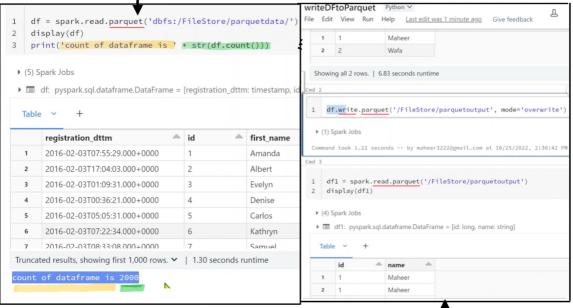
Parquet-Bead, Write / Show C) / with Column

Read parquet file data into Dataframe Using read.parquet("path") or read.format("parquet").load("path") you can read a parquet file into a PySpark DataFrame df = spark.read.parquet('dbfs:/FileStore/data/userdatal.parquet') display(df)

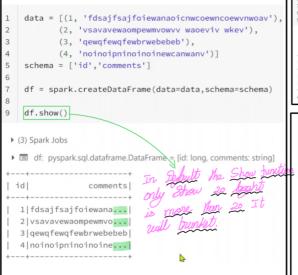


Write Dataframe into Parquet

Use the DataFrameWriter object to write PySpark DataFrame to a parquet file.

df.write.parquet('/FileStore/data/parquetoutput')

Show ()









Show()

- DataFrame show() is used to display the contents of the DataFrame in a Table Row and Column Format.
- By default, it shows only 20 Rows, and the column values are truncated at 20 characters.
- To display full column contents, we should use "truncate=False" parameter inside show(). We can also truncate column value to desired length.

```
# Show full contents of column

df.show(truncate=8)
```

· We can control rows to display as well as shown below.

```
df.show(n=1,truncate=False)
```

 We can display content vertically too as show below. df.show(truncate=False,vertical=True)

withColumn()

 PySpark withColumn() is a transformation function of DataFrame which is used to change the value, convert the datatype of an existing column, create a new column, and many more

```
from pyspark.sql.functions import col

data = [(1,'Maheer','38000'), (2,'Wafa','4800')]
schema = ['id','namer','salary']
df = spark.createDataFrame(data,schema)

df.withColumn("CopiedColumn",col("salary") - 1).show()

*Change column datatype

df1 = df.withColumn("Salary",col("salary").cast("Integer"))

df1.printSchema()
```

```
data = [(1, 'Maheer','3000'),(2, 'Wafa','4000')]
    columns = ['id','name','salary']
4
5
6
    df = spark.createDataFrame(data=data,schema=columns)
   df1 = df.withColumn(colName='salary',col=col('salary').cast('Integer'))
8
9
   df2 = df1.withColumn('salary', col('salary') * 2) ___
10
11
12
  df3 = df2.withColumn('country',lit('india'))
13 df3.show()
 (3) Spark Jobs
 ▶ ■ df: pyspark.sql.dataframe.DataFrame = [id: long, name: string ... 1 more field]
 ▶ ■ df1: pyspark.sql.dataframe.DataFrame = [id: long, name: string ... 1 more field]
 ▶ ■ df2: pyspark.sql.dataframe.DataFrame = [id: long, name: string ... 1 more field]
 ▶ ■ df3: pyspark.sql.dataframe.DataFrame = [id: long, name: string ... 2 more fields]
| id| name|salary|country|
| 1|Maheer| 6000| india|
| 2| Wafa| 8000| india|
```

-> Convert the data type to Integer

> Multiply the '<u>Ealary'</u> Column Value with '2

— Create New <u>Country</u> Column with Tralia Value in the table