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()
```