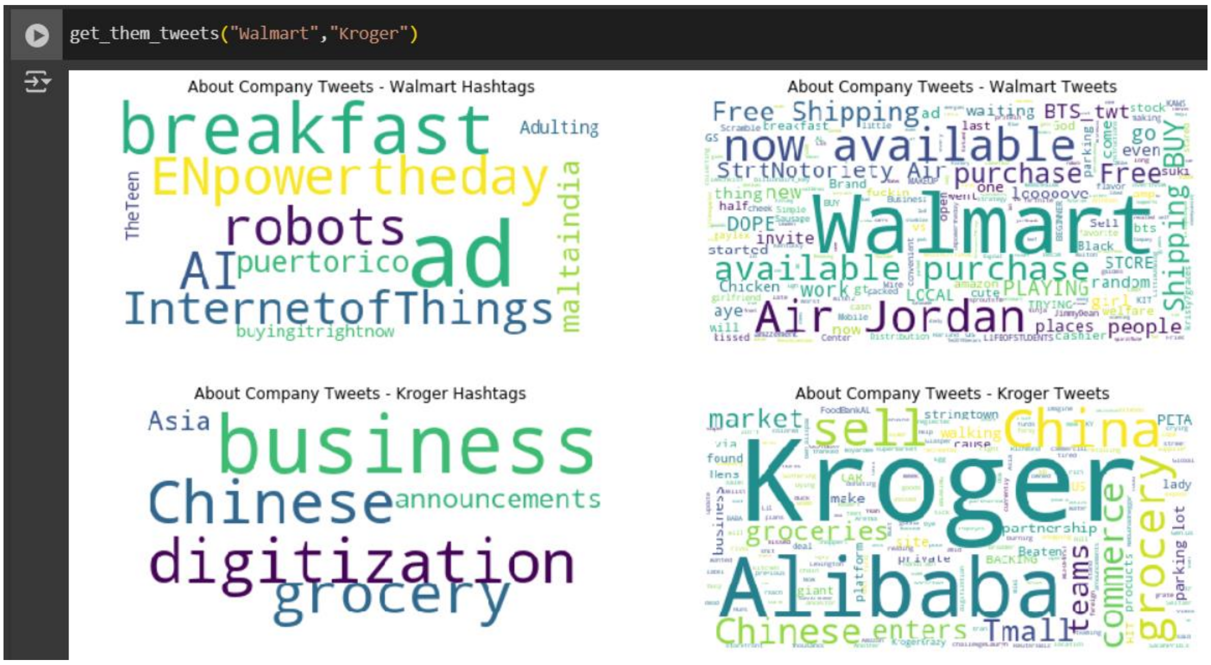
EXPERIMENT NO – 09

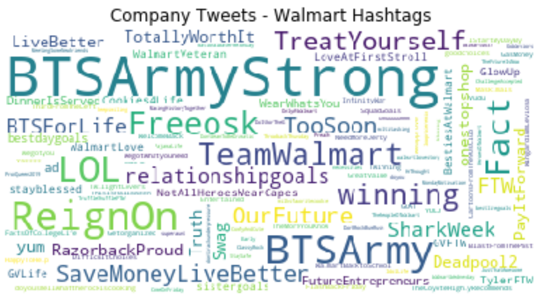
CODE :

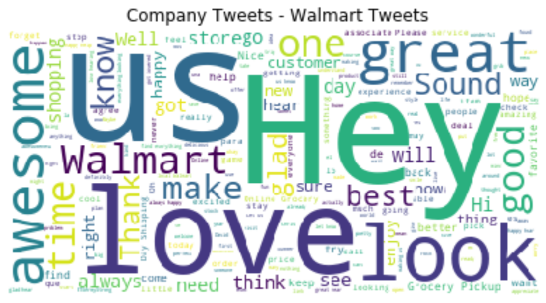
```
 import tweepy
import pandas as pd
import matplotlib.pyplot as plt
from textblob import TextBlob
from wordcloud import WordCloud, STOPWORDS
import datetime

def authenticate_twitter():
    auth = tweepy.OAuthHandler('cust_token', 'consumer_secret')
    auth.set_access_token('access_token', 'access_token_secret')
    return tweepy.API(auth, wait_on_rate_limit=True)

def fetch_tweets(api, company, route):
    results = []
    if route == 1:
        results = [tweet for tweet in tweepy.Cursor(api.search, q=company,
lang="en").items(101)]
        title = "About Company Tweets - "
    else:
        results = [tweet for page in tweepy.Cursor(api.user_timeline, id=company,
count=101).pages() for tweet in page]
        title = "Company Tweets - "
    return results, title

def process_tweets(results):
    data = pd.DataFrame({
        'id': [t.id for t in results],
        'text': [t.text.split('https:')[0] for t in results],
        'created_at': [t.created_at for t in results],
        'retweet_count': [t.retweet_count for t in results],
        'user_followers_count': [t.author.followers_count for t in results],
        'user_location': [t.author.location for t in results],
        'hashtags': [t.entities.get('hashtags') for t in results]
    })

    data.drop_duplicates('text', inplace=True)
    data['Sentiment'] = data['text'].apply(lambda x: TextBlob(x).sentiment.polarity)
    data['SentimentClass'] = pd.cut(data['Sentiment'], [-float('inf'), 0, 0.01, float('inf')],
                                labels=['Negative', 'Neutral', 'Positive'])
    return data

def generate_wordclouds(data, hashtags, company, title):
    plt.figure(figsize=[15,15])
    wc = WordCloud(background_color="white", stopwords=STOPWORDS)

    plt.subplot(221)
    plt.title(f"{title}{company} Hashtags")
    ht_text = " ".join(h['text'] for hashtag in hashtags for h in hashtag if h['text'].lower() !=
'fuck')
    wc.generate(ht_text)
    plt.imshow(wc)
    plt.axis("off")

    plt.subplot(222)
    plt.title(f"{title}{company} Tweets")
    tweet_text = " ".join(data['text'].str.replace('RT', ''))
    wc.generate(tweet_text)
    plt.imshow(wc)
    plt.axis("off")
    plt.show()

def plot_sentiment_analysis(cmp1_data, cmp2_data, cmp1_id, cmp2_id):
    for cmp_data, cmp_id in [(cmp1_data, cmp1_id), (cmp2_data, cmp2_id)]:
        best = cmp_data.loc[cmp_data['Sentiment'].idxmax()]
        worst = cmp_data.loc[cmp_data['Sentiment'].idxmin()]
        print(f"\n{cmp_id} Tweets\nBest: {best['text']}\nWorst: {worst['text']}")

    plt.figure(figsize=[15,6])
    sentiment_pct = pd.DataFrame({
        cmp1_id: cmp1_data['SentimentClass'].value_counts(normalize=True),
        cmp2_id: cmp2_data['SentimentClass'].value_counts(normalize=True)
    }).reindex(['Negative', 'Neutral', 'Positive']).fillna(0)
    sentiment_pct.plot.bar()
    plt.show()

def plot_company_metrics(cmp1_data, cmp2_data, cmp1_id, cmp2_id):
    plt.figure(figsize=[10,15])

    metrics = [
        ('user_followers_count', 'max', "Number of Followers"),
        ('Sentiment', 'mean', "Average Sentiment"),
        ('retweet_count', 'mean', "Average Retweets")
    ]

    for i, (col, agg, title) in enumerate(metrics, 221):
        plt.subplot(i)
        plt.bar([cmp1_id, cmp2_id], [cmp1_data[col].agg(agg), cmp2_data[col].agg(agg)])
        plt.title(f"Comparison of {title}")

    plt.subplot(224)
    for data in [cmp1_data, cmp2_data]:
        data['created_at'] = pd.to_datetime(data['created_at']).dt.normalize()
    plt.bar([cmp1_id, cmp2_id],
            [d.groupby('created_at').size().mean() for d in [cmp1_data, cmp2_data]])
    plt.title("Comparison of Number of Tweets")
    plt.show()

def analyze_companies(company, rival):
    api = authenticate_twitter()

    # Fetch and process tweets
    cmp_data, _ = fetch_tweets(api, company, 1)
    cmp_data = process_tweets(cmp_data)
    cmp_own_data, _ = fetch_tweets(api, company, 2)
    cmp_own_data = process_tweets(cmp_own_data)

    rival_data, _ = fetch_tweets(api, rival, 1)
    rival_data = process_tweets(rival_data)
    rival_own_data, _ = fetch_tweets(api, rival, 2)
    rival_own_data = process_tweets(rival_own_data)

    # Generate visualizations
    generate_wordclouds(cmp_data, cmp_data['hashtags'], company, "About Company Tweets
- ")
    generate_wordclouds(rival_data, rival_data['hashtags'], rival, "About Company Tweets - ")
    plot_sentiment_analysis(cmp_data, rival_data, company, rival)
    plot_company_metrics(cmp_own_data, rival_own_data, company, rival)

# Usage: analyze_companies("company1", "company2")
```

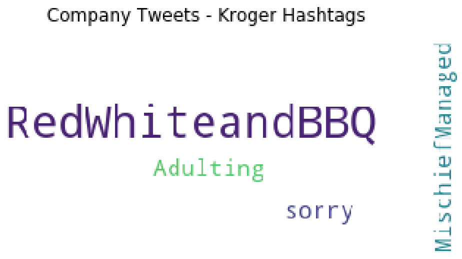
OUTPUT

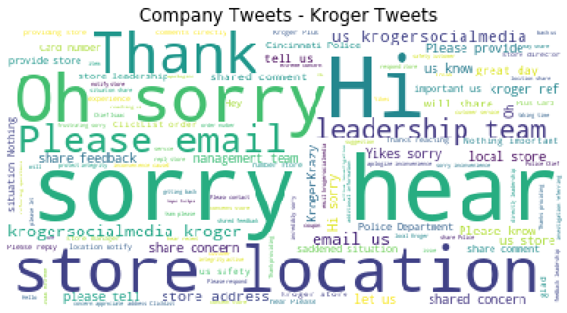


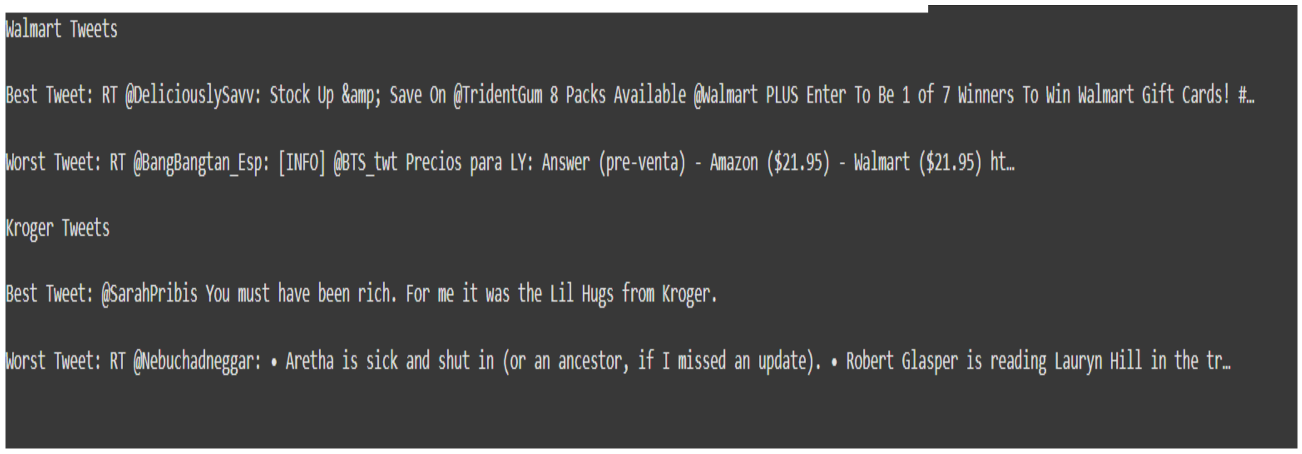
```
get_them_tweets("Walmart","Kroger")
```

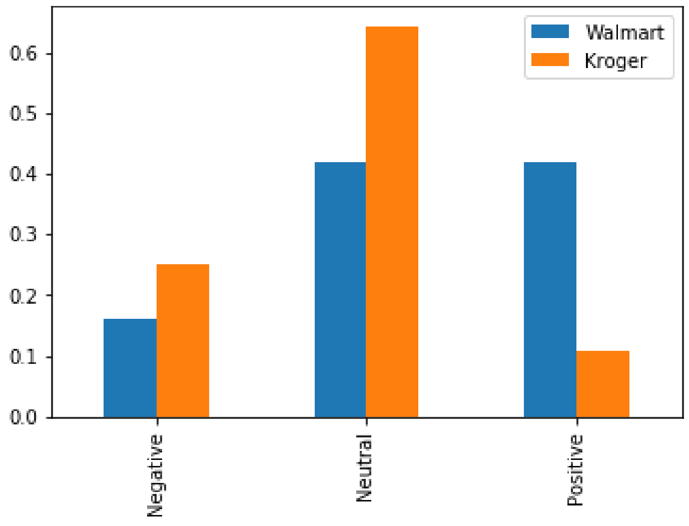


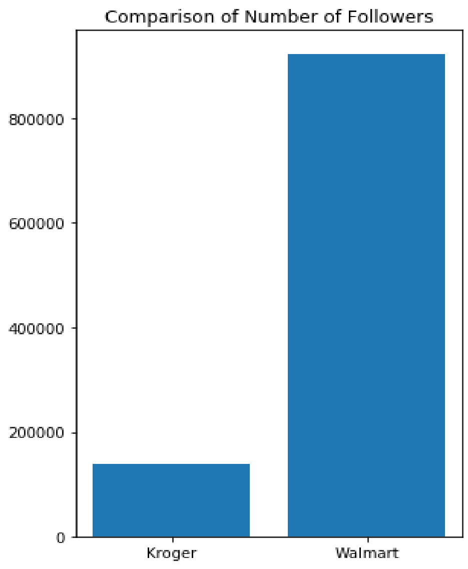


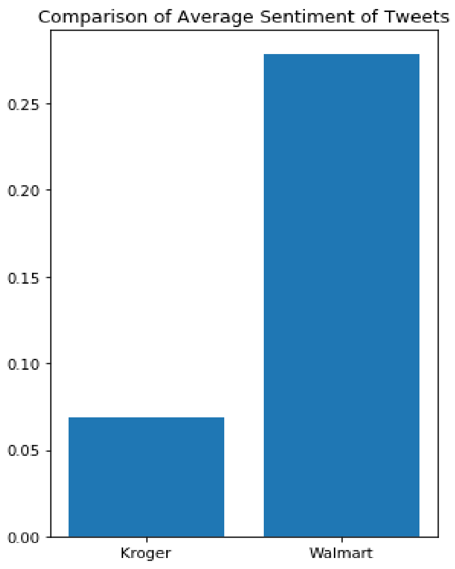


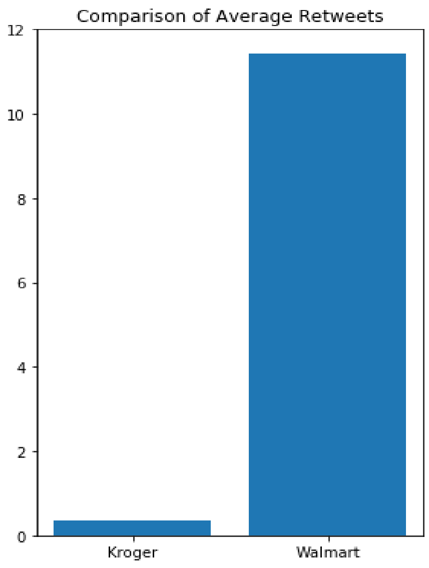


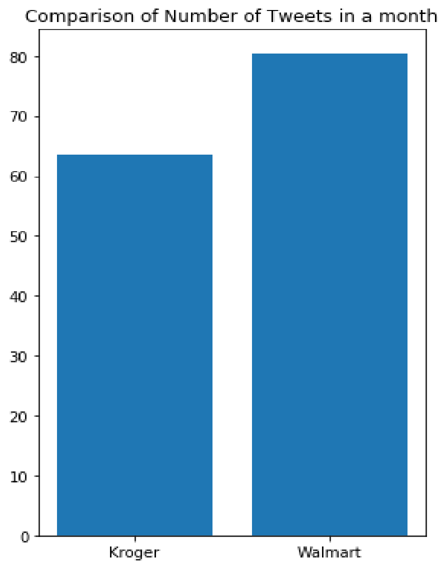

```
Walmart Tweets

Best Tweet: RT @DeliciouslySavv: Stock Up &amp; Save On @TridentGum 8 Packs Available @Walmart PLUS Enter To Be 1 of 7 Winners To Win Walmart Gift Cards! #…

Worst Tweet: RT @BangBangtan_Esp: [INFO] @BTS_twt Precios para LY: Answer (pre-venta) - Amazon ($21.95) - Walmart ($21.95) ht…

Kroger Tweets

Best Tweet: @SarahPribis You must have been rich. For me it was the Lil Hugs from Kroger.

Worst Tweet: RT @Nebuchadneggar: • Aretha is sick and shut in (or an ancestor, if I missed an update). • Robert Glasper is reading Lauryn Hill in the tr…
```


Walmart
Kroger
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Negative
Neutral
Positive


Comparison of Number of Followers
0
200000
400000
600000
800000
Kroger
Walmart


Comparison of Average Sentiment of Tweets
0.00
0.05
0.10
0.15
0.20
0.25
Kroger
Walmart


Comparison of Average Retweets
0
2
4
6
8
10
12
Kroger
Walmart


Comparison of Number of Tweets in a month
0
10
20
30
40
50
60
70
80
Kroger
Walmart