CREATE TABLE if not exists CITY\_ADDRESSES

(

Serial\_No double,PIN STRING,PIND STRING,HSE\_NBR double,HSE\_FRAC\_NBR STRING,

HSE\_DIR\_CD STRING,STR\_NM STRING,STR\_SFX\_CD STRING,STR\_SFX\_DIR\_CD STRING,

UNIT\_RANGE STRING,ZIP\_CD double,LAT double,LON double,X\_COORD\_NBR double,

Y\_COORD\_NBR double,ASGN\_STTS\_IND String,ENG\_DIST STRING,CNCL\_DIST STRING

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

TBLPROPERTIES("skip.header.line.count"="1");

LOAD DATA INPATH '\HdiSamples\Addresses\_in\_the\_City\_of\_Los\_Angeles.csv'OVERWRITE INTO TABLE CITY\_ADDRESSES;

QUERIES LIST:

Select LAT,LON from CITY\_ADDRESSES;

Select STR\_NM,COUNT(STR\_NM) FROM CITY\_ADDRESSES GROUP BY STR\_NM;

Select HSE\_DIR\_CD,COUNT(HSE\_DIR\_CD) FROM CITY\_ADDRESSES GROUP BY HSE\_DIR\_CD;

Select ASGN\_STTS\_IND,COUNT(ASGN\_STTS\_IND) FROM CITY\_ADDRESSES GROUP BY ASGN\_STTS\_IND;

Create table if not exists location\_dis(LAT double,LON double,DIR STRING,distance double);

INSERT OVERWRITE TABLE location\_dis Select LAT,LON,HSE\_DIR\_CD,

2 \* asin(

sqrt(

cos(radians(34.0667))\*

cos(radians(LAT)) \*

pow(sin(radians(-118.1678 - LON)/2),2)

+

pow(sin(radians(34.0667 - LAT)/2),2)

)) \*3956

from CITY\_ADDRESSES;

ASGN\_STTS\_IND

select location\_dis.distance as score, count(\*) as occurences

from (

select case

when distance between 0 and 5 then ' 0-5'

when distance between 6 and 10 then '6-10'

when distance between 11 and 15 then '11-15'

when distance between 16 and 20 then '16-20'

when distance between 21 and 25 then '21-25'

when distance between 26 and 30 then '26-30'

when distance between 31 and 35 then '31-35'

when distance between 36 and 40 then '36-40'

when distance is Null then 'others'

end as distance

from location\_dis) location\_dis

group by location\_dis.distance

order by occurences;

select location\_dis.distance as score, count(\*) as occurences

from (

select case

when distance between 0 and 5 then ' 0- 5'

when distance between 5 and 10 then '5-10'

when distance between 10 and 15 then '10-15'

when distance between 15 and 20 then '15-20'

when distance between 20 and 25 then '20-25'

when distance between 25 and 30 then '25-30'

when distance between 30 and 35 then '30-35'

when distance between 35 and 40 then '35-40'

when distance between 40 and 50 then '40-50'

else '>50'

end as distance

from location\_dis DIR='N') location\_dis

group by location\_dis.distance

order by occurences;

TABLE 2:

Create table if not exists location\_dis1(LAT double,LON double,DIR STRING,ASGN\_STTS\_IND STRING,distance double);

INSERT OVERWRITE TABLE location\_dis1 Select LAT,LON,HSE\_DIR\_CD,ASGN\_STTS\_IND,

2 \* asin(

sqrt(

cos(radians(34.0667))\*

cos(radians(LAT)) \*

pow(sin(radians(-118.1678 - LON)/2),2)

+

pow(sin(radians(34.0667 - LAT)/2),2)

)) \*3956

from CITY\_ADDRESSES;

CREATE TABLE if not exists CITY\_ADDRESSES

(

Serial\_No double,PIN STRING,PIND STRING,HSE\_NBR double,HSE\_FRAC\_NBR STRING,

HSE\_DIR\_CD STRING,STR\_NM STRING,STR\_SFX\_CD STRING,STR\_SFX\_DIR\_CD STRING,

UNIT\_RANGE STRING,ZIP\_CD double,LAT double,LON double,X\_COORD\_NBR double,

Y\_COORD\_NBR double,ASGN\_STTS\_IND String,ENG\_DIST STRING,CNCL\_DIST STRING

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY ','

TBLPROPERTIES("skip.header.line.count"="1");

LOAD DATA INPATH '\HdiSamples\Addresses\_in\_the\_City\_of\_Los\_Angeles.csv'OVERWRITE INTO TABLE CITY\_ADDRESSES;

Select ASGN\_STTS\_IND,COUNT(ASGN\_STTS\_IND) AS cnt FROM CITY\_ADDRESSES GROUP BY ASGN\_STTS\_IND

ORDER BY cnt DESC;

Select STR\_NM,COUNT(STR\_NM) as cnt FROM CITY\_ADDRESSES WHERE ASGN\_STTS\_IND='U' GROUP BY STR\_NM Order BY cnt DESC;

Select HSE\_DIR\_CD,COUNT(HSE\_DIR\_CD) AS cnt FROM CITY\_ADDRESSES GROUP BY HSE\_DIR\_CD

ORDER BY cnt DESC;

Create table if not exists location\_dis1(LAT double,LON double,DIR STRING,ASGN\_STTS\_IND STRING,distance double);

INSERT OVERWRITE TABLE location\_dis1 Select LAT,LON,HSE\_DIR\_CD,ASGN\_STTS\_IND,

2 \* asin(

sqrt(

cos(radians(34.0667))\*

cos(radians(LAT)) \*

pow(sin(radians(-118.1678 - LON)/2),2)

+

pow(sin(radians(34.0667 - LAT)/2),2)

)) \*3956

from CITY\_ADDRESSES;

select location\_dis1.distance as score, count(\*) as occurences

from (

select case

when distance between 0 and 5 then ' 0- 5'

when distance between 5 and 10 then '5-10'

when distance between 10 and 15 then '10-15'

when distance between 15 and 20 then '15-20'

when distance between 20 and 25 then '20-25'

when distance between 25 and 30 then '25-30'

when distance between 30 and 35 then '30-35'

when distance between 35 and 40 then '35-40'

when distance between 40 and 50 then '40-50'

else '>50'

end as distance

from location\_dis1 where ASGN\_STTS\_IND='U' ) location\_dis1

group by location\_dis1.distance

order by occurences;

from pyspark.sql.functions import pow, col

from math import sin, cos, sqrt, atan2, radians,asin

import math

def distance(lat1,lon1):

try:

lat2 = 34.0522

lon2 = -118.2437

radius = 3965

dlat = math.radians(lat2-lat1)

dlon = math.radians(lon2-lon1)

a = math.sin(dlat/2) \* math.sin(dlat/2) + math.cos(math.radians(lat1)) \

\* math.cos(math.radians(lat2)) \* math.sin(dlon/2) \* math.sin(dlon/2)

c = 2 \* math.atan2(math.sqrt(a), math.sqrt(1-a))

d = radius \* c

return d

except:

return 0.0

data1=sqlContext.sql("SELECT distinct(business\_id), CAST(latitude as float),CAST(longitude as float),Tuesday\_\_close,Tuesday\_\_open,categories\_\_001 FROM business\_data5");

fil = sqlContext.sql("SELECT distinct(business\_id), CAST(latitude as float),CAST(longitude as float) FROM business\_data5");

lamapdata=fil.map(lambda (x,y,z):(x,distance(y,z)))

busdata=lamapdata.toDF(['id','distance'])

finalresult=fil.join(busdata,fil.business\_id==busdata.id)

finalresult.show()