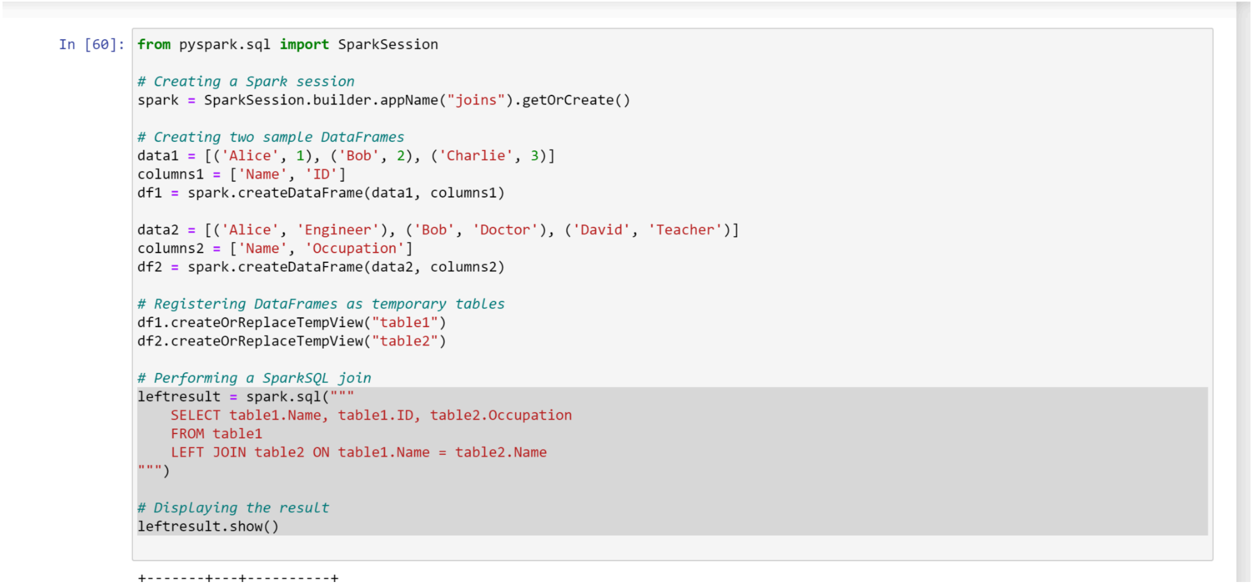
Asmita Porwal
Data Engineering
Batch-1
12/02/2024

Coding Challenge -3 Question-2

Execute Pyspark -sparksql joins

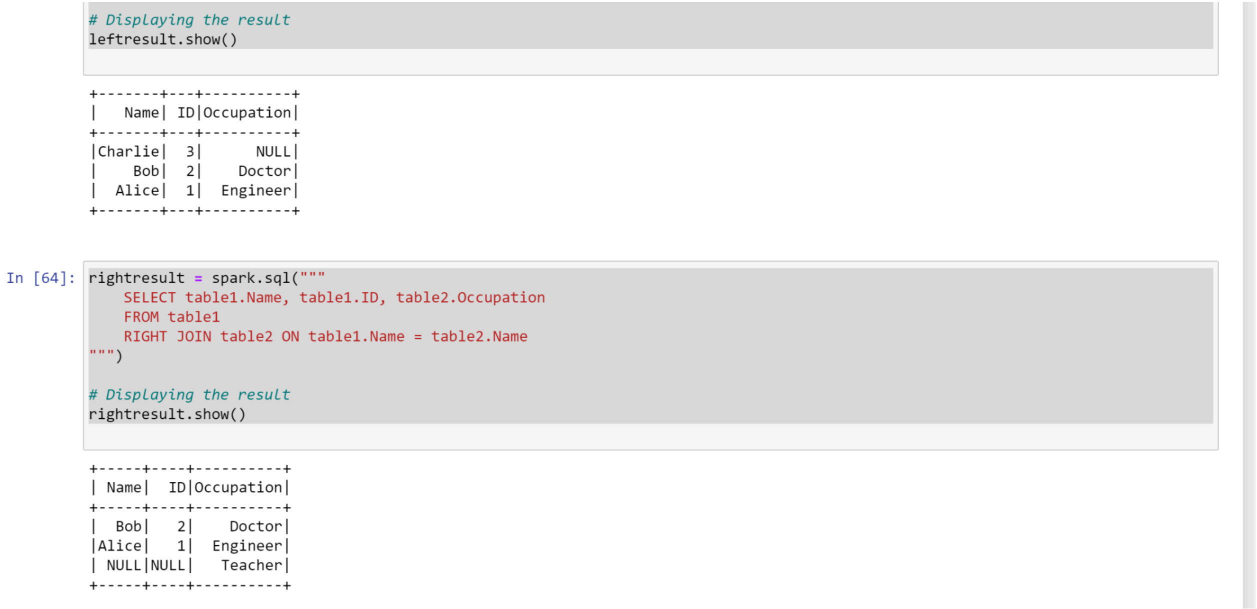
Left join



```
from pyspark.sql import SparkSession

# Creating a Spark session
spark = SparkSession.builder.appName("joins").getOrCreate()

# Creating two sample DataFrames
data1 = [('Alice', 1), ('Bob', 2), ('Charlie', 3)]
columns1 = ['Name', 'ID']
df1 = spark.createDataFrame(data1, columns1)

data2 = [('Alice', 'Engineer'), ('Bob', 'Doctor'), ('David', 'Teacher')]
columns2 = ['Name', 'Occupation']
df2 = spark.createDataFrame(data2, columns2)

# Registering DataFrames as temporary tables
df1.createOrReplaceTempView("table1")
df2.createOrReplaceTempView("table2")

# Performing a SparkSQL join
leftresult = spark.sql("""
    SELECT table1.Name, table1.ID, table2.Occupation
    FROM table1
    LEFT JOIN table2 ON table1.Name = table2.Name
""")

# Displaying the result
leftresult.show()
```

```
+-------+---+----------+
```

Right join



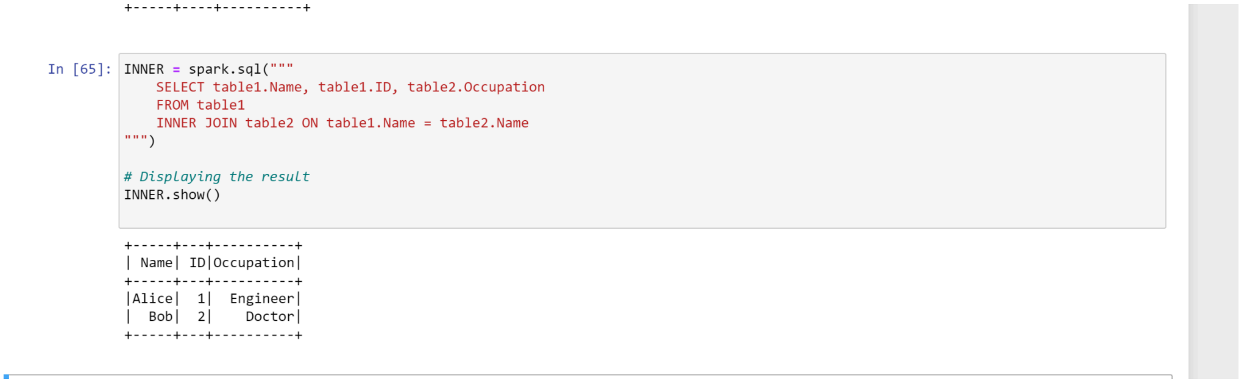
```
# Displaying the result
leftresult.show()
```

```
+-------+---+----------+
|   Name| ID|Occupation|
+-------+---+----------+
|Charlie|  3|      NULL|
|    Bob|  2|    Doctor|
|  Alice|  1|  Engineer|
+-------+---+----------+
```

```
In [64]: rightresult = spark.sql("""
             SELECT table1.Name, table1.ID, table2.Occupation
             FROM table1
             RIGHT JOIN table2 ON table1.Name = table2.Name
         """)

         # Displaying the result
         rightresult.show()
```

```
+-----+----+----------+
| Name|  ID|Occupation|
+-----+----+----------+
|  Bob|   2|    Doctor|
|Alice|   1|  Engineer|
| NULL|NULL|   Teacher|
+-----+----+----------+
```

Inner join

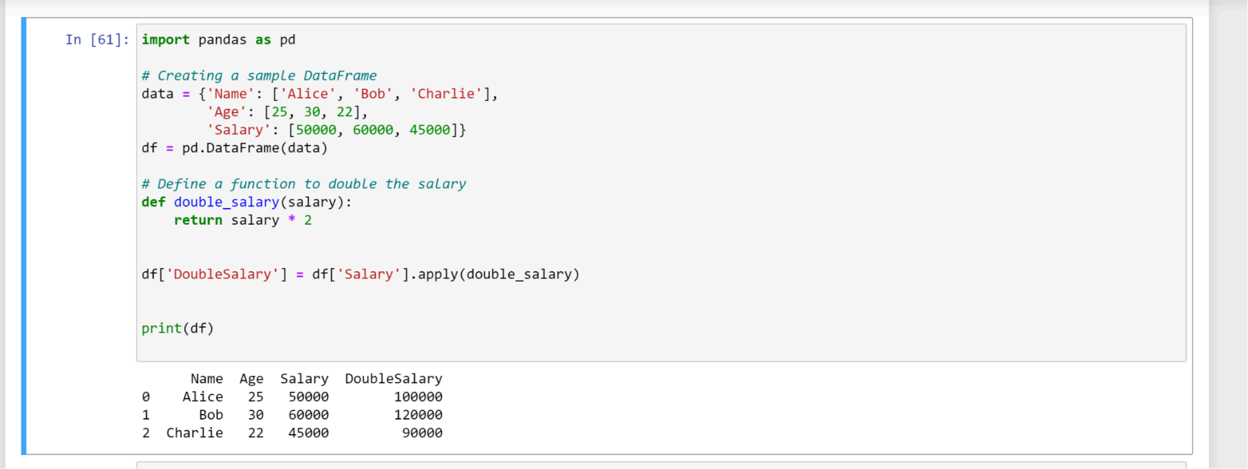


```
+-----+----+----------+
```

```
In [65]: INNER = spark.sql("""
             SELECT table1.Name, table1.ID, table2.Occupation
             FROM table1
             INNER JOIN table2 ON table1.Name = table2.Name
         """)

         # Displaying the result
         INNER.show()
```

```
+-----+---+----------+
| Name| ID|Occupation|
+-----+---+----------+
|Alice|  1|  Engineer|
|  Bob|  2|    Doctor|
+-----+---+----------+
```

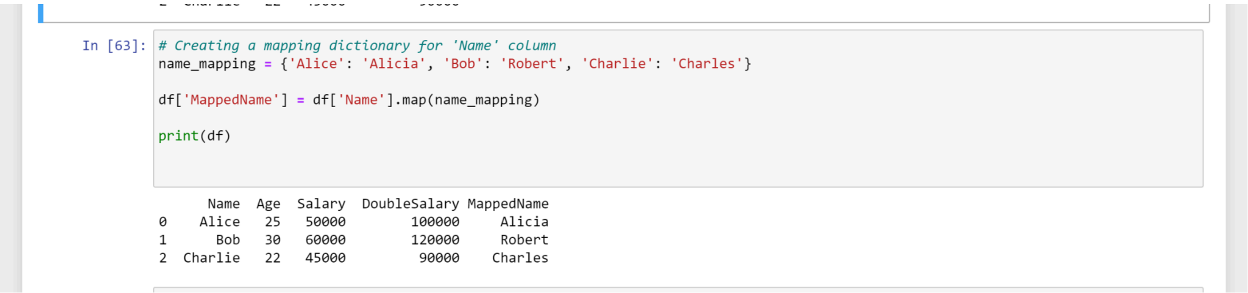
Applying Functions in a Pandas DataFrame



In [61]:

```
import pandas as pd

# Creating a sample DataFrame
data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 22],
        'Salary': [50000, 60000, 45000]}
df = pd.DataFrame(data)

# Define a function to double the salary
def double_salary(salary):
    return salary * 2


df['DoubleSalary'] = df['Salary'].apply(double_salary)


print(df)
```

```
      Name  Age  Salary  DoubleSalary
0    Alice   25   50000        100000
1      Bob   30   60000        120000
2  Charlie   22   45000         90000
```



In [63]:

```
# Creating a mapping dictionary for 'Name' column
name_mapping = {'Alice': 'Alicia', 'Bob': 'Robert', 'Charlie': 'Charles'}

df['MappedName'] = df['Name'].map(name_mapping)

print(df)
```

```
      Name  Age  Salary  DoubleSalary MappedName
0    Alice   25   50000        100000     Alicia
1      Bob   30   60000        120000     Robert
2  Charlie   22   45000         90000    Charles
```