**Assignment 15 June**

**Name – Atharva Shivaji Jadhav Roll Number - DXC262AB12035**

**Batch – Dxc-262-Analytics-B12-Azure Company – DXC Technology**

**Employee Domain –Azure Analytics Training Under – Manipal Pro Learn**

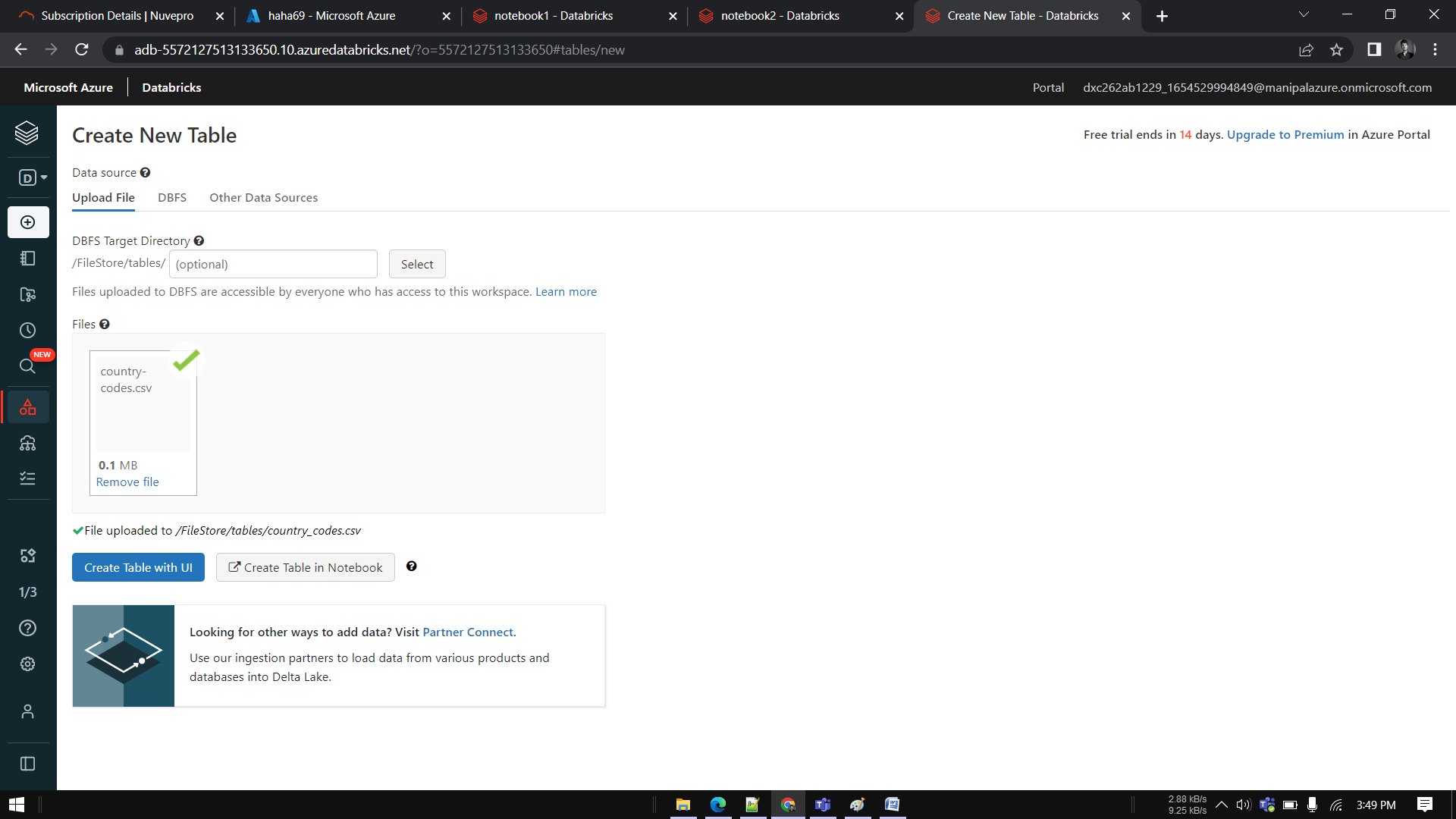
Assignement - 15th June 2022:

------------------------------

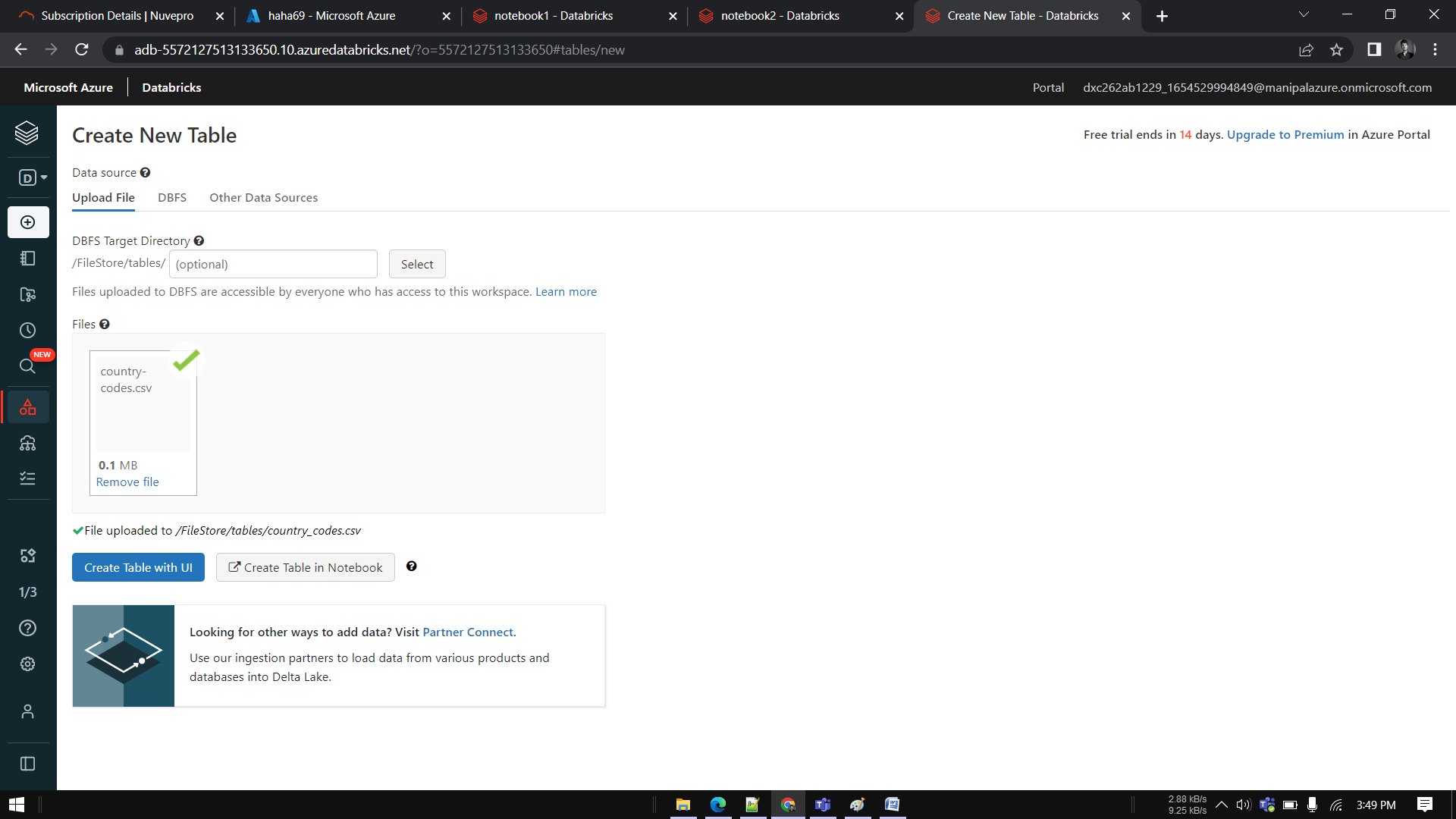
**1.Using archive1.zip file - please ingest data into databricks DBFS path & query the data,**

**redesign columns accordingly using DataFrame commands - display with notebooks accordingly**

Create Cluster and create a notebook attach it to the cluster



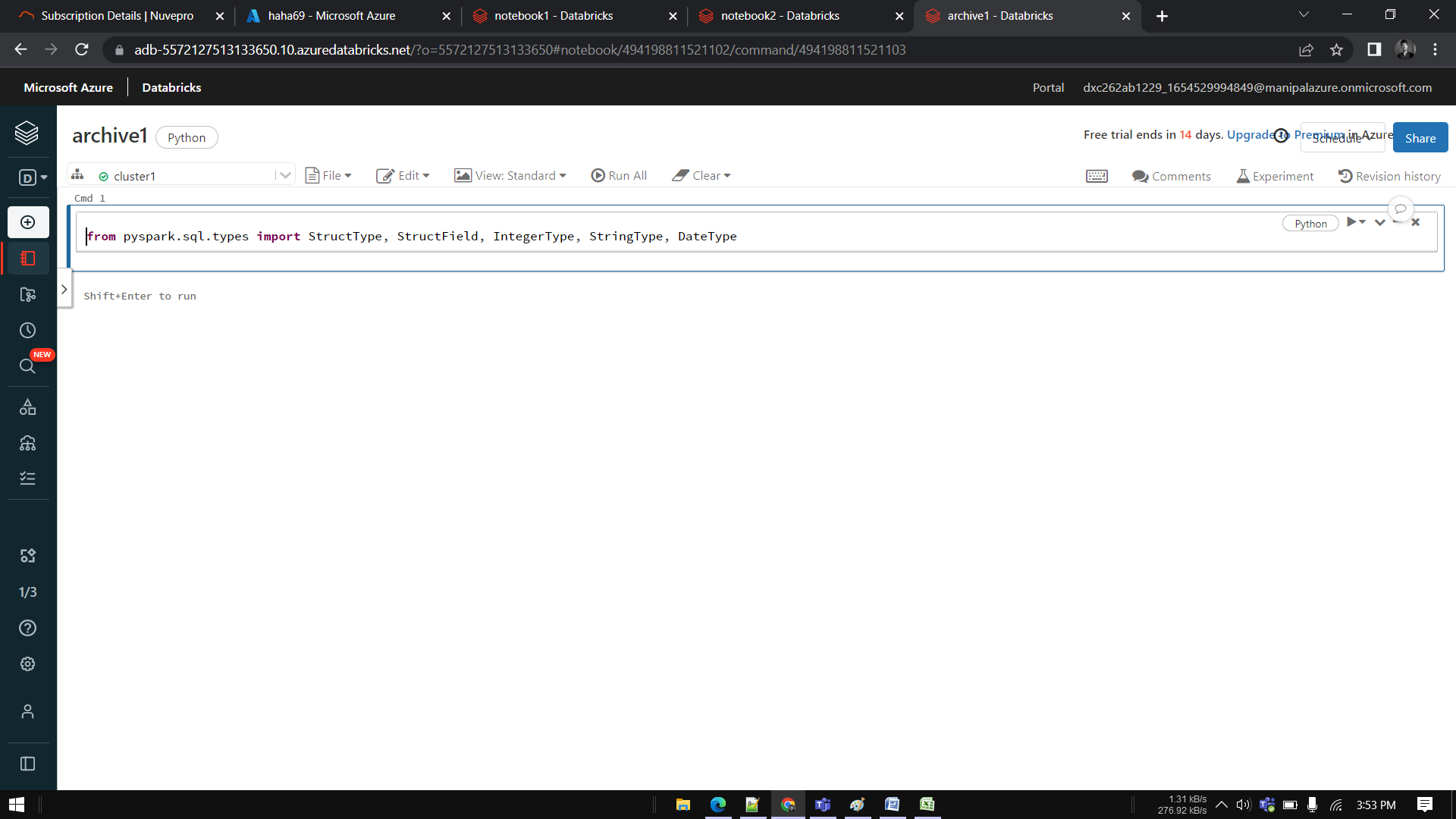
Go to notebook and check whether cluster is running and attached to notebook



Declare data types—

And Run command

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType



Create schemas –

country\_codes\_schema=StructType(fields=[StructField("Dial", IntegerType(),False),

StructField("FIFA", StringType(),True),

StructField("MARC", StringType(),True),

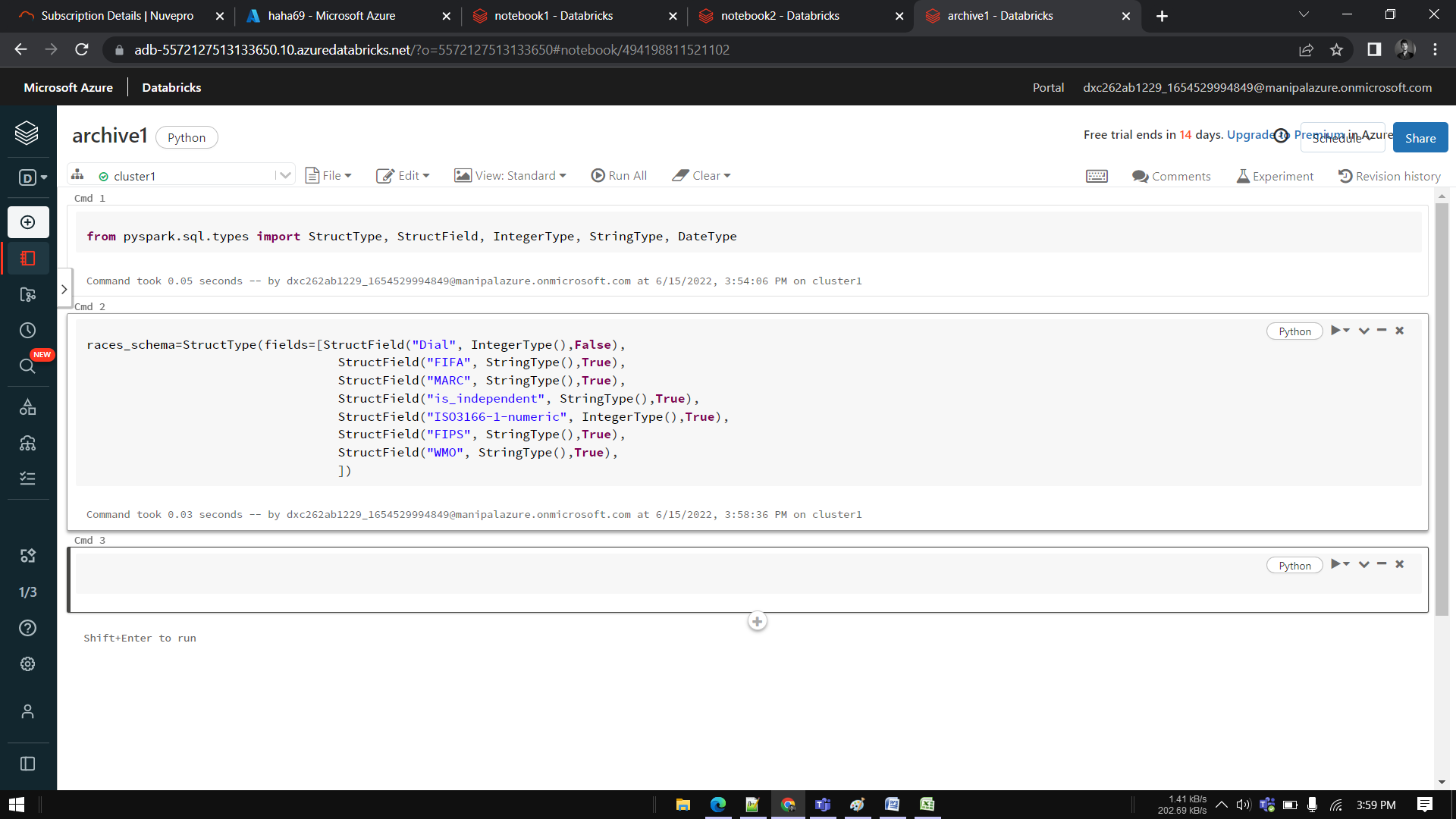
StructField("is\_independent", StringType(),True),

StructField("ISO3166-1-numeric", IntegerType(),True),

StructField("FIPS", StringType(),True),

StructField("WMO", StringType(),True),

])



Creating dataFrame

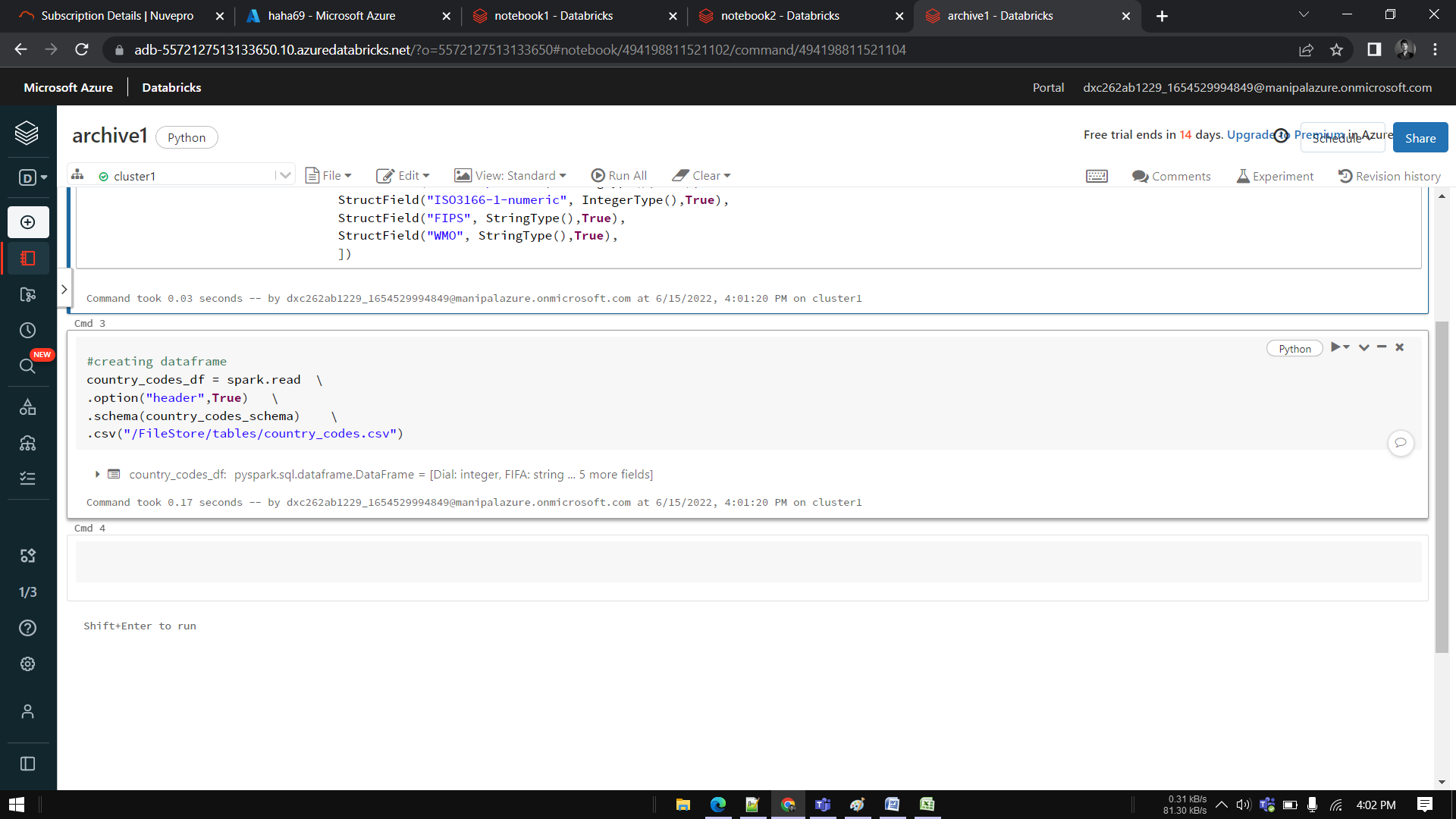
#creating dataframe

country\_codes\_df = spark.read \

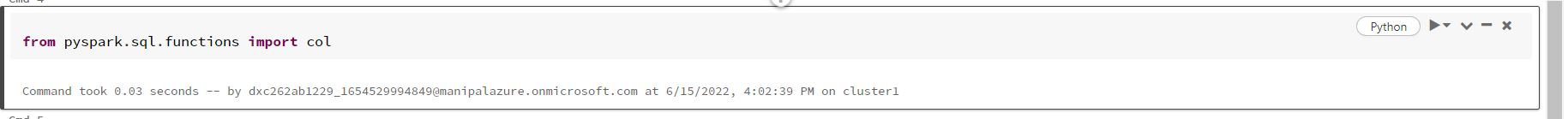
.option("header",True) \

.schema(country\_codes\_schema) \

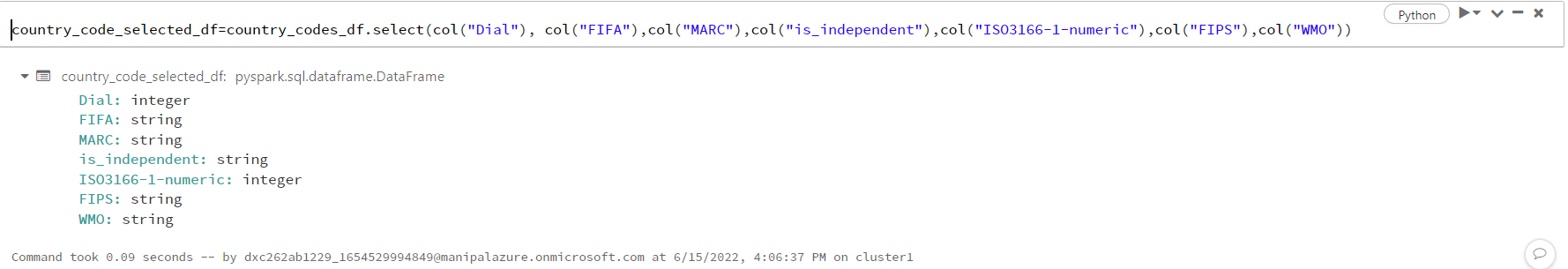
.csv("/FileStore/tables/country\_codes.csv")



from pyspark.sql.functions import col

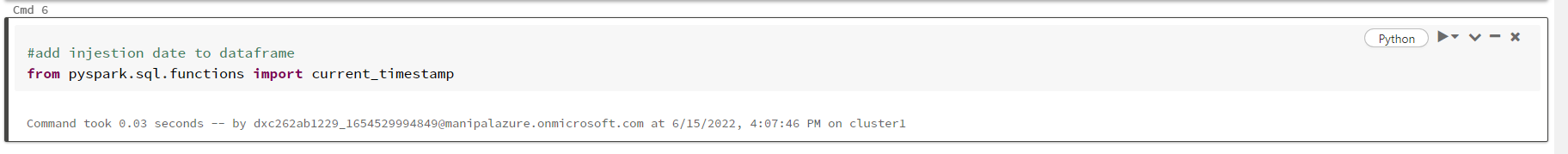


country\_code\_selected\_df=country\_codes\_df.select(col("Dial"), col("FIFA"),col("MARC"),col("is\_independent"),col("ISO3166-1-numeric"),col("FIPS"),col("WMO"))

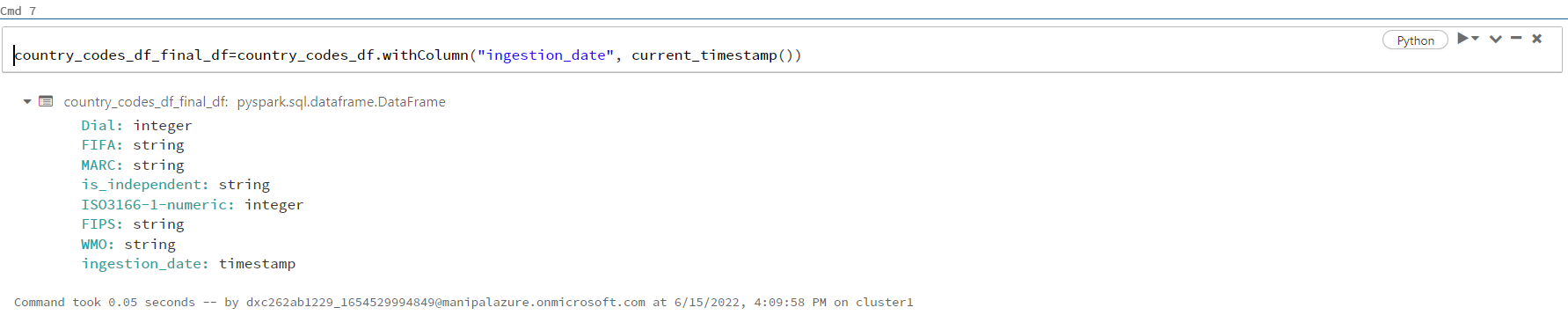


#add injestion date to dataframe

from pyspark.sql.functions import current\_timestamp

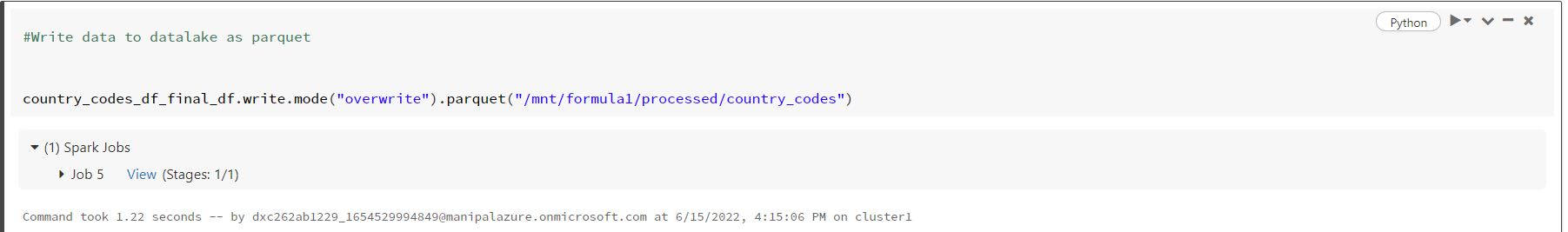


country\_codes\_df\_final\_df=country\_codes\_df.withColumn("ingestion\_date", current\_timestamp())

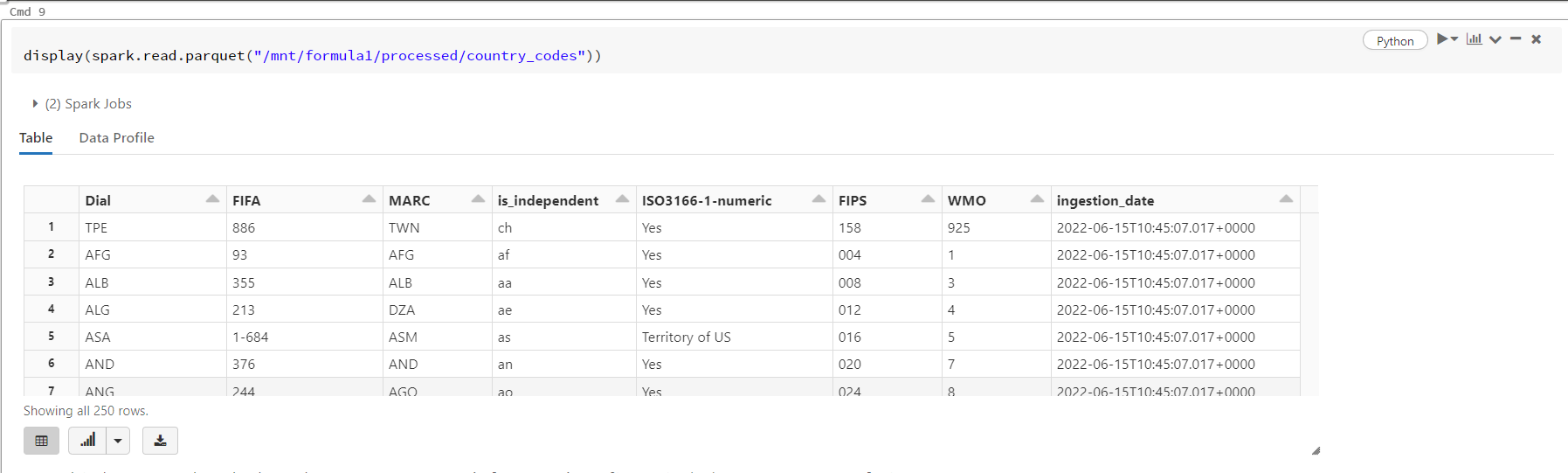


#Write data to datalake as parquet

country\_codes\_df\_final\_df.write.mode("overwrite").parquet("/mnt/formula1/processed/country\_codes")



display(spark.read.parquet("/mnt/formula1/processed/country\_codes"))



country\_codes\_with\_timestamp\_df=country\_codes\_df\_final\_df.withColumn("injestion\_date", current\_timestamp()) \

.withColumn("country\_code\_timestamp", to\_timestamp(concat(col('dial')),'yyyy-mm-dd HH:mm:ss'))



Rename a column—

country\_codes\_df\_final\_df.withColumnRenamed("FIFA","ID").printSchema()

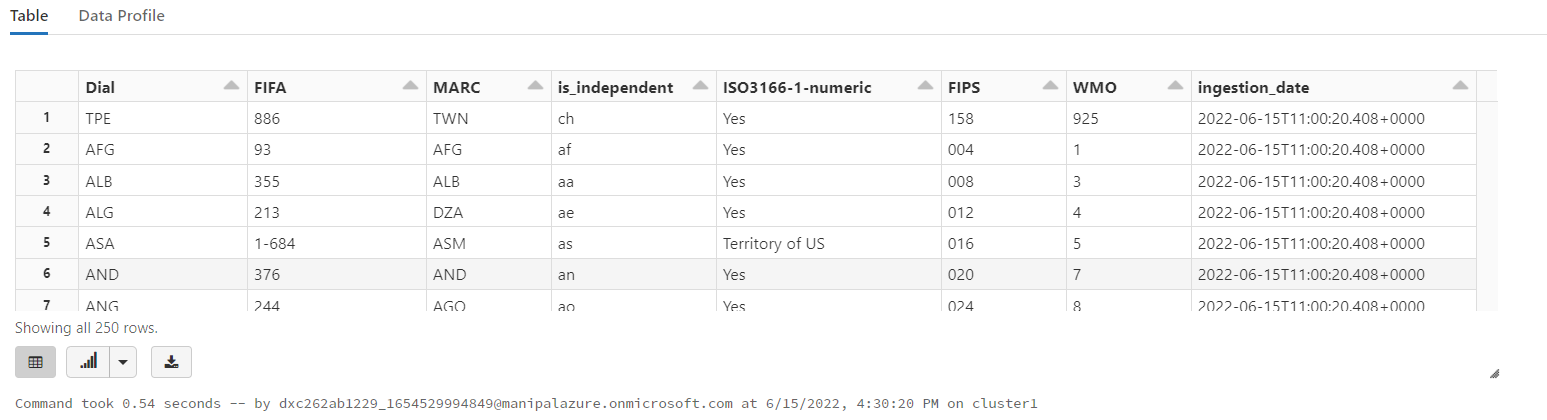


country\_codes\_with\_timestamp\_df=country\_codes\_df\_final\_df.withColumn("injestion\_date", current\_timestamp()) \

.withColumn("country\_code\_timestamp", to\_timestamp(concat(col('dial')),'yyyy-mm-dd HH:mm:ss'))



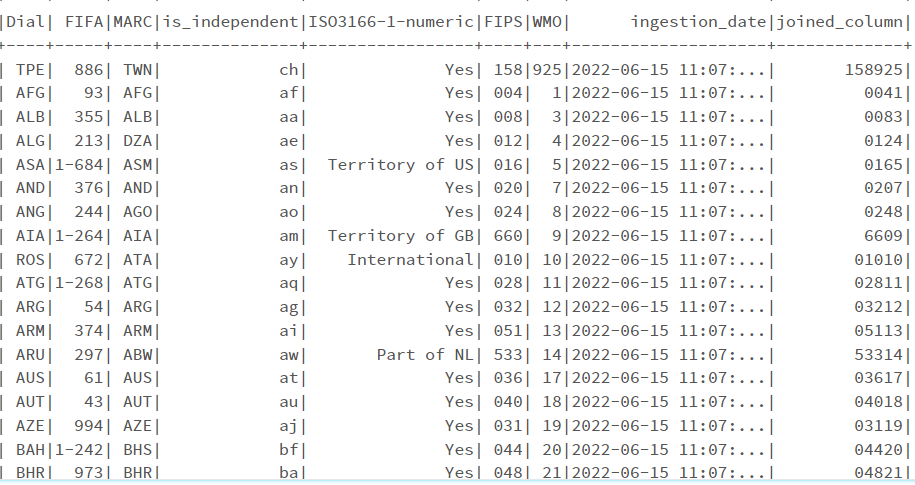
display(country\_codes\_df\_final\_df)



Merge 2 columns—

df=country\_codes\_df\_final\_df.withColumn('joined\_column', sf.concat(sf.col('FIPS'),sf.col('WMO')))

df.show()



**2.Using archive2.zip file - please ingest data into databricks DBFS path & query the data redesign columns accordingly using DataFrame commands - display with notebooks accordingly**

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType

nces\_schema=StructType(fields=[StructField("Value", IntegerType(),False),

StructField("State", StringType(),True),

StructField("Type", StringType(),True),

StructField("Length", StringType(),True),

StructField("Expense", StringType(),True),

])

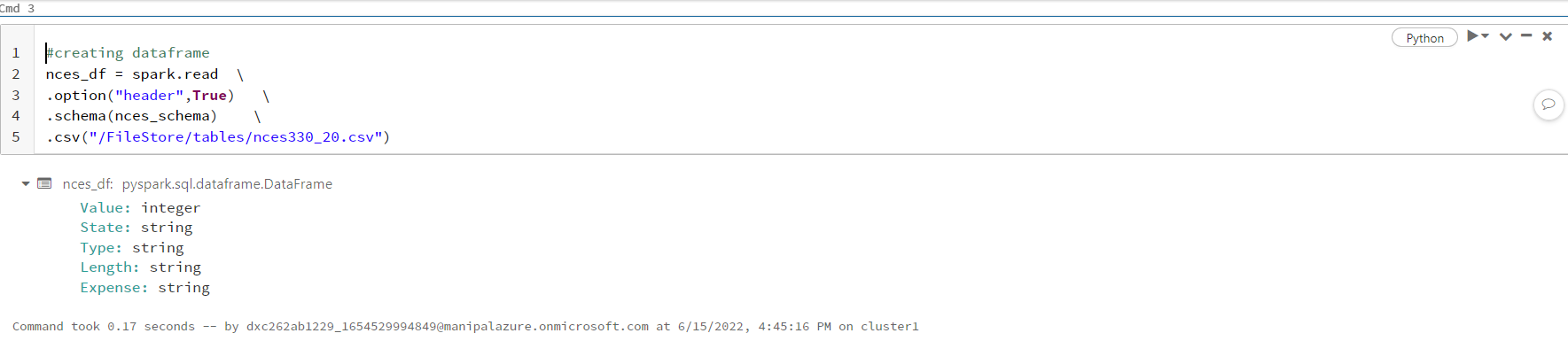
#creating dataframe

nces\_df = spark.read \

.option("header",True) \

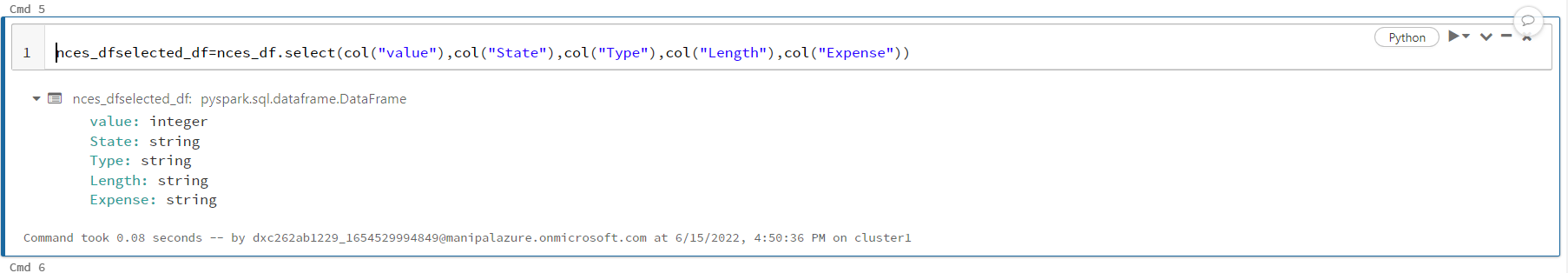
.schema(nces\_schema) \

.csv("/FileStore/tables/nces330\_20.csv")



from pyspark.sql.functions import col

nces\_dfselected\_df=nces\_df.select(col("value"),col("State"),col("Type"),col("Length"),col("Expense"))



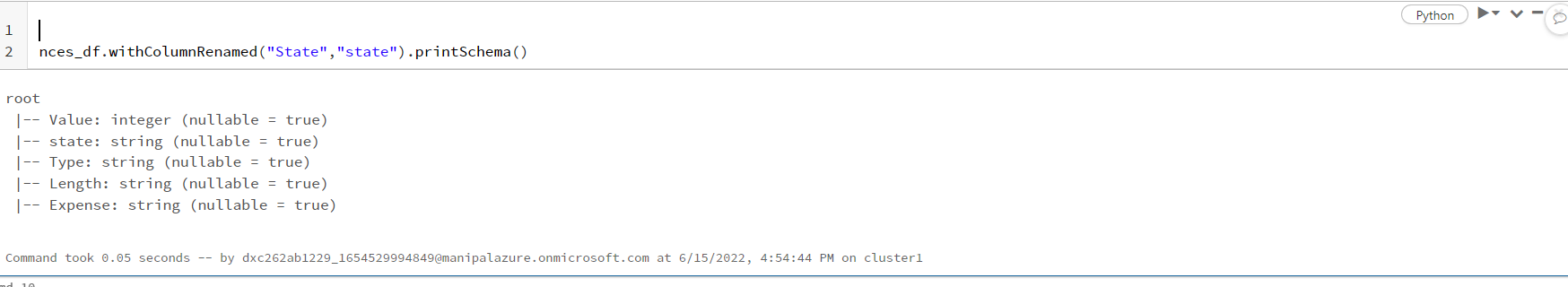
#add injestion date to dataframe

from pyspark.sql.functions import current\_timestamp

#Write data to datalake as parquet

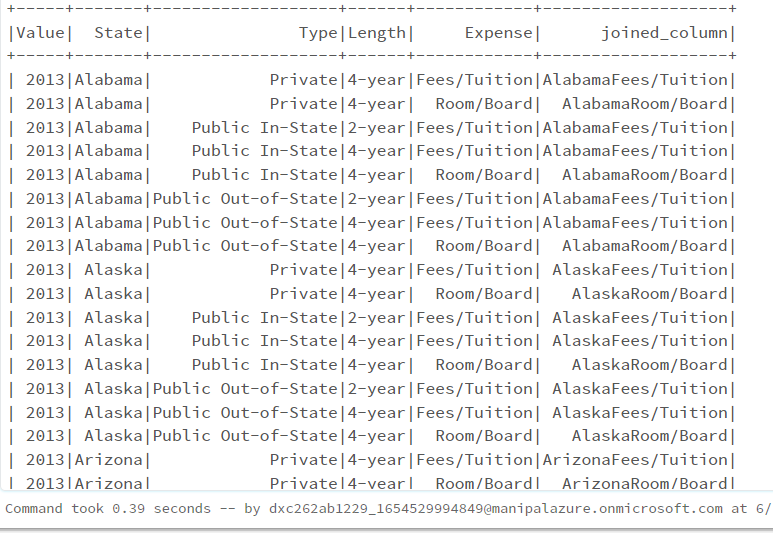
nces\_df.write.mode("overwrite").parquet("/mnt/formula1/processed/nces")

nces\_df.withColumnRenamed("State","state").printSchema()



df=nces\_df.withColumn('joined\_column', sf.concat(sf.col('state'),sf.col('Expense')))

df.show()



**3.Using archive3.zip file - please ingest data into databricks DBFS path & query the data**

**redesign columns accordingly using DataFrame commands - display with notebooks accordingly**

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType

data\_schema=StructType(fields=[

StructField("tweet\_text", StringType(),True),

StructField("emotion\_in\_tweet\_is\_directed\_at", StringType(),True),

StructField("is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product", StringType(),True),

StructField("Expense", StringType(),True),

])

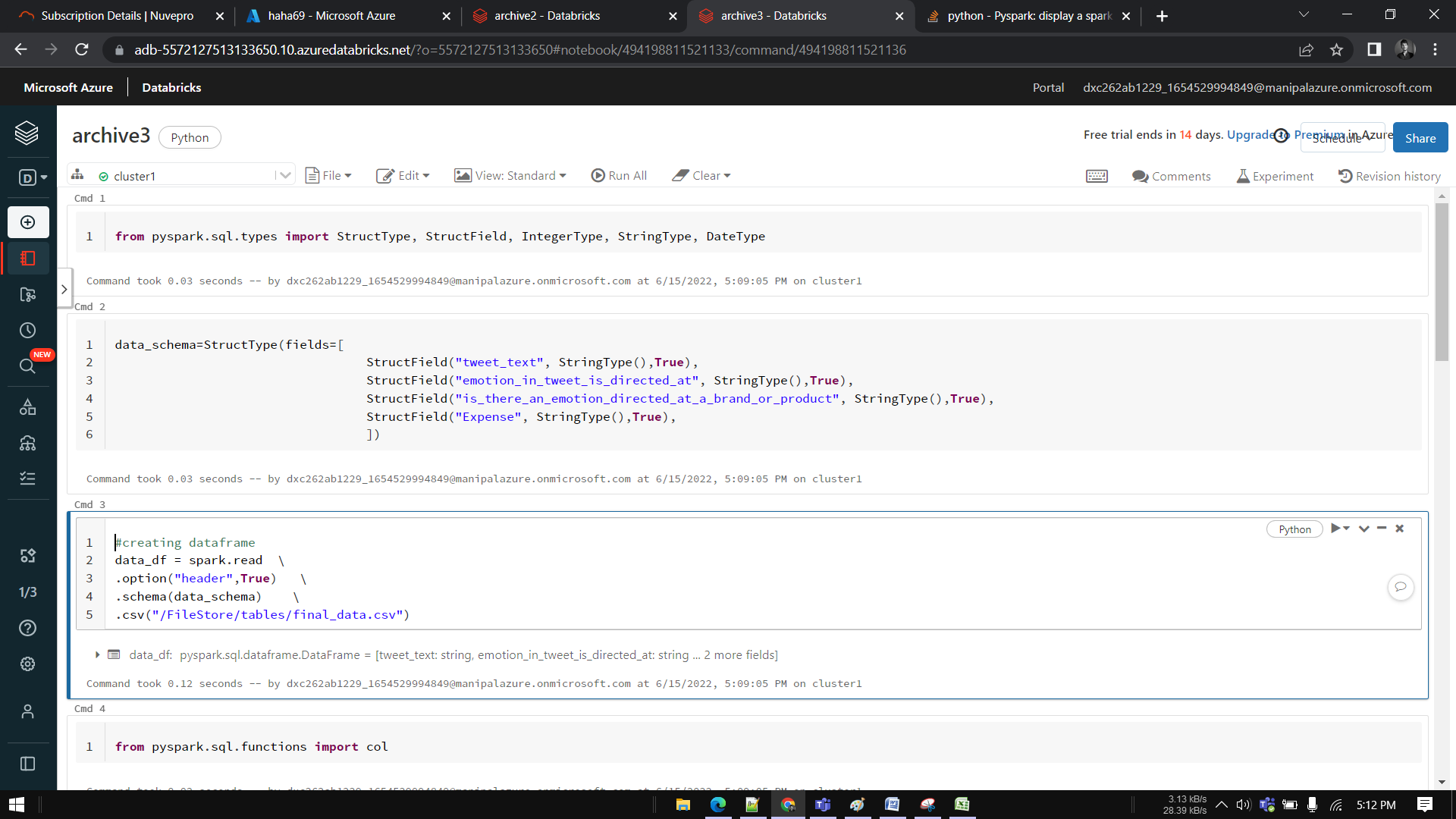
#creating dataframe

data\_df = spark.read \

.option("header",True) \

.schema(data\_schema) \

.csv("/FileStore/tables/final\_data.csv")



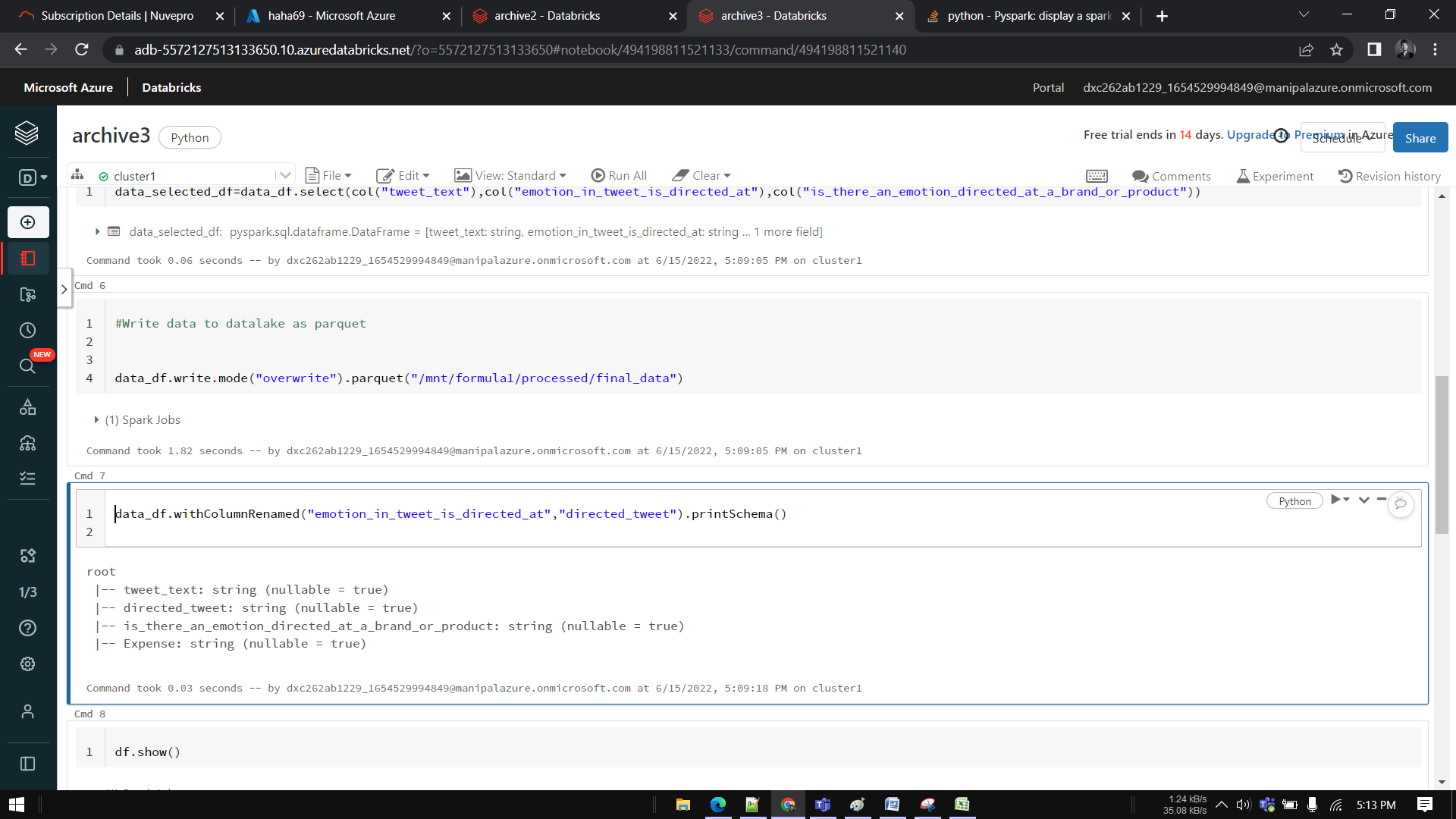
from pyspark.sql.functions import col

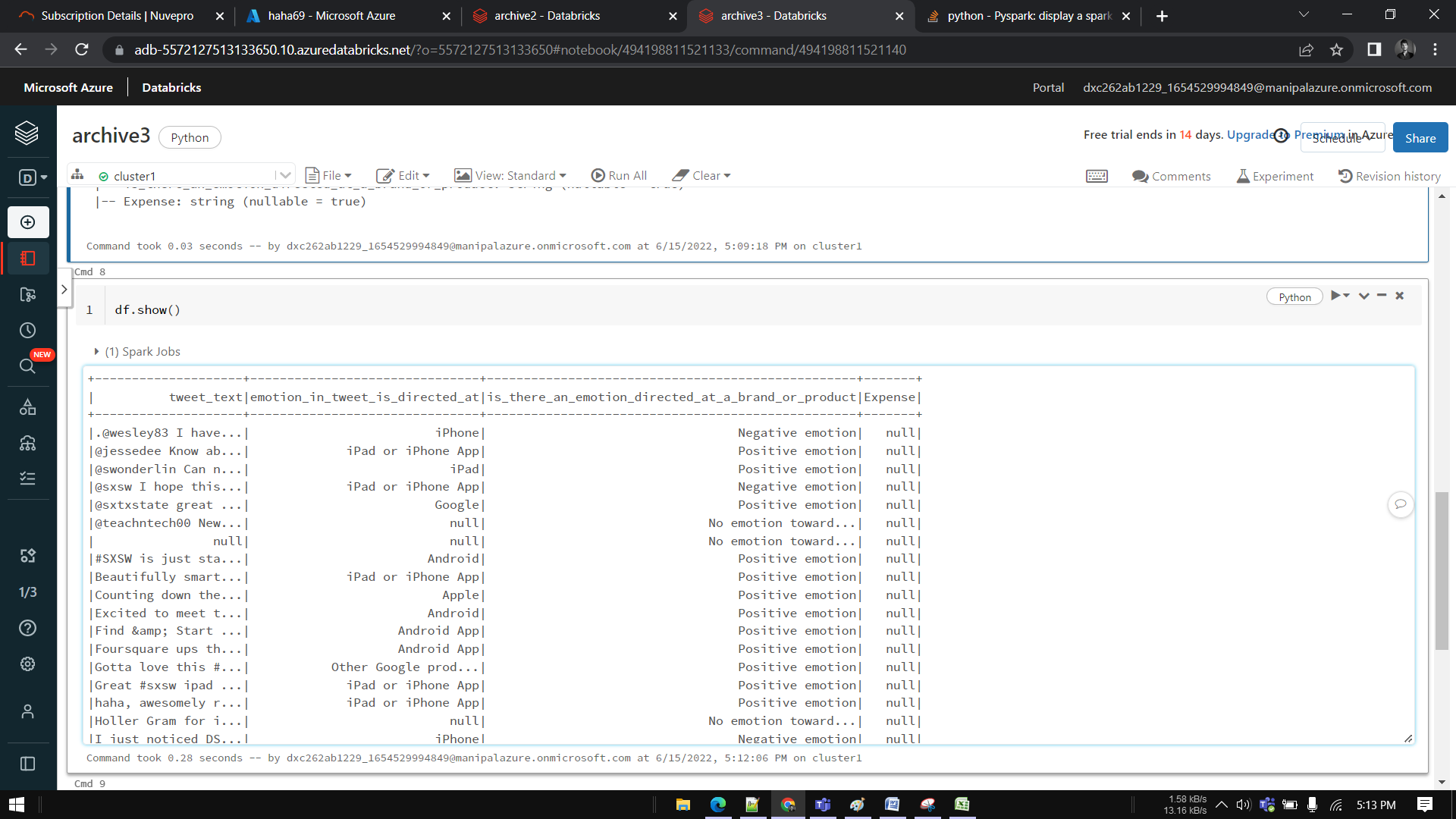
data\_selected\_df=data\_df.select(col("tweet\_text"),col("emotion\_in\_tweet\_is\_directed\_at"),col("is\_there\_an\_emotion\_directed\_at\_a\_brand\_or\_product"))

#Write data to datalake as parquet

data\_df.write.mode("overwrite").parquet("/mnt/formula1/processed/final\_data")

data\_df.withColumnRenamed("emotion\_in\_tweet\_is\_directed\_at","directed\_tweet").printSchema()





**4.Using archive4.zip file - please ingest data into databricks DBFS path & query the data**

**redesign columns accordingly using DataFrame commands - display with notebooks accordingly**

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType

data\_schema=StructType(fields=[

StructField("S No.", IntegerType(),False),

StructField("Title", StringType(),True),

StructField("Decisions", StringType(),True),

StructField("Words", IntegerType(),True),

])

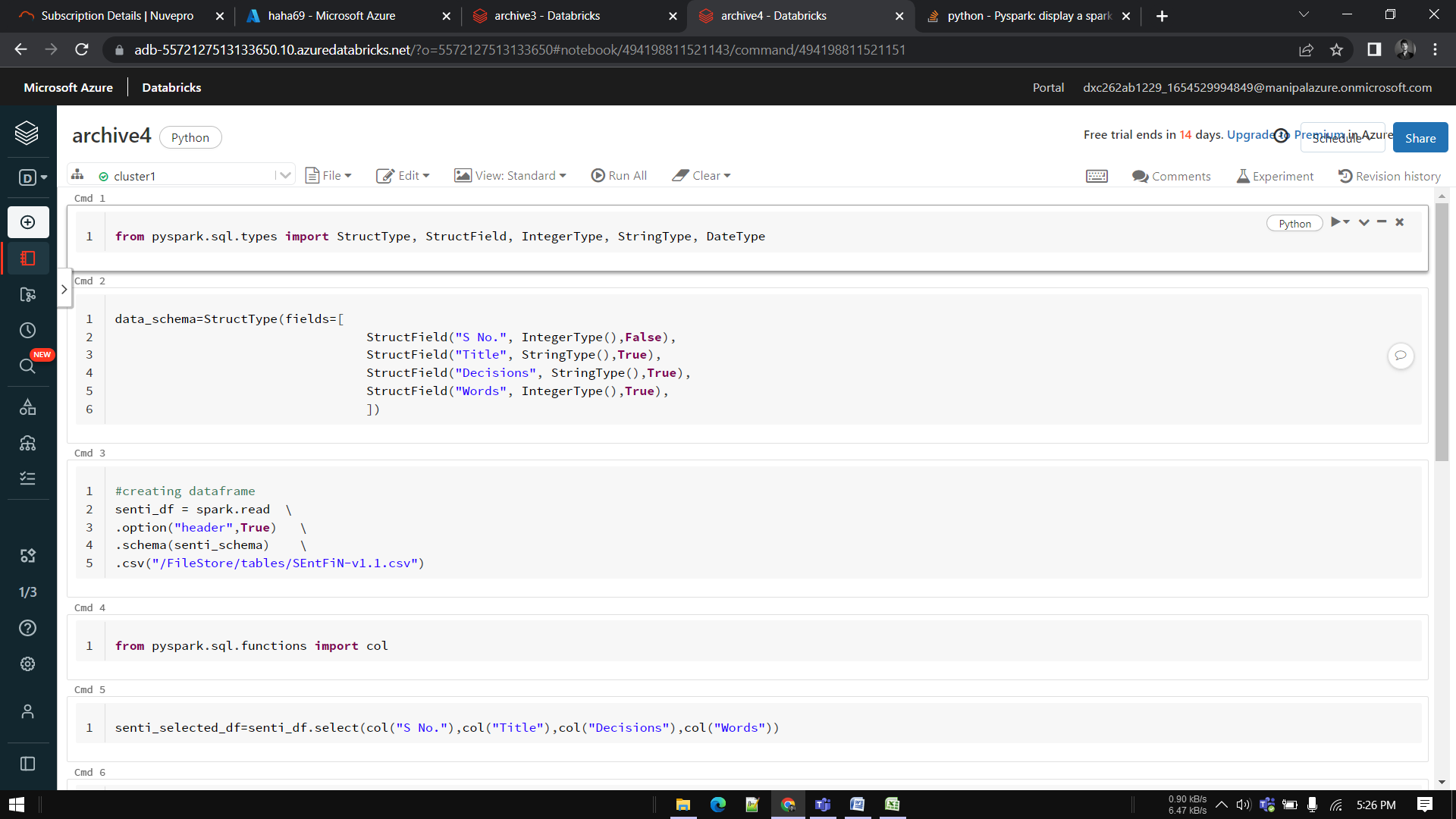
#creating dataframe

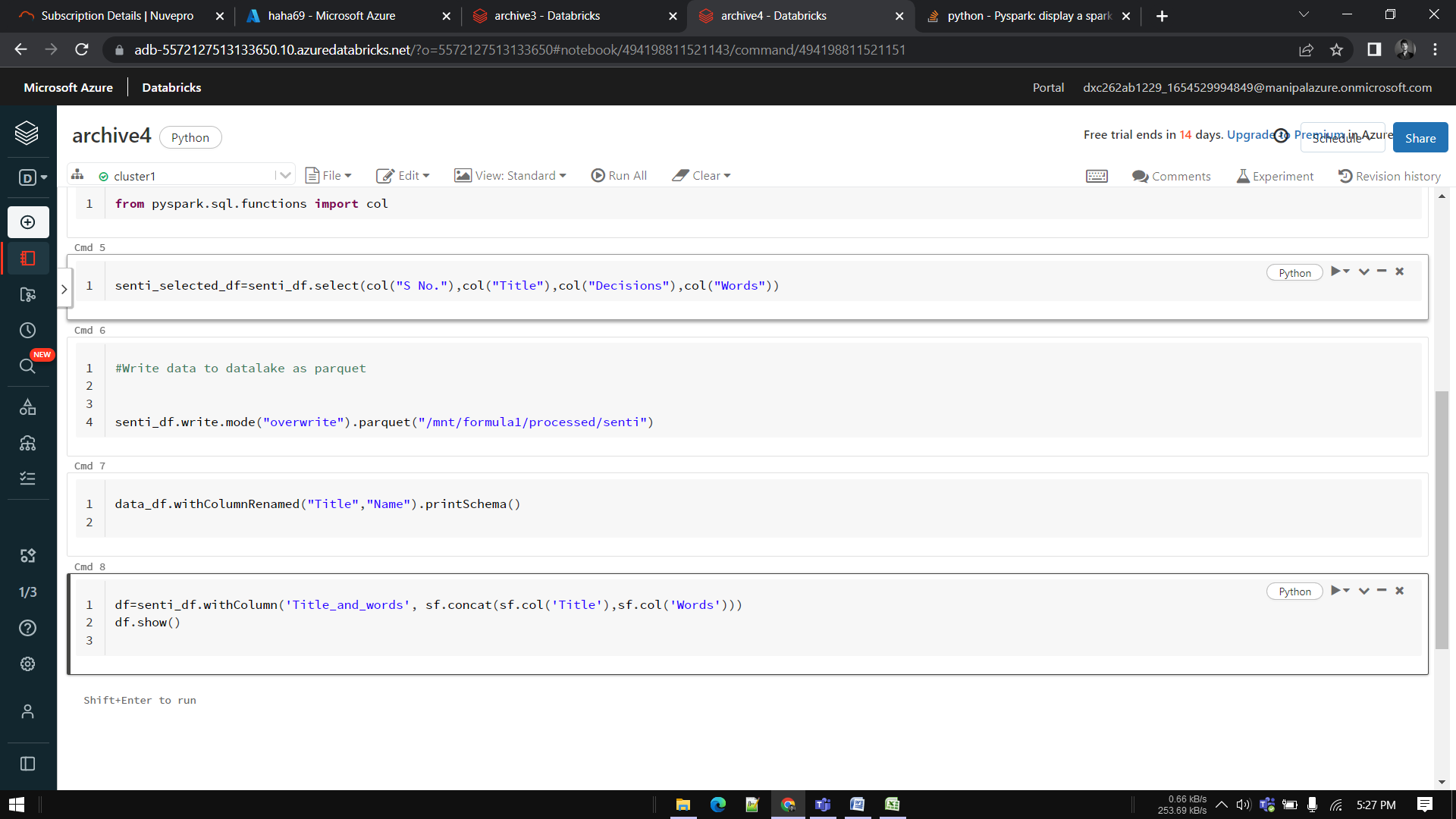
senti\_df = spark.read \

.option("header",True) \

.schema(senti\_schema) \

.csv("/FileStore/tables/SEntFiN-v1.1.csv")





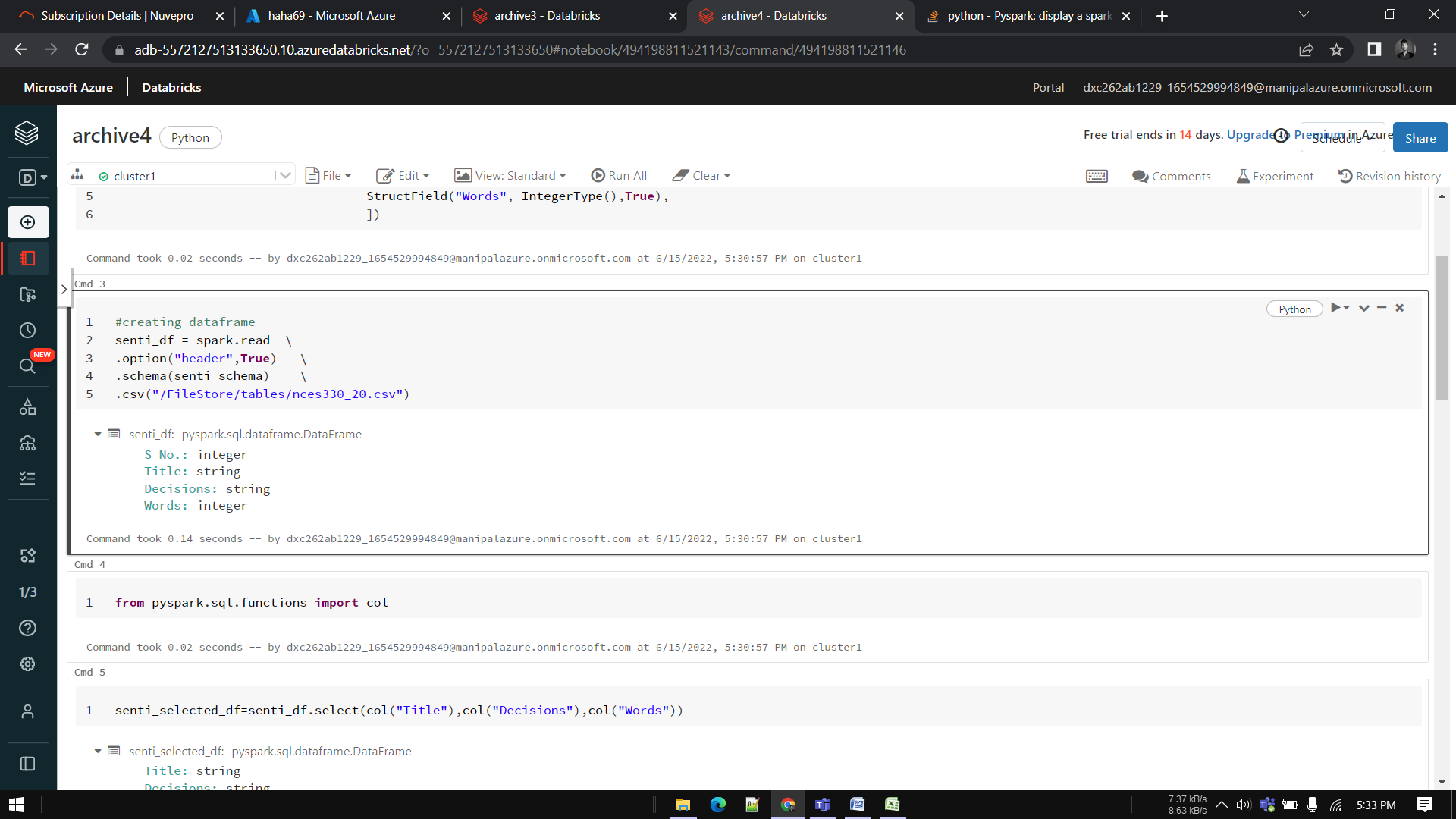
#creating dataframe

senti\_df = spark.read \

.option("header",True) \

.schema(senti\_schema) \

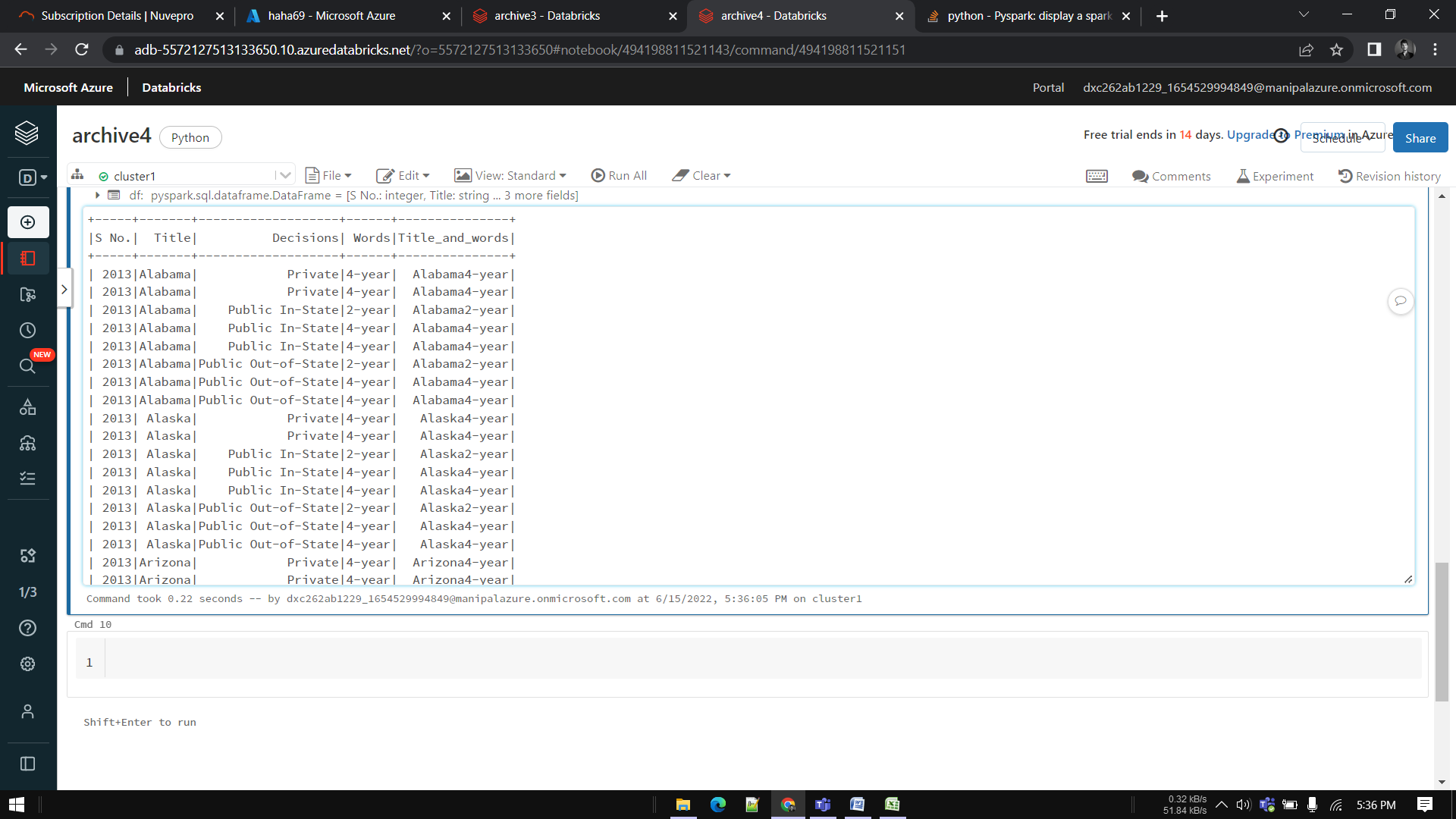
.csv("/FileStore/tables/nces330\_20.csv")



Redesigning a column—

df=senti\_df.withColumn('Title\_and\_words', sf.concat(sf.col('Title'),sf.col('Words')))

df.show()



**5.Using archive5.zip file - please ingest data into databricks DBFS path & query the data**

**redesign columns accordingly using DataFrame commands - display with notebooks accordingly**

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DateType

cancer\_schema=StructType(fields=[

StructField("Entity", StringType(),False),

StructField("Code", StringType(),True),

StructField("Year", IntegerType(),True),

StructField("Deaths", IntegerType(),True),

]

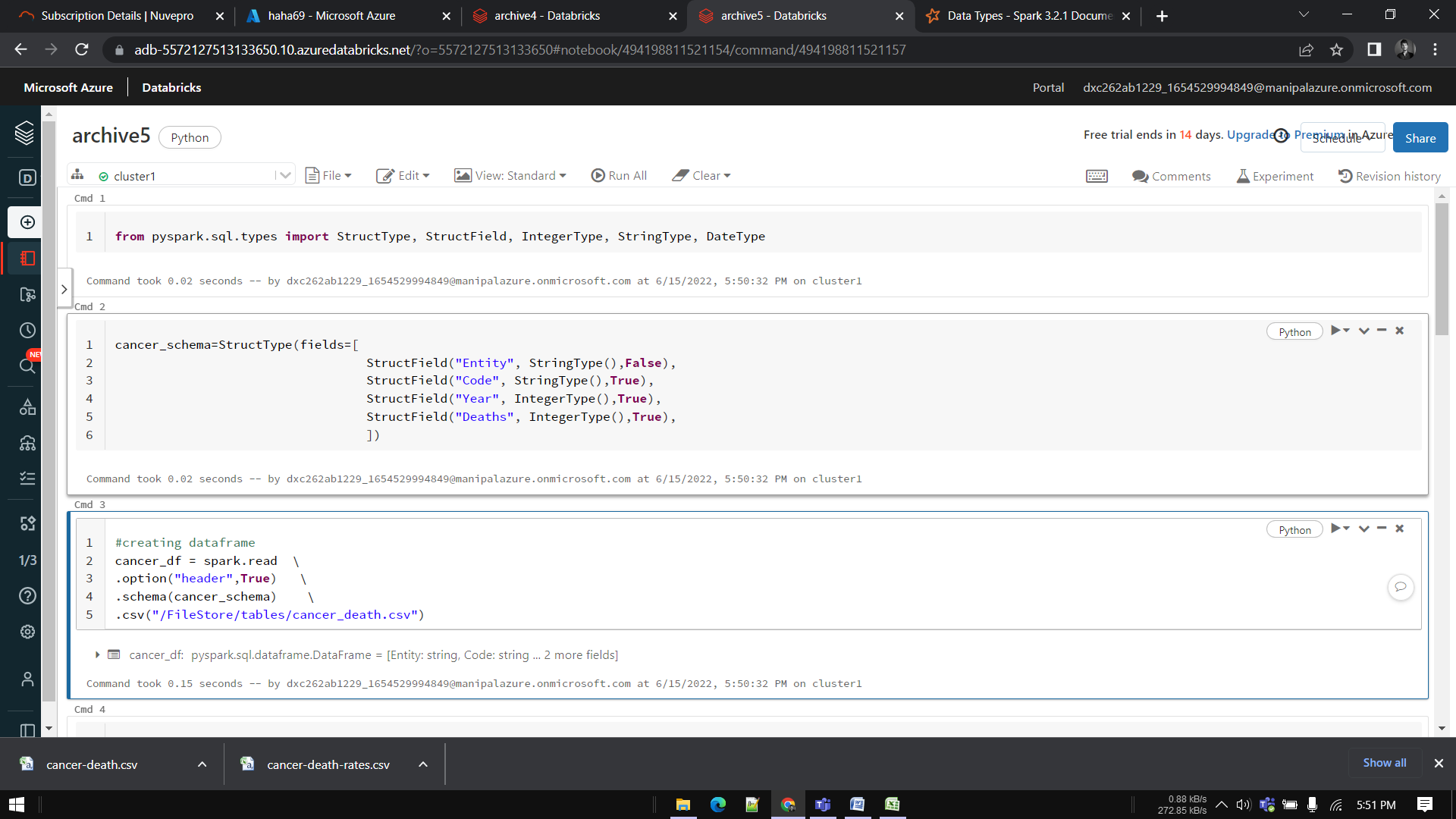
#creating dataframe

cancer\_df = spark.read \

.option("header",True) \

.schema(cancer\_schema) \

.csv("/FileStore/tables/cancer\_death.csv")



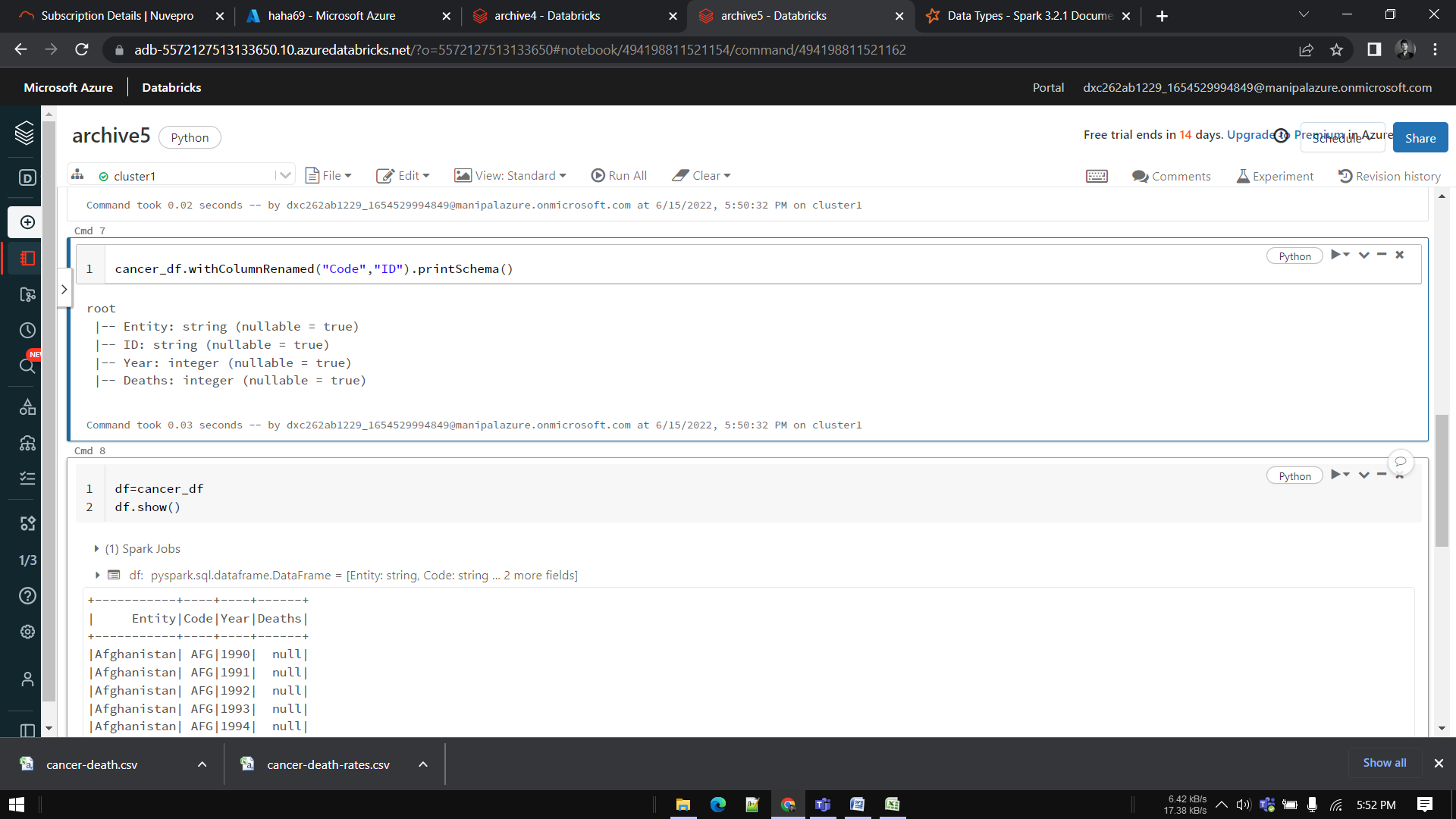
from pyspark.sql.functions import col

#Write data to datalake as parquet

cancer\_df.write.mode("overwrite").parquet("/mnt/formula1/processed/cancer")

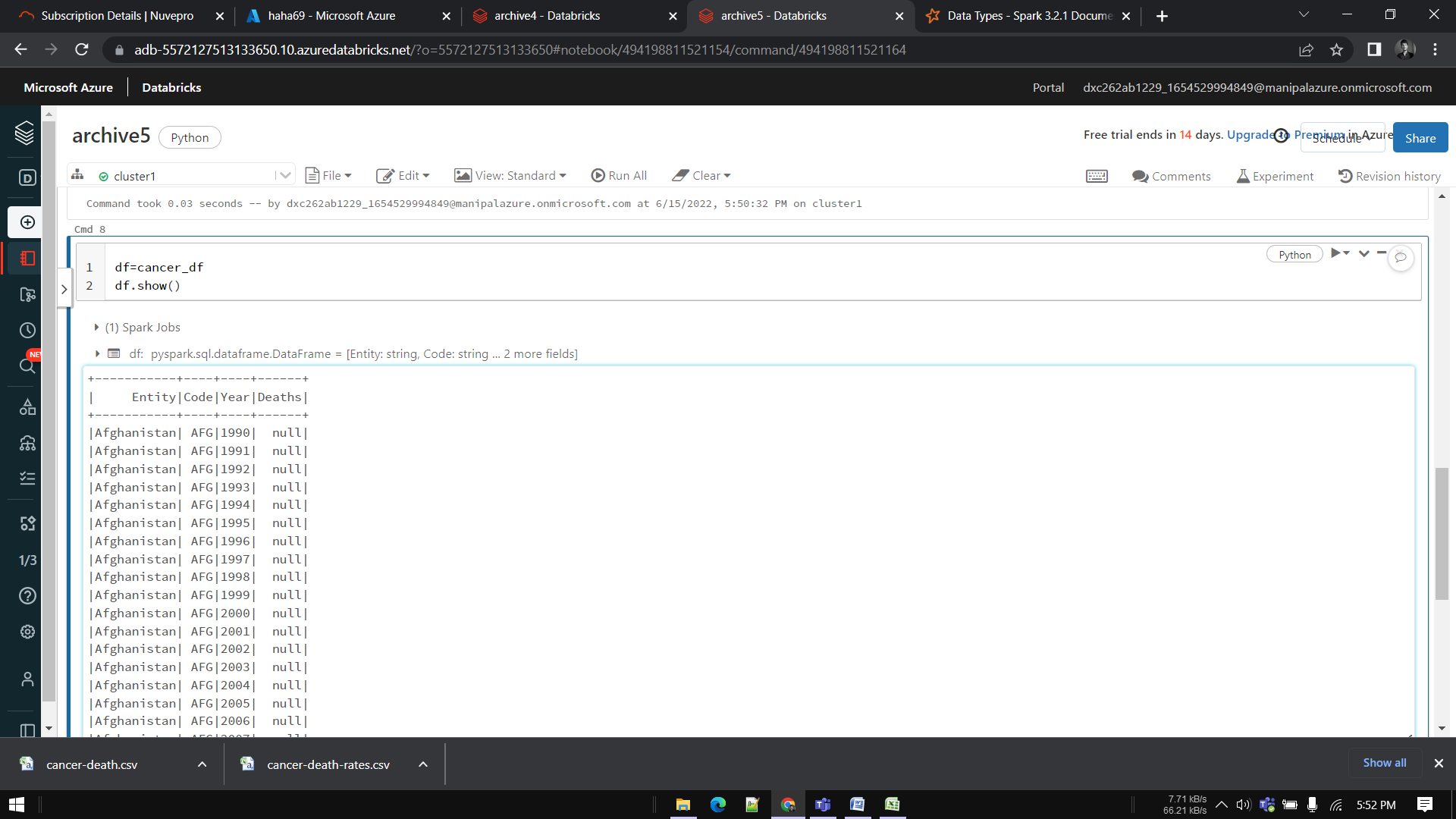
import pyspark.sql.functions as sf

cancer\_df.withColumnRenamed("Code","ID").printSchema()



df=cancer\_df

df.show()



**6.Using archive6.zip file - please ingest data into databricks DBFS path & query the data**

**redesign columns accordingly using data frame commands - display with notebooks accordingly**

from pyspark.sql.types import StructType, StructField, IntegerType, StringType, DoubleType

inflation\_schema=StructType(fields=[

StructField("country", StringType(),False),

StructField("county\_Code", StringType(),True),

StructField("Year", IntegerType(),True),

StructField("inflation", IntegerType(),True),

]

#creating dataframe

inflation\_df = spark.read \

.option("header",True) \

.schema(inflation\_schema) \

.csv("/FileStore/tables/inflation.csv")

from pyspark.sql.functions import col

#Write data to datalake as parquet

cancer\_df.write.mode("overwrite").parquet("/mnt/formula1/processed/inflation")

import pyspark.sql.functions as sf

cancer\_df.withColumnRenamed("code","ID").printSchema()

df=inflation\_df

df.show()

Redesigning a column—

df=inflation\_df.withColumn('Country\_and\_code', sf.concat(sf.col(Country),sf.col(Code)))

df.show()

