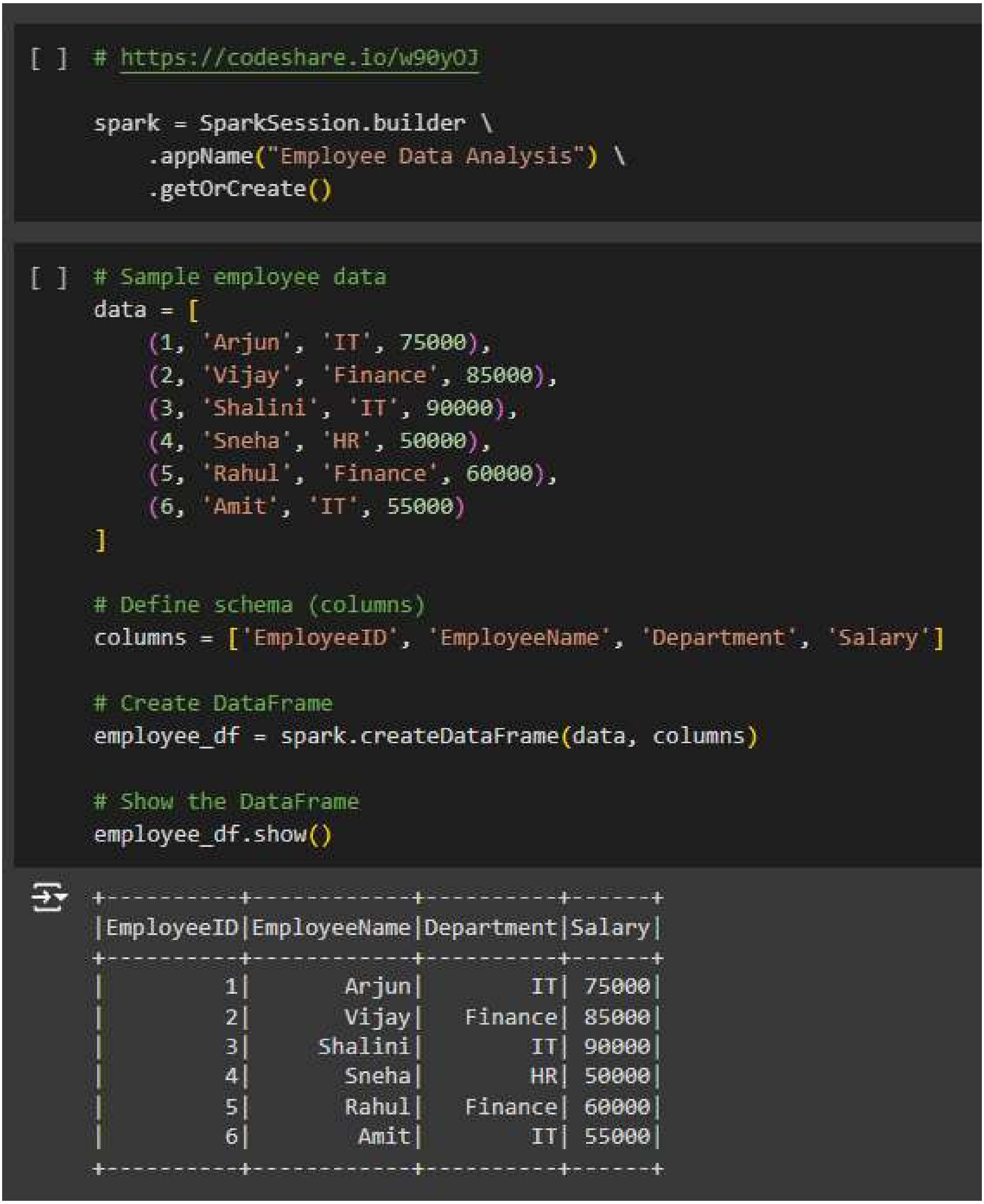
PySpark HandsOn-1 4th Sep

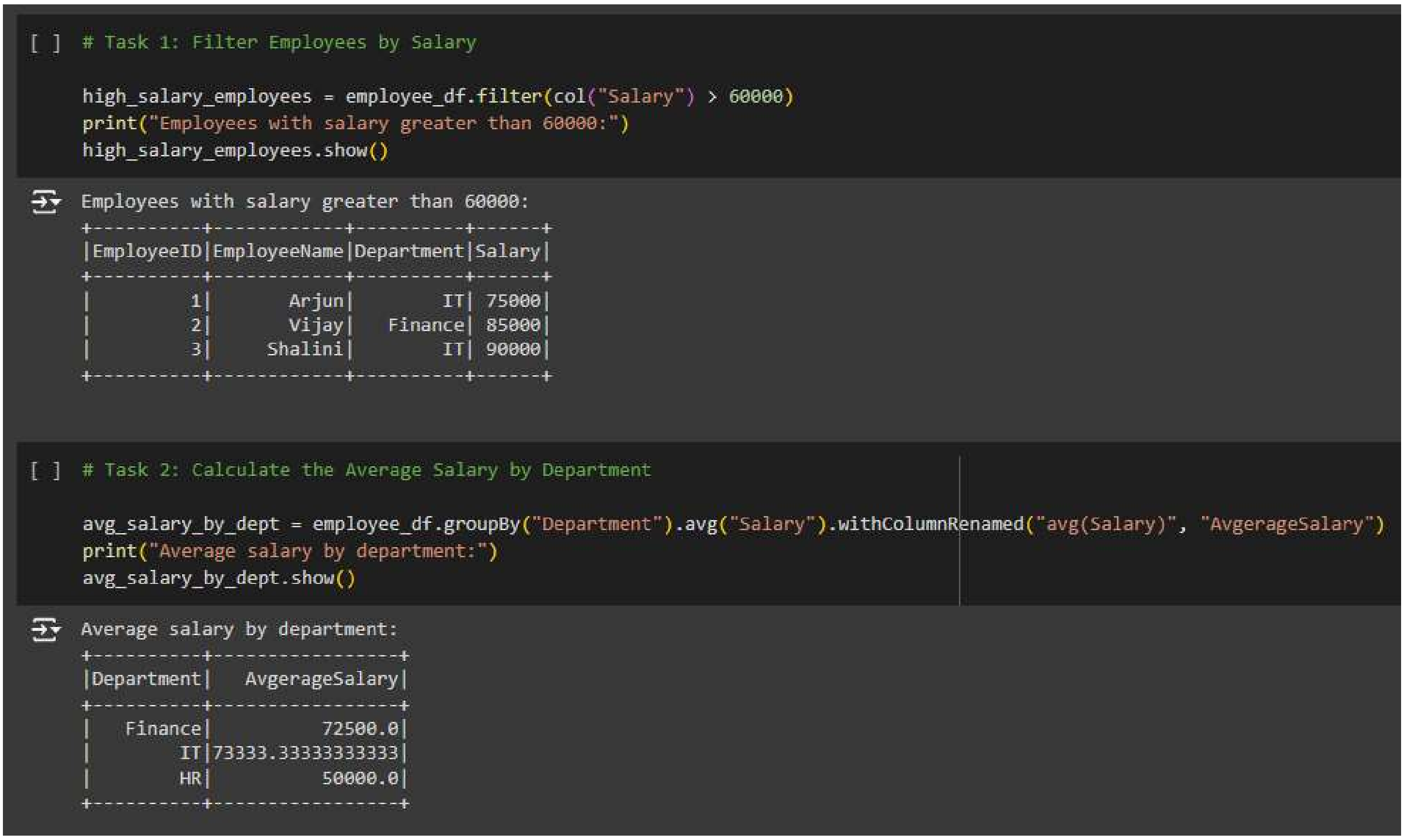
SaiPrabath Chowdary S



```
[ ]  # https://codeshare.io/w90yOJ

     spark = SparkSession.builder \
         .appName("Employee Data Analysis") \
         .getOrCreate()
```

```
[ ]  # Sample employee data
     data = [
         (1, 'Arjun', 'IT', 75000),
         (2, 'Vijay', 'Finance', 85000),
         (3, 'Shalini', 'IT', 90000),
         (4, 'Sneha', 'HR', 50000),
         (5, 'Rahul', 'Finance', 60000),
         (6, 'Amit', 'IT', 55000)
     ]

     # Define schema (columns)
     columns = ['EmployeeID', 'EmployeeName', 'Department', 'Salary']

     # Create DataFrame
     employee_df = spark.createDataFrame(data, columns)

     # Show the DataFrame
     employee_df.show()
```

```
+----------+------------+----------+------+
|EmployeeID|EmployeeName|Department|Salary|
+----------+------------+----------+------+
|         1|       Arjun|        IT| 75000|
|         2|       Vijay|   Finance| 85000|
|         3|     Shalini|        IT| 90000|
|         4|       Sneha|        HR| 50000|
|         5|       Rahul|   Finance| 60000|
|         6|        Amit|        IT| 55000|
+----------+------------+----------+------+
```

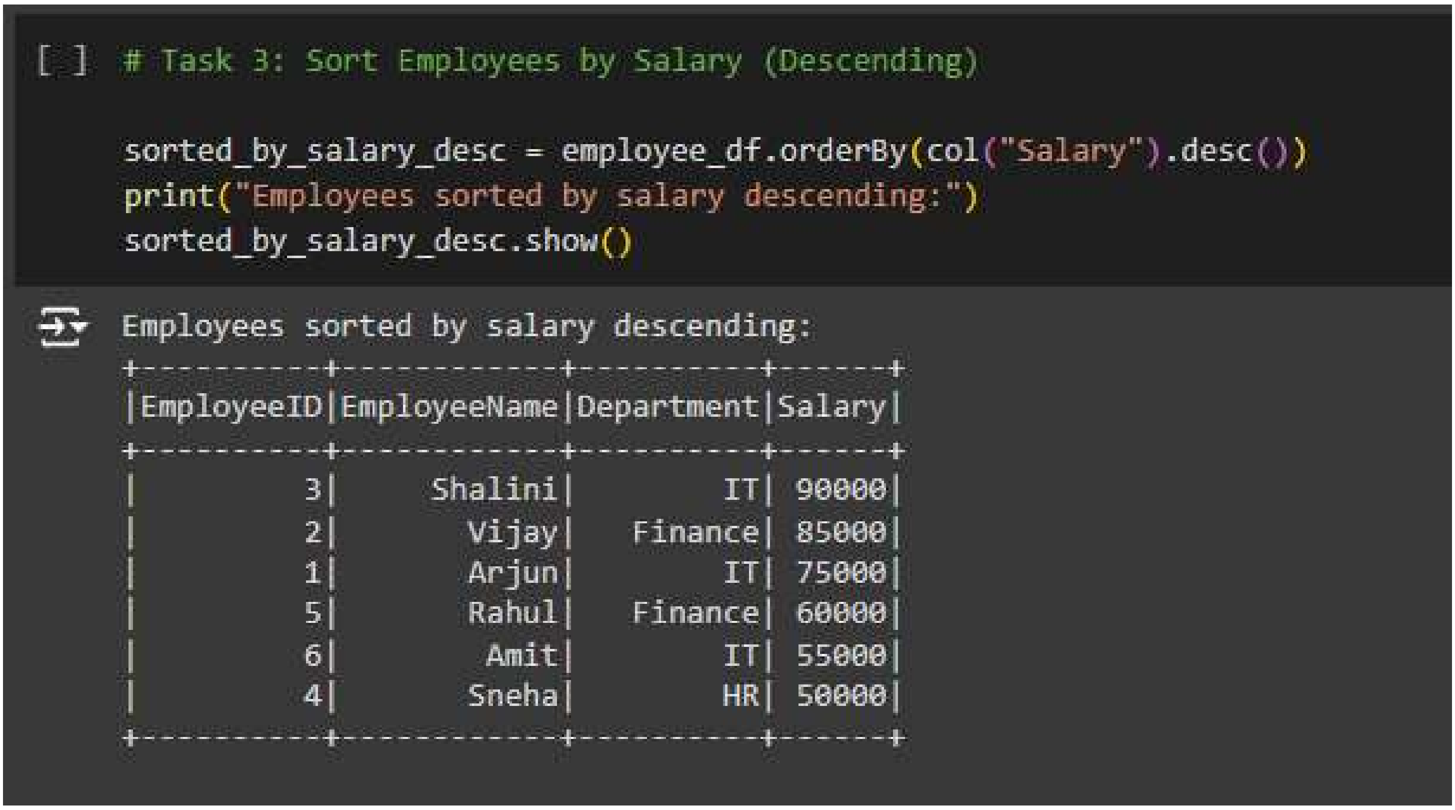


```
# Task 1: Filter Employees by Salary

high_salary_employees = employee_df.filter(col("Salary") > 60000)
print("Employees with salary greater than 60000:")
high_salary_employees.show()
```

```
Employees with salary greater than 60000:
+----------+------------+----------+------+
|EmployeeID|EmployeeName|Department|Salary|
+----------+------------+----------+------+
|         1|       Arjun|        IT| 75000|
|         2|       Vijay|   Finance| 85000|
|         3|     Shalini|        IT| 90000|
+----------+------------+----------+------+
```

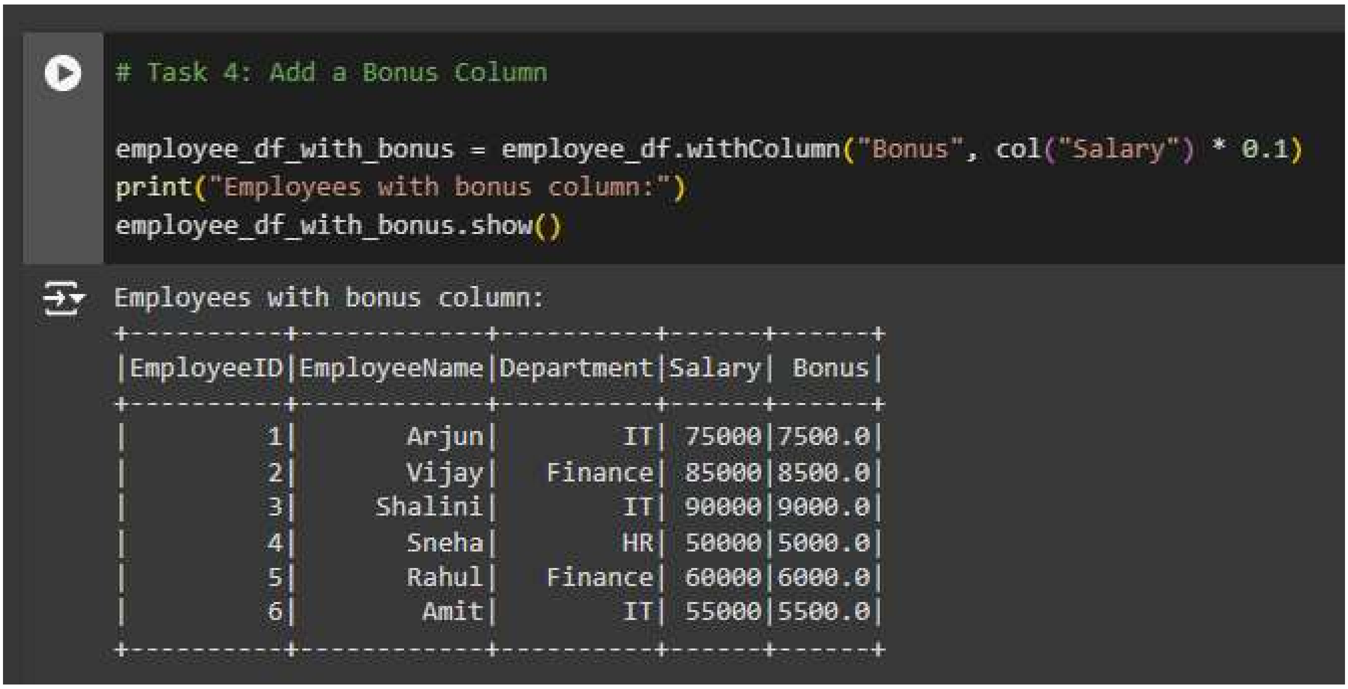
```
# Task 2: Calculate the Average Salary by Department

avg_salary_by_dept = employee_df.groupBy("Department").avg("Salary").withColumnRenamed("avg(Salary)", "AvgerageSalary")
print("Average salary by department:")
avg_salary_by_dept.show()
```

```
Average salary by department:
+----------+-----------------+
|Department|   AvgerageSalary|
+----------+-----------------+
|   Finance|          72500.0|
|        IT|73333.33333333333|
|        HR|          50000.0|
+----------+-----------------+
```



```
# Task 3: Sort Employees by Salary (Descending)

sorted_by_salary_desc = employee_df.orderBy(col("Salary").desc())
print("Employees sorted by salary descending:")
sorted_by_salary_desc.show()
```

```
Employees sorted by salary descending:
+----------+------------+----------+------+
|EmployeeID|EmployeeName|Department|Salary|
+----------+------------+----------+------+
|         3|     Shalini|        IT| 90000|
|         2|       Vijay|   Finance| 85000|
|         1|       Arjun|        IT| 75000|
|         5|       Rahul|   Finance| 60000|
|         6|        Amit|        IT| 55000|
|         4|       Sneha|        HR| 50000|
+----------+------------+----------+------+
```



```
# Task 4: Add a Bonus Column

employee_df_with_bonus = employee_df.withColumn("Bonus", col("Salary") * 0.1)
print("Employees with bonus column:")
employee_df_with_bonus.show()
```

```
Employees with bonus column:
+----------+------------+----------+------+------+
|EmployeeID|EmployeeName|Department|Salary| Bonus|
+----------+------------+----------+------+------+
|         1|       Arjun|        IT| 75000|7500.0|
|         2|       Vijay|   Finance| 85000|8500.0|
|         3|     Shalini|        IT| 90000|9000.0|
|         4|       Sneha|        HR| 50000|5000.0|
|         5|       Rahul|   Finance| 60000|6000.0|
|         6|        Amit|        IT| 55000|5500.0|
+----------+------------+----------+------+------+
```