Exercise 1 & 2

from pyspark.sql import SparkSession

import ipywidgets as widgets

from IPython.display import display

# Step 1: Initialize a Spark session

spark = SparkSession.builder.appName("PySpark with Widgets Example").getOrCreate()

# Step 2: Create a simple DataFrame

data = [

("John", 28, "Male", 60000),

("Jane", 32, "Female", 72000),

("Mike", 45, "Male", 84000),

("Emily", 23, "Female", 52000),

("Alex", 36, "Male", 67000)

]

df = spark.createDataFrame(data, ["name", "age", "gender", "salary"])

# Show the DataFrame

df.show()

from pyspark.sql import SparkSession

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

df\_ex=spark.read.format("csv").option("header","true").option("inferSchema","true").load(csv\_file\_path)

df\_ex.show()

df\_ex.createOrReplaceTempView("people\_temp\_view")

result1=spark.sql("select \* from people\_temp\_view where age>30")

result1.show()

result2=spark.sql("select gender,avg(Salary) as average\_salary from people\_temp\_view group by gender")

result2.show()

bonus=df\_ex.withColumn("bonus",col("Salary")\*0.1)

bonus.show()

df\_ex.write.parquet("people\_data.parquet")

from pyspark.sql import SparkSession

# Initialize Spark session

spark = SparkSession.builder.appName("Movie Data Transformations").getOrCreate()

# Load CSV file into DataFrame

file\_path = "/content/sample\_data/movie\_date.csv"

df = spark.read.csv(file\_path, header=True, inferSchema=True)

# Show the DataFrame

df.show()

sci\_fi\_movies = df.filter(df.genre == "Sci-Fi")

sci\_fi\_movies.show()

top\_rated\_movies = df.orderBy(df.rating.desc()).limit(3)

top\_rated\_movies.show()

from pyspark.sql.functions import year

movies\_after\_2010 = df.filter(year(df.date) > 2010)

movies\_after\_2010.show()

from pyspark.sql.functions import col, avg

avg\_box\_office\_by\_genre = df.groupBy("genre").agg(avg("box\_office").alias("avg\_box\_office"))

avg\_box\_office\_by\_genre.show()

df\_with\_billions = df.withColumn("box\_office\_in\_billions", col("box\_office") / 1\_000\_000\_000)

df\_with\_billions.show()

sorted\_by\_box\_office = df.orderBy(col("box\_office").desc())

sorted\_by\_box\_office.show()