,Moisturizing Therapy Cream,,,475,Apollo Health and Beauty Care,Natural Concepts,90,Skin Care Products ,102,Skin Moisturizers (making a cosmetic claim),656,13463-67-7,22131,Titanium dioxide,07-01-10,07-01-10,,07-01-10,07-01-10,,1  
41452,14656,Tropical Renewal Softening Body Wash,,,475,Apollo Health and Beauty Care,Equate,6,Bath Products,159,Body Washes and Soaps,656,13463-67-7,22408,Titanium dioxide,07-12-10,07-12-10,,07-12-10,07-12-10,,1  
43022,15133,Frizz Release Hold Gel,,,475,Apollo Health and Beauty Care,Natural Concepts,18,Hair Care Products (non-coloring),26,Hair Styling Products,656,13463-67-7,23224,Titanium dioxide,08/20/2010,08/20/2010,,08/20/2010,08/20/2010,,1  
48508,16825,Cool Moisture Body Wash,,,475,Apollo Health and Beauty Care,IMAGE ESSENTIALS,6,Bath Products,159,Body Washes and Soaps,656,13463-67-7,26185,Titanium dioxide,07-05-11,07-05-11,,07-05-11,07-05-11,,1  
48509,16826,Deep Moisture Boday Wash,,,475,Apollo Health and Beauty Care,IMAGE ESSENTIALS,6,Bath Products,159,Body Washes and Soaps,656,13463-67-7,26186,Titanium dioxide,07-05-11,07-05-11,,07-05-11,07-05-11,,1  
48677,16912,Moisturizing Shampoo,,,475,Apollo Health and Beauty Care,Rusk,18,Hair Care Products (non-coloring),25,Hair Shampoos (making a cosmetic claim) ,656,13463-67-7,26306,Titanium dioxide,08-05-11,08-05-11,,08-05-11,08-05-11,,1  
48678,16915,Volumizing Shampoo,,,475,Apollo Health and Beauty Care,Rusk,18,Hair Care Products (non-coloring),25,Hair Shampoos (making a cosmetic claim) ,656,13463-67-7,26308,Titanium dioxide,08-05-11,08-05-11,,08-05-11,08-05-11,,1

### Output data

CompanyName,ProductsCount  
Apollo Health and Beauty Care,10

# Analysis 2

## Analysis Performed

Finding all the chemicals associated with a product launched by a company under one brand name – Here the output of the MapReduce program will give us the chemicals used to manufacture the cosmetic product launched by a company under a brand name.

## Input & Output Attributes

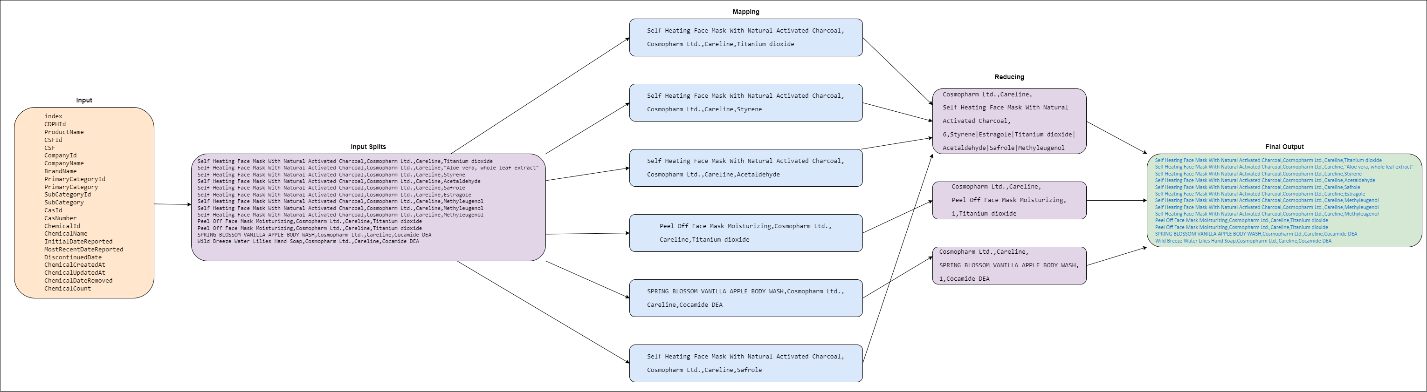
### Input attribute

*CompanyName*, *BrandName*, *ProductName*, *ChemicalName*

### Output Attribute

*CompanyName*, *BrandName*, *ProductName*, *ChemicalCount*, *ChemicalName*

## MapReduce Diagrams



## Mapper & Reducer Pseudo Codes

### Mapper Pseudo Code

*class Mapper:*

*method map(fullColumns):*

*columns = fullColumns.split(‘,’)*

*if(length(columns) == total\_colums)*

*companyName = columns[companyNamePosition]*

*brandName = columns[brandNamePosition]*

*productName = columns[productNamePosition]*

*chemicalName = columns[chemicalNamePosition]*

*write(selected columns)*

### Reducer Pseudo Code

*class Reducer:*

*method reduce(companyName, brandName, productName, chemicalName)*

*key = companyName, brandName, productName*

*value = chemicalName*

*dict = {key:value} # key is string, value is a set*

*if value not in dict:*

*dict.getValue().add(value)*

*write(dict.key, dict.value)*

## Mapper & Reducer Programs

### Mapper Program

*#!/usr/bin/env python3  
import sys  
  
delimiter = ","  
  
  
def map():  
 for line in sys.stdin:  
 rows = line.strip()  
 columns = rows.split(delimiter)  
  
 if len(columns) == 23:  
 company\_name = columns[6]  
 brand\_name = columns[7]  
 product\_name = columns[2]  
 chemical\_name = columns[15]  
  
 print(f"{company\_name}{delimiter}{brand\_name}{delimiter}{product\_name}{delimiter}{chemical\_name}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 map()*

### Reducer Program

*#!/usr/bin/env python3  
import sys  
  
delimiter = ","  
content\_separator = "|"  
unique\_products = {}  
  
  
def reduce():  
 my\_iterator = iter(sys.stdin.readline, "")  
 header = next(my\_iterator)  
 company\_name\_header, brand\_name\_header, product\_name\_header, chemical\_name\_header = header.strip().split(delimiter)  
 print(f"{company\_name\_header}{delimiter}{brand\_name\_header}{delimiter}{product\_name\_header}{delimiter}ChemicalCount{delimiter}{chemical\_name\_header}")  
  
 for line in sys.stdin:  
 line = line.strip()  
  
 company\_name, brand\_name, product\_name, chemical\_name = line.split(delimiter)  
 key = f"{company\_name}{delimiter}{brand\_name}{delimiter}{product\_name}"  
  
 if key in unique\_products.keys():  
 unique\_chemicals = unique\_products[key]  
 unique\_chemicals.add(chemical\_name)  
 else:  
 unique\_chemicals = set()  
 unique\_chemicals.add(chemical\_name)  
 unique\_products[key] = unique\_chemicals  
  
 for key in unique\_products.keys():  
 print(f"{key}{delimiter}{len(unique\_products[key])}{delimiter}{content\_separator.join(unique\_products[key])}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 reduce()*

## Statistics

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## Sample Input and Output data’s

### Input data

index,CDPHId,ProductName,CSFId,CSF,CompanyId,CompanyName,BrandName,PrimaryCategoryId,PrimaryCategory,SubCategoryId,SubCategory,CasId,CasNumber,ChemicalId,ChemicalName,InitialDateReported,MostRecentDateReported,DiscontinuedDate,ChemicalCreatedAt,ChemicalUpdatedAt,ChemicalDateRemoved,ChemicalCount  
113768,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,656,13463-67-7,67675,Titanium dioxide,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,03/20/2020,6  
113769,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,1108,,67676,*"*Aloe vera, whole leaf extract*"*,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,6  
113770,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,620,100-42-5,67677,Styrene,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,6  
113771,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,2,75-07-0,67678,Acetaldehyde,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,6  
113772,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,608,94-59-7,67679,Safrole,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,6  
113773,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,293,140-67-0,67680,Estragole,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,6  
113774,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,442,93-15-2,67681,Methyleugenol,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,03/20/2020,6  
113775,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,442,93-15-2,67682,Methyleugenol,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,6  
113776,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,442,93-15-2,67683,Methyleugenol,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,03/20/2020,6  
113777,41309,Peel Off Face Mask Moisturizing,,,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,95,Facial Masks,656,13463-67-7,67684,Titanium dioxide,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,1  
113778,41309,Peel Off Face Mask Moisturizing,,,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,102,Skin Moisturizers (making a cosmetic claim),656,13463-67-7,67684,Titanium dioxide,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,,1  
114295,41449,SPRING BLOSSOM VANILLA APPLE BODY WASH,,,1388,Cosmopharm Ltd.,Careline,6,Bath Products,159,Body Washes and Soaps,969,,67905,Cocamide DEA,04/30/2020,04/30/2020,,04/30/2020,04/30/2020,,1  
114296,41450,Wild Breeze Water Lilies Hand Soap,,,1388,Cosmopharm Ltd.,Careline,6,Bath Products,159,Body Washes and Soaps,969,,67906,Cocamide DEA,04/30/2020,04/30/2020,,04/30/2020,04/30/2020,,1

### Output data

CompanyName,BrandName,ProductName,ChemicalCount,ChemicalName  
Cosmopharm Ltd.,Careline,Self Heating Face Mask With Natural Activated Charcoal,6,Styrene|Estragole|Titanium dioxide|Acetaldehyde|Safrole|Methyleugenol  
Cosmopharm Ltd.,Careline,Peel Off Face Mask Moisturizing,1,Titanium dioxide   
Cosmopharm Ltd.,Careline,SPRING BLOSSOM VANILLA APPLE BODY WASH,1,Cocamide DEA   
Cosmopharm Ltd.,Careline,Wild Breeze Water Lilies Hand Soap,1,Cocamide DEA

# Analysis 3

## Analysis Performed

Finding primary category of cosmetics which has highest discontinued chemicals – In this MapReduce problem, the output will give us the primary category which contains the highest no. of chemicals that are discontinued.

## Input & Output Attributers

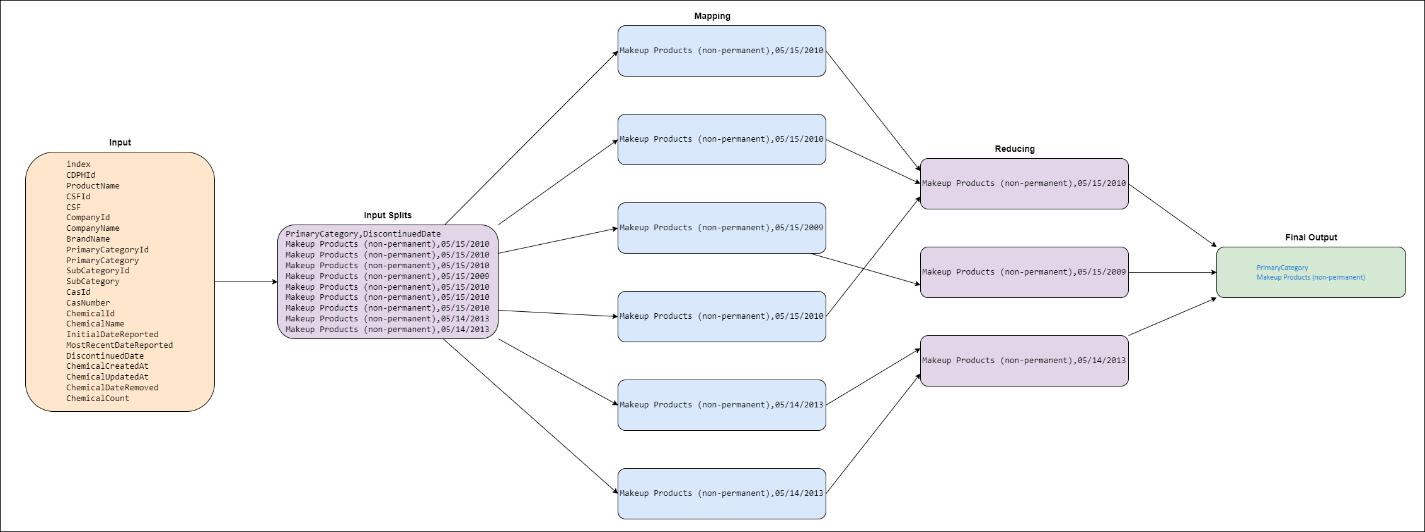
### Input Attribute

*PrimaryCategory*, *ChemicalDateRemoved*

### Output Attribute

*PrimaryCategory*

## MapReduce Diagrams



## Mapper & Reducer Pseudo Codes

### Mapper Pseudo Code

*class Mapper:*

*method map(fullColumns):*

*columns = fullColumns.split(‘,’)*

*if(length(columns) == total\_colums)*

*primaryCategory = columns[primaryCategoryPosition]*

*chemicalDateRemoved = columns[chemicalDateRemovedPosition]*

*write(selected columns)*

### Reducer Pseudo Code

*class Reducer:*

*method reduce(primaryCategory, chemicalDateRemoved)*

*key = primaryCategory*

*value = chemicalName*

*dict = {key:value} # key is string, value is a counter*

*if key not in dict:*

*dict.getValue(key).add(1)*

*else:*

*dict.getValue(key)+1*

*write(dict.key, dict.value)*

## Mapper & Reducer Programs

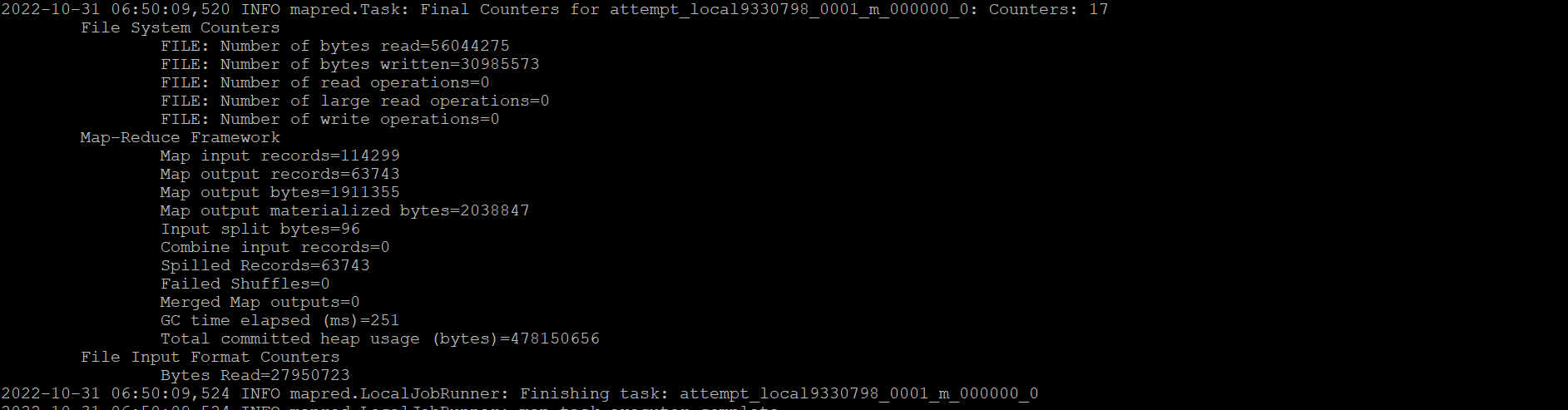
### Mapper Program

*#!/usr/bin/env python3  
import sys  
  
delimiter = ","  
  
def map():  
 for line in sys.stdin:  
 rows = line.strip()  
 columns = rows.split(delimiter)  
  
 if len(columns) == 23:  
 primary\_category = columns[9]  
 discontinued\_date = columns[18]  
  
 print(f"{primary\_category}{delimiter}{discontinued\_date}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 map()*

### Reducer Program

*#!/usr/bin/env python3  
import sys  
import collections  
  
delimiter = ","  
primary\_category\_with\_discontinued\_chemicals = collections.Counter()  
  
def reduce():  
 my\_iterator = iter(sys.stdin.readline, "")  
 header = next(my\_iterator)  
 primary\_category\_header, discontinued\_date\_header = header.strip().split(delimiter)  
 print(f"{primary\_category\_header}")  
  
 for line in sys.stdin:  
 line = line.strip()  
  
 primary\_category, discontinued\_date = line.split(delimiter)  
  
 if discontinued\_date is not None and discontinued\_date!= "":  
 if primary\_category in primary\_category\_with\_discontinued\_chemicals.keys():  
 count = primary\_category\_with\_discontinued\_chemicals[primary\_category]  
 primary\_category\_with\_discontinued\_chemicals[primary\_category] = count + 1  
 else:  
 primary\_category\_with\_discontinued\_chemicals[primary\_category] = 1  
  
 print(primary\_category\_with\_discontinued\_chemicals.most\_common(1)[0][0])  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 reduce()*

## Statistics



Text

Description automatically generated

Text

Description automatically generated

## Sample Input & Output data’s

### Input data

index,CDPHId,ProductName,CSFId,CSF,CompanyId,CompanyName,BrandName,PrimaryCategoryId,PrimaryCategory,SubCategoryId,SubCategory,CasId,CasNumber,ChemicalId,ChemicalName,InitialDateReported,MostRecentDateReported,DiscontinuedDate,ChemicalCreatedAt,ChemicalUpdatedAt,ChemicalDateRemoved,ChemicalCount  
416,254,COLOR TREND LIQUID EYE LINER BRIGHTS-ALL SHADES �,,,4,New Avon LLC,AVON,44,Makeup Products (non-permanent),46,Eyeliner/Eyebrow Pencils,656,13463-67-7,265,Titanium dioxide,09-01-09,08/28/2013,05/15/2010,09-01-09,09-01-09,,1  
4572,1359,AVON SHIMMER SWIRLS FACE ILLUMINATOR-ALL SHADES �,,,4,New Avon LLC,AVON,44,Makeup Products (non-permanent),49,Face Powders,656,13463-67-7,1488,Titanium dioxide,09/21/2009,08/28/2013,05/15/2010,09/21/2009,09/21/2009,,1  
4733,1439,AVON 8-IN-1! LIP PALETTE-ALL SHADES �,,,4,New Avon LLC,AVON,44,Makeup Products (non-permanent),52,Lip Gloss/Shine,656,13463-67-7,1576,Titanium dioxide,09/22/2009,08/28/2013,05/15/2010,09/22/2009,09/22/2009,,1  
19762,4928,MARK C-THRU-U BEAUTIFYING SHEER TINT-ALL SHADES,,,4,New Avon LLC,MARK,44,Makeup Products (non-permanent),50,Foundations and Bases,656,13463-67-7,8639,Titanium dioxide,10/14/2009,09-04-13,05/15/2009,10/14/2009,10/14/2009,,1  
19773,4939,MARK LIP GLOSS TRIANGLES-ALL SHADES,,,4,New Avon LLC,MARK,44,Makeup Products (non-permanent),52,Lip Gloss/Shine,656,13463-67-7,8654,Titanium dioxide,10/14/2009,09-04-13,05/15/2010,10/14/2009,10/14/2009,,1  
19925,4998,MARK I-SHEER CREAMY EYE SHADOW HOOK UP-ALL SHADES �,,,4,New Avon LLC,MARK,44,Makeup Products (non-permanent),48,Eye Shadow,656,13463-67-7,8760,Titanium dioxide,10/15/2009,09-05-13,05/15/2010,10/15/2009,10/15/2009,,1  
21609,5790,MARK JUICE GEMS LIP GLOSS �(SOLD IN KIT 'JUIC GEMS MINI GIFT SET),,,4,New Avon LLC,AVON,44,Makeup Products (non-permanent),53,*"*Lip Color - Lipsticks, Liners, and Pencils*"*,656,13463-67-7,9693,Titanium dioxide,10/16/2009,10-02-13,05/15/2010,10/16/2009,10/16/2009,,1  
25252,7073,Rouge glossy lipstick,,,301,Yves Rocher Inc.,Luminelle,44,Makeup Products (non-permanent),53,*"*Lip Color - Lipsticks, Liners, and Pencils*"*,656,13463-67-7,11085,Titanium dioxide,11/19/2009,11-08-13,05/14/2013,11/19/2009,11/19/2009,11/19/2009,1  
25253,7073,Rouge glossy lipstick,,,301,Yves Rocher Inc.,Luminelle,44,Makeup Products (non-permanent),53,*"*Lip Color - Lipsticks, Liners, and Pencils*"*,656,13463-67-7,11088,Titanium dioxide,11/19/2009,11-08-13,05/14/2013,11/19/2009,11/19/2009,,1

### Output data

PrimaryCategory  
Makeup Products (non-permanent)

# Analysis 4

## Analysis Performed

Finding latest 5 removed chemicals in the cosmetics products – In this MapReduce problem, the output will give us the chemicals that are removed to manufacture the cosmetic products.

## Inputs & Output Attributes

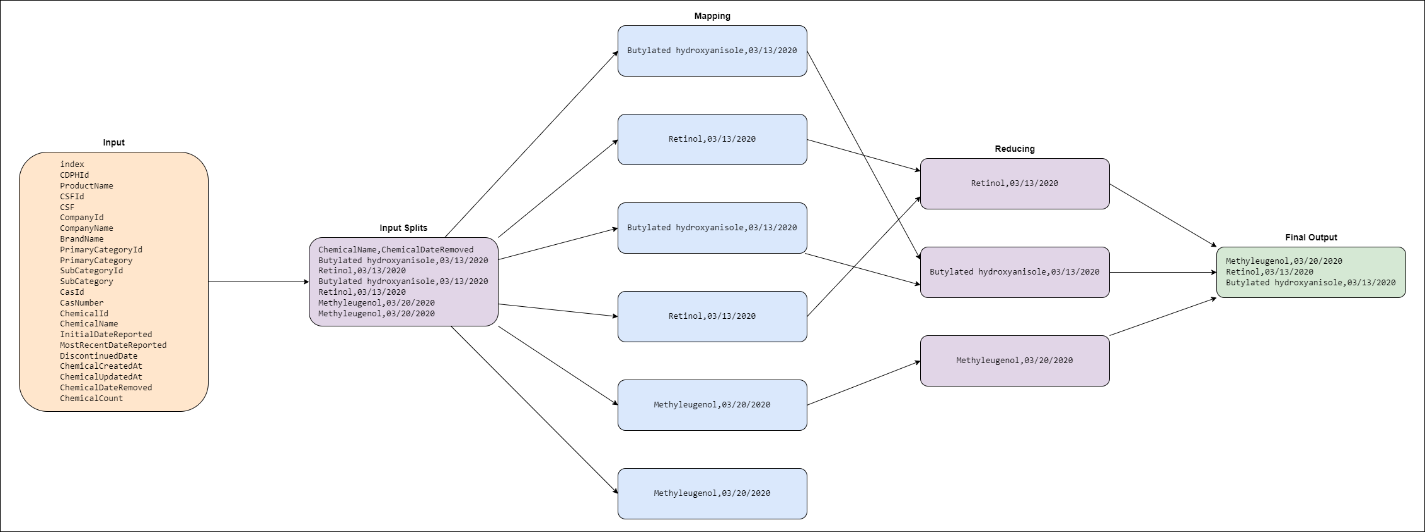
### Input Attribute

*ChemicalName*, *ChemicalDateRemoved*

### Outputs Attribute

*ChemicalName*, *ChemicalDateRemoved*

## MapReduce Diagrams



## Mapper & Reducer Pseudo Codes

### Mapper Pseudo Code

*class Mapper:*

*method map(fullColumns):*

*columns = fullColumns.split(‘,’)*

*if(length(columns) == total\_colums)*

*chemicalName = columns[chemicalNamePosition]*

*chemicalDateRemoved = columns[chemicalDateRemovedPosition]*

*write(selected columns)*

### Reducer Pseudo Code

*class Reducer:*

*method reduce(chemicalName, chemicalDateRemoved)*

*key = chemicalName*

*value = chemicalDateRemoved*

*dict = {key:value} # key is string, value is a date field*

*if key not in dict:*

*if value > dict.get(key)*

*dict.getValue(key)= value*

*else:*

*dict.getValue(key)= value*

*write(dict.key, dict.value)*

## Mapper & Reducer Programs

### Mapper Program

*#!/usr/bin/env python3  
import sys  
  
delimiter = ","  
  
  
def map():  
 for line in sys.stdin:  
 rows = line.strip()  
 columns = rows.split(delimiter)  
  
 if len(columns) == 23:  
 chemical\_name = columns[15]  
 chemical\_date\_removed = columns[21]  
  
 print(f"{chemical\_name}{delimiter}{chemical\_date\_removed}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 map()*

### Reducer Program

*#!/usr/bin/env python3  
import sys  
from datetime import datetime  
  
delimiter = ","  
chemicals\_removed = {}  
  
  
def reduce():  
 my\_iterator = iter(sys.stdin.readline, "")  
 header = next(my\_iterator)  
 chemical\_name\_header, chemical\_date\_removed\_header = header.strip().split(delimiter)  
 print(f"{chemical\_name\_header}{delimiter}{chemical\_date\_removed\_header}")  
  
 for line in sys.stdin:  
 line = line.strip()  
  
 chemical\_name, chemical\_date\_removed = line.split(delimiter)  
  
 if chemical\_date\_removed is not None and "/" in chemical\_date\_removed:  
 chemical\_date\_removed = datetime.strptime(chemical\_date\_removed, "%m/%d/%Y").strftime("%Y/%m/%d")  
  
 if chemical\_name in chemicals\_removed.keys():  
 if chemical\_date\_removed > chemicals\_removed[chemical\_name]:  
 chemicals\_removed[chemical\_name] = chemical\_date\_removed  
 else:  
 chemicals\_removed[chemical\_name] = chemical\_date\_removed  
  
 sorted\_chemicals\_removed = sorted(chemicals\_removed.items(), key=lambda kv: (kv[1], kv[0]), reverse=True)[0:5]  
  
 for chemicals in sorted\_chemicals\_removed:  
 print(f"{chemicals[0]}{delimiter}{chemicals[1]}")  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 reduce()*

## Statistics

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## Sample Input & Output data’s

### Input data

index,CDPHId,ProductName,CSFId,CSF,CompanyId,CompanyName,BrandName,PrimaryCategoryId,PrimaryCategory,SubCategoryId,SubCategory,CasId,CasNumber,ChemicalId,ChemicalName,InitialDateReported,MostRecentDateReported,DiscontinuedDate,ChemicalCreatedAt,ChemicalUpdatedAt,ChemicalDateRemoved,ChemicalCount  
103327,37147,A-Zyme Peel,,,1316,Ultraceuticals Pty Ltd,Ultraceuticals,90,Skin Care Products ,105,Other Skin Care Product,92,25013-16-5,61078,Butylated hydroxyanisole,04/23/2019,03-12-20,,04/23/2019,03-12-20,03/13/2020,2  
103328,37147,A-Zyme Peel,,,1316,Ultraceuticals Pty Ltd,Ultraceuticals,90,Skin Care Products ,105,Other Skin Care Product,958,68-26-8,67601,Retinol,04/23/2019,03-12-20,,03-12-20,03-12-20,03/13/2020,2  
103329,37147,A-Zyme Peel,,,1316,Ultraceuticals Pty Ltd,Ultraceuticals,90,Skin Care Products ,105,Other Skin Care Product,92,25013-16-5,67602,Butylated hydroxyanisole,04/23/2019,03-12-20,,03-12-20,03-12-20,03/13/2020,2  
113550,41264,Ultra A Skin Perfecting Serum Mild,,,1316,Ultraceuticals Pty Ltd,Ultraceuticals,90,Skin Care Products ,92,Anti-Wrinkle/Anti-Aging Products (making a cosmetic claim),958,68-26-8,67604,Retinol,03-12-20,03-12-20,,03-12-20,03-12-20,03/13/2020,2  
113774,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,442,93-15-2,67681,Methyleugenol,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,03/20/2020,6  
113776,41308,Self Heating Face Mask With Natural Activated Charcoal,64642,Fragrance,1388,Cosmopharm Ltd.,Careline,90,Skin Care Products ,93,Skin Cleansers,442,93-15-2,67683,Methyleugenol,03/20/2020,03/20/2020,,03/20/2020,03/20/2020,03/20/2020,6

### Output data

ChemicalName,ChemicalDateRemoved  
Methyleugenol,2020/03/20  
Retinol,2020/03/13  
Butylated hydroxyanisole,2020/03/13

# References

* [Chemicals in Cosmetics](https://www.kaggle.com/datasets/thedevastator/chemicals-in-cosmetics-what-s-really-in-your?resource=download)
* [Running Hadoop on Ubuntu Linux (Single-Node Cluster)](https://www.michael-noll.com/tutorials/running-hadoop-on-ubuntu-linux-single-node-cluster/)
* [Setting up a Single Node Cluster](https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html)
* [Writing a Hadoop MapReduce Program in Python](https://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/)
* [Source code](https://github.com/noeljohnk007/datasets/tree/main/mapreduce)