Queries

QUERY 1

#number of users(accounts) created per month

val Query1 = sqlContext.sql("SELECT substring(user.created\_at,5,3) as month, count(user.id) from tweetDatatable group by month");

#to display output in spark

Query1.show();

#importing output from spark to local system(a folder q1.csv will be created in this path C:\\Users\\manas\\Desktop\\pb\\phase2

Query1.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q1.csv");

QUERY 2

#number of tweets from different countries

val Query2 = sqlContext.sql("SELECT place.country,count(\*) AS count FROM tweetDatatable GROUP BY place.country ORDER BY count DESC");

Query2.show(); Query2.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q2.csv");

QUERY 3

#number of tweets based of different states in USA

val Query3=sqlContext.sql("SELECT user.location,count(text) as count FROM tweetDatatable WHERE place.country='United States' AND user.location is not null GROUP BY user.location ORDER BY count DESC");

Query3.show(); Query3.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q3.csv");

QUERY 4

#user with more number of retweets he got for his tweet

val Query4 = sqlContext.sql("SELECT user.screen\_name,text,retweeted\_status.retweet\_count FROM tweetDatatable ORDER BY retweeted\_status.retweet\_count DESC LIMIT 20");

Query4.show(); Query4.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q4.csv");

QUERY 5

#number of tweets based on day

val Query5=sqlContext.sql("SELECT substring(quoted\_status.created\_at,1,3) as day,count(text) as count FROM tweetDatatable GROUP BY day"); Query5.show();

Query5.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q5.csv");

QUERY 6:

#Top Tweet text and Retweet count

val Query6 = sqlContext.sql("SELECT user.name ,retweeted\_status.text AS Retweet\_Text,retweeted\_status.retweet\_count AS Retweet\_Count FROM tweetDatatable WHERE retweeted\_status.retweet\_count IS NOT NULL ORDER BY retweeted\_status.retweet\_count DESC limit 10");

Q6.show();

Q6.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q6.csv");

QUERY 7 :

val Query7 = sqlContext.sql(

("Select count(\*) as count,q.text from (select case when text like '%coronavirus%' then 'coronavirus' when text like '%Narendra Modi%' then 'Narendra Modi' when text like '%realDonaldTrump%' then 'realDonaldTrump' when text like '%BTS\_twt%' then 'BTS\_twt' when text like '%Oscars%' then 'Oscars' when text like '%Grammy%' then 'Grammy' when text like '%iphone%'then 'iphone' when text like '%tesla%' then 'tesla' else 'different sports' end as text from tweetDatatable)q group by q.text"));

Query7.show();

Query7.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q7.csv");

QUERY 8:

#number of users created according to day wise

val Query8=sqlContext.sql("SELECT substring(user.created\_at,1,3) as day,count(\*) as count from tweetDatatable group by day");

Query8.show(); Query8.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q8.csv");

QUERY 9:

number of user accounts created per year

val Query9 = sqlContext.sql("SELECT substring(user.created\_at,27,4) as year,count(\*) from tweetDatatable where user.created\_at is not null group by substring(user.created\_at,27,4) order by count(1) desc")

Query9.show();

Query9.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q9.csv");

QUERY 10:

account verification tweets

val acctVerify=sqlContext.sql("SELECT distinct id, " +

"CASE when user.verified LIKE '%true%' THEN 'VERIFIED ACCOUNT'"+

"when user.verified LIKE '%false%' THEN 'NON-VERIFIED ACCOUNT'"+

"END AS Verified from tweetDatatable where text is not null")

acctVerify.createOrReplaceTempView("acctVerify")

val Query10=sqlContext.sql("SELECT Verified, Count(Verified) as Count from acctVerify where id is NOT NULL and Verified is not null group by Verified order by Count DESC")

Query10.show();

Query10.coalesce(1).write.format("com.databricks.spark.csv").save("C:\\Users\\manas\\Desktop\\pb\\phase2\\q10.csv");