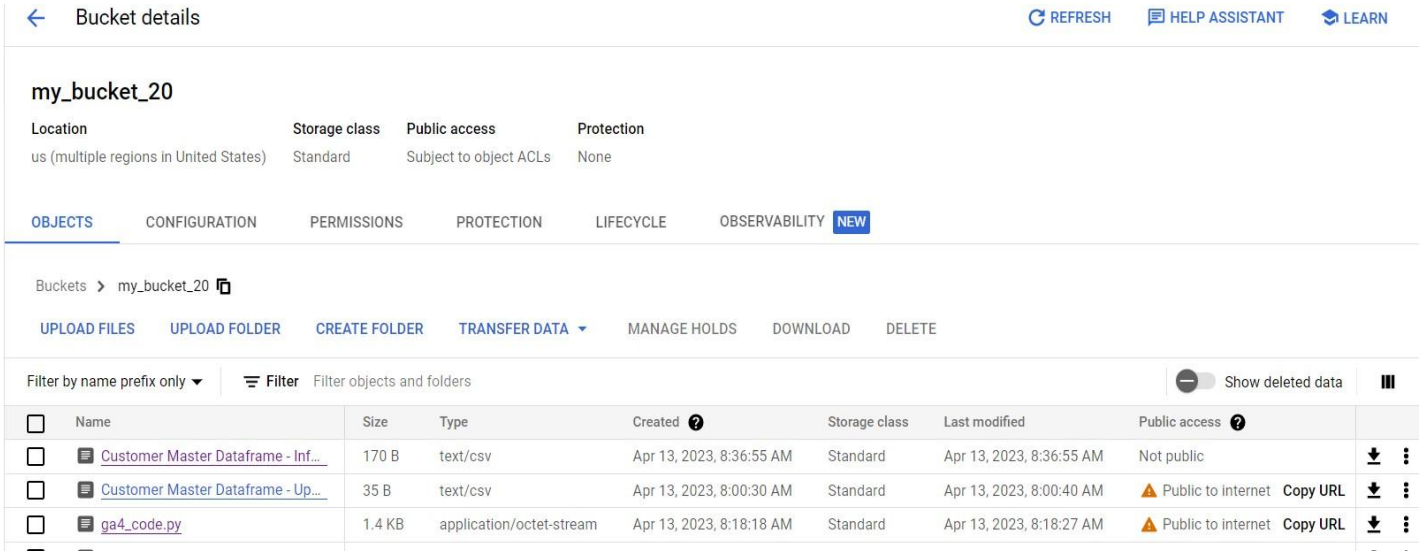
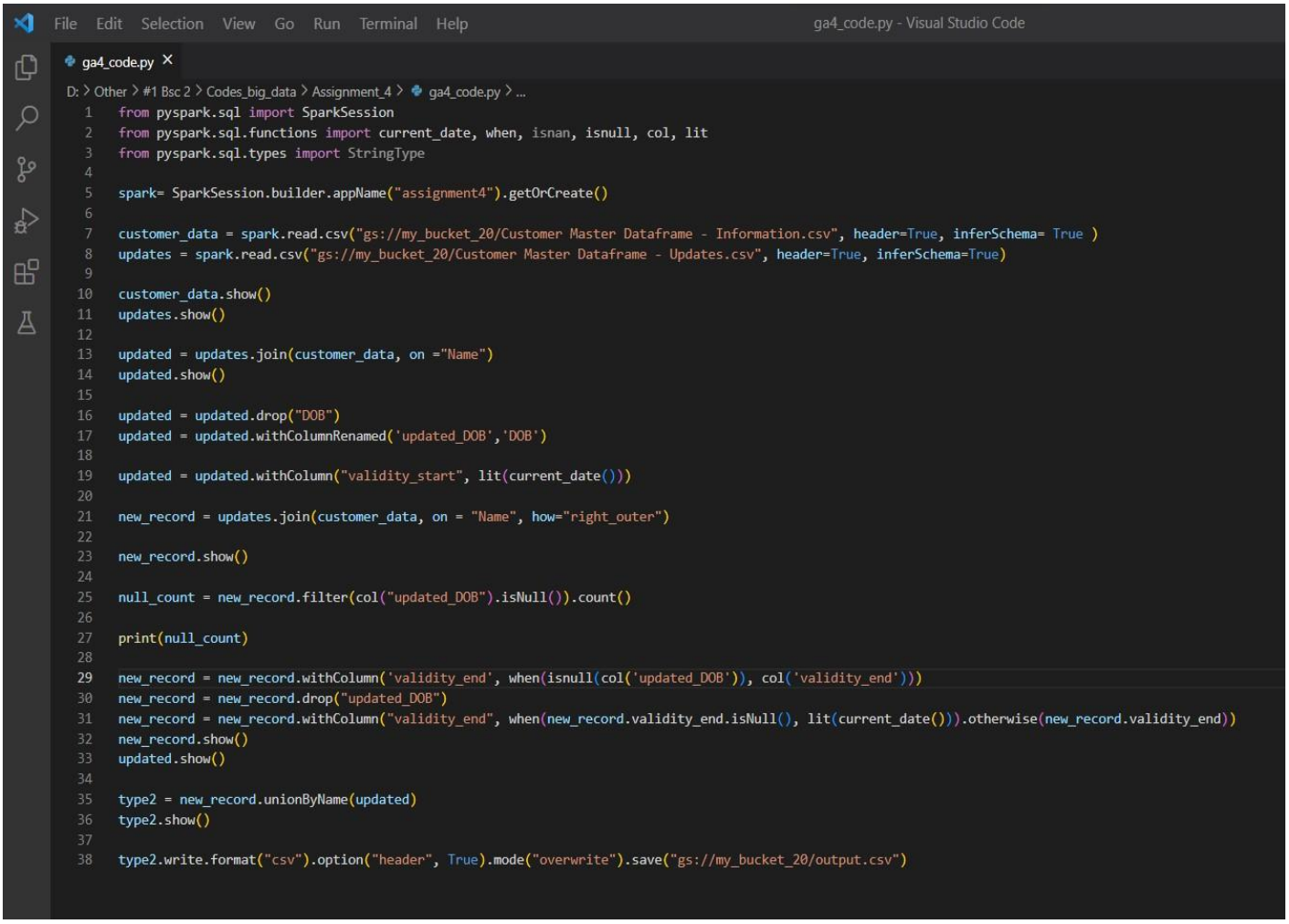
1. Moving files to bucket

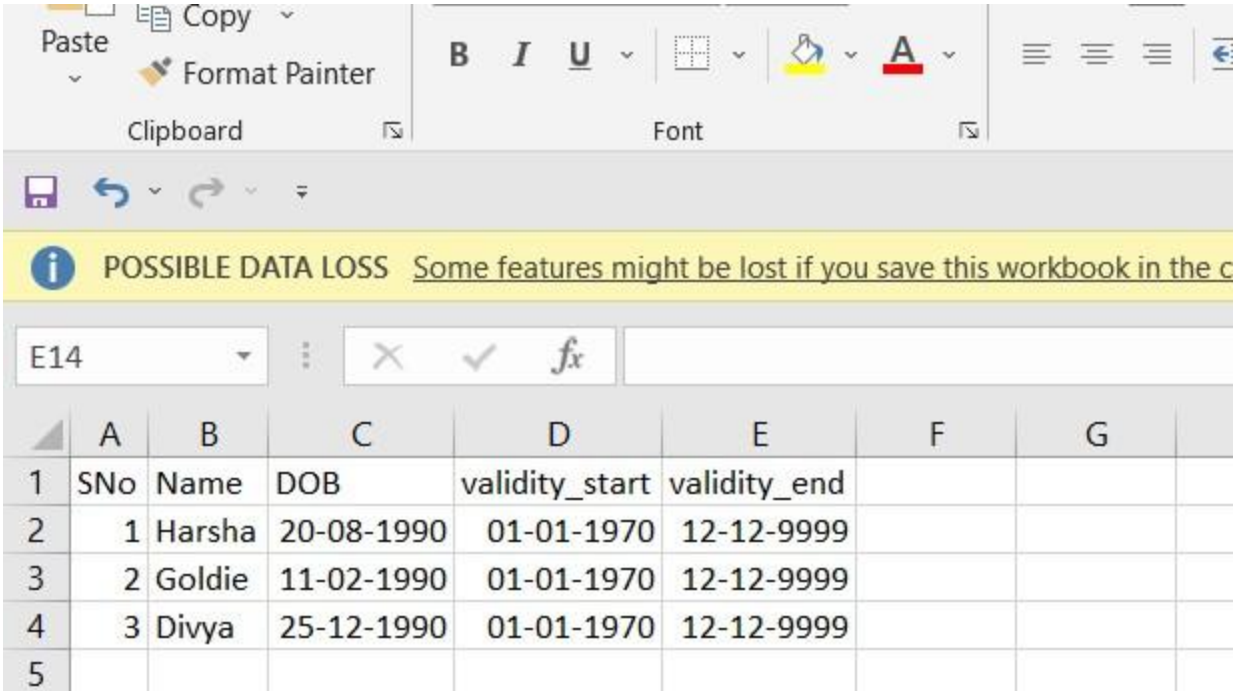


2. Codes

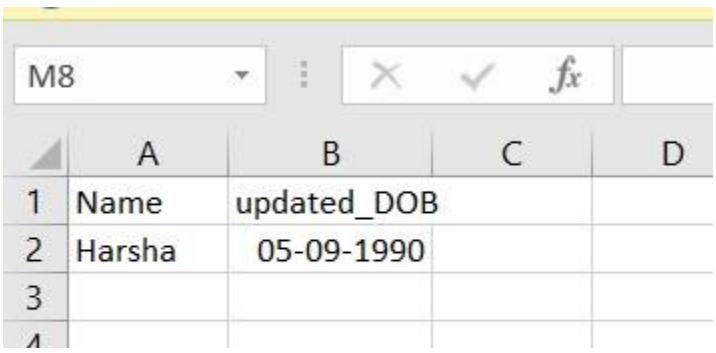



```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import current_date, when, isnan, isnull, col, lit
from pyspark.sql.types import StringType

spark= SparkSession.builder.appName("assignment4").getOrCreate()

customer_data = spark.read.csv("gs://my_bucket_20/Customer Master Dataframe - Information.csv", header=True, inferSchema= True )
updates = spark.read.csv("gs://my_bucket_20/Customer Master Dataframe - Updates.csv", header=True, inferSchema=True)

customer_data.show()
updates.show()

updated = updates.join(customer_data, on ="Name")
updated.show()

updated = updated.drop("DOB")
updated = updated.withColumnRenamed('updated_DOB','DOB')

updated = updated.withColumn("validity_start", lit(current_date()))

new_record = updates.join(customer_data, on = "Name", how="right_outer")

new_record.show()

null_count = new_record.filter(col("updated_DOB").isNull()).count()

print(null_count)

new_record = new_record.withColumn('validity_end', when(isnull(col('updated_DOB')), col('validity_end')))
new_record = new_record.drop("updated_DOB")
new_record = new_record.withColumn("validity_end", when(new_record.validity_end.isNull(), lit(current_date())).otherwise(new_record.validity_end))
new_record.show()
updated.show()

type2 = new_record.unionByName(updated)
type2.show()

type2.write.format("csv").option("header", True).mode("overwrite").save("gs://my_bucket_20/output.csv")
```

3. The CSV Files

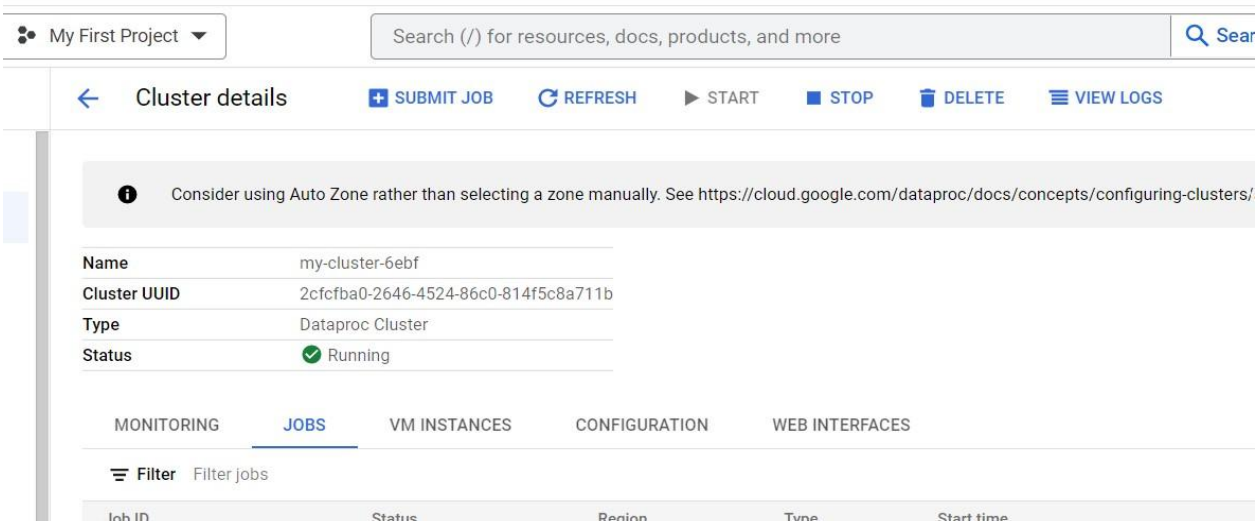


	A	B	C	D	E	F	G
1	SNo	Name	DOB	validity_start	validity_end		
2	1	Harsha	20-08-1990	01-01-1970	12-12-9999		
3	2	Goldie	11-02-1990	01-01-1970	12-12-9999		
4	3	Divya	25-12-1990	01-01-1970	12-12-9999		
5							

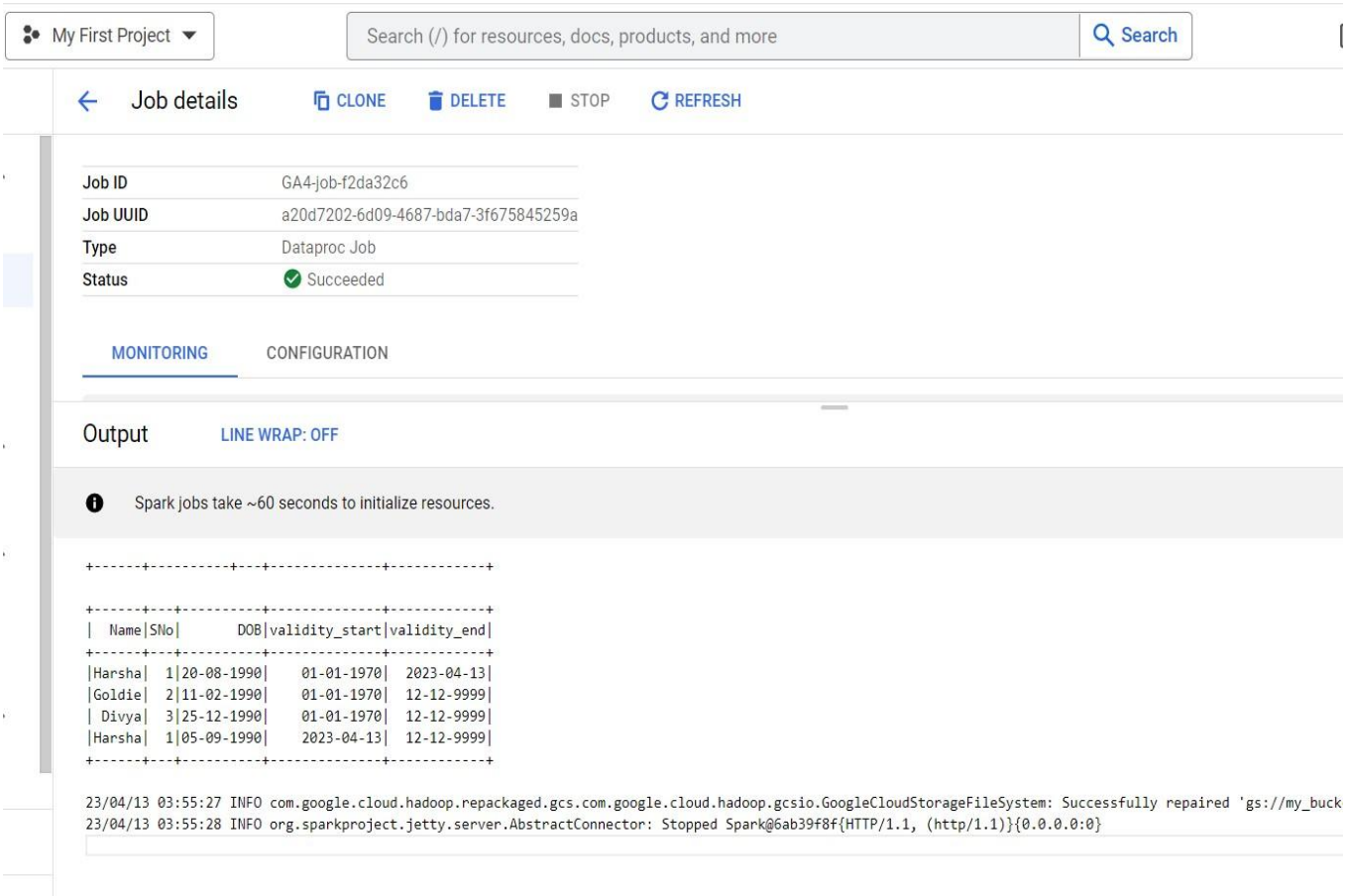


	A	B	C	D
1	Name	updated_DOB		
2	Harsha	05-09-1990		
3				

4. Creating Clusters



5. Creating Job & Output



6. The final Output.csv