Spark use case

**Requirement:**

Conversion of Hive queries into spark job for performance improvement,

**Data preparation/ingestion:**

Since data is already available in PostgreSQL server, we are going to extract the data from server through JDBC call and save into temp table in spark memory using below code

As there are more than one table for data ingestion instead of rewriting the code, have looped the JDBC call and it read the table names from list provided and extract it recursively

tablename\_list = ['cust\_info\_s','trans\_info\_s','offer\_info\_s']

url = "jdbc:postgresql://localhost:5432/postgres"

reader = (

sqlContext.read.format("jdbc")

.option("url", url)

.option("user", "postgres")

.option("password", "puvi")

.option("driver", "org.postgresql.Driver")

)

for tablename in tablename\_list:

reader.option("dbtable", tablename).load().registerTempTable(tablename)

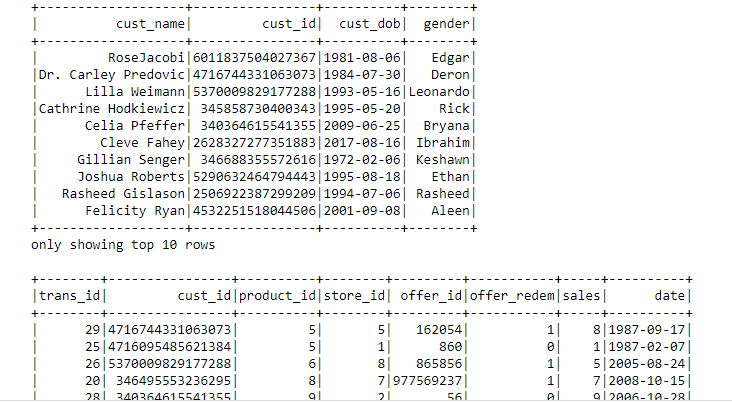
**Data check of source:**

sqlc = SQLContext(sc)

sqlc.sql("select \* from cust\_info\_s").show(10)

sqlc.sql("select \* from trans\_info\_s").show(10)

sqlc.sql("select \* from offer\_info\_s").show(10)



**Data transformation:**

Implementing business logic and transformation of data according to the requirement’s,

df\_transf = sqlc.sql(""" WITH sales\_data AS ( select sum(sales) as total\_sales,count(trans\_id) as visits,cust\_id,date

from trans\_info\_s

group by date,cust\_id),

offer\_data AS (

select count(o.offer\_id) as no\_offer\_received,sum(offer\_redem) as no\_offer\_redem,o.cust\_id

from offer\_info\_s o

inner join trans\_info\_s t

on t.cust\_id = o.cust\_id

group by o.cust\_id order by o.cust\_id)

select b.date,b.cust\_id,a.cust\_name,a.cust\_dob,

int(datediff(current\_date(),TO\_DATE(CAST(UNIX\_TIMESTAMP(a.cust\_dob,'yyyy-MM-dd') AS TIMESTAMP)))/365) as age,

total\_sales,c.no\_offer\_received,c.no\_offer\_redem,visits, date\_format(to\_date(b.date),'E') as days,

int(date\_format(to\_date(b.date),'w')) AS week\_number

from cust\_info\_s a

left join sales\_data b

on a.cust\_id = b.cust\_id

inner join offer\_data c

on b.cust\_id = c.cust\_id

group by b.date,a.cust\_id,b.cust\_id,a.cust\_name,a.cust\_dob,c.no\_offer\_received,c.no\_offer\_redem,total\_sales,visits,days

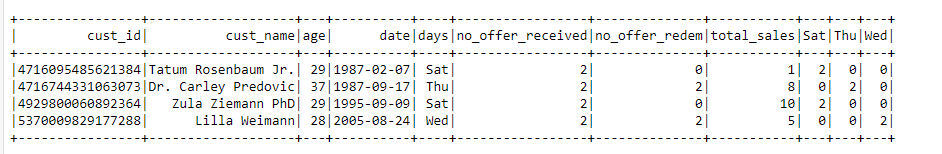
order by a.cust\_id

""")

final\_dfs = df\_transf. orderBy('cust\_id'). groupby('cust\_id','cust\_name','age','date','days','no\_offer\_received','no\_offer\_redem','total\_sales'). pivot('days').max('no\_offer\_received').fillna(0)

#Sampling data - In case of job failure data can be viewed in Spark UI

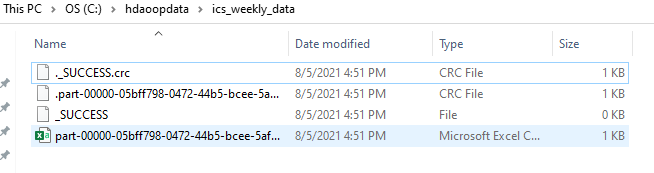
final\_dfs.show()



**Data Loading:**

Loading the final data frame into single CSV file using repartition.

final\_dfs.repartition(1).write.mode("overwrite").option("header",True).csv("file:///C:/hdaoopdata/ics\_weekly\_data/")



**Code base :**