Assignment 1: Working with CSV Data (employee_data.csv)

Tasks:

1. Load the CSV data:

```
2 minutes ago (1s)                                                    6

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, to_date, avg, count

# Create a SparkSession
spark = SparkSession.builder.appName("EmployeeDataAnalysis").getOrCreate()
# Convert into dbfs
dbutils.fs.cp("file:/Workspace/Users/azuser2130_mml.local@techademy.com/Employees_data.csv", "dbfs:/FileStore/Employees_data.csv")
# Load CSV data
df = spark.read.format("csv").option("header","true").load("dbfs:/FileStore/Employees_data.csv")
# Display the first 10 rows
df.show(10)
```

Output Terminal Debug Console

df: pyspark.sql.dataframe.DataFrame = [EmployeeID: string, Name: string ... 3 more fields]

EmployeeID	Name	Department	JoiningDate	Salary
1001	John Doe	HR	2021-01-15	55000
1002	Jane Smith	IT	2020-03-10	62000
1003	Emily Johnson	Finance	2019-07-01	70000
1004	Michael Brown	HR	2018-12-22	54000
1005	David Wilson	IT	2021-06-25	58000
1006	Linda Davis	Finance	2020-11-15	67000
1007	James Miller	IT	2019-08-14	65000
1008	Barbara Moore	HR	2021-03-29	53000

2. Data Cleaning:

```
Just now (<1s)                                                        6

# Remove rows where Salary is less than 55,000
df_cleaned = df.filter(col('Salary') >= 55000)
df_cleaned.show()

# Filter employees who joined after 2020
df_cleaned = df_cleaned.filter(to_date(df_cleaned['JoiningDate']) >= '2020-01-01')
df_cleaned.show()
```

(2) Spark Jobs

df_cleaned: pyspark.sql.dataframe.DataFrame = [EmployeeID: string, Name: string ... 3 more fields]

EmployeeID	Name	Department	JoiningDate	Salary
1001	John Doe	HR	2021-01-15	55000
1002	Jane Smith	IT	2020-03-10	62000
1003	Emily Johnson	Finance	2019-07-01	70000
1005	David Wilson	IT	2021-06-25	58000
1006	Linda Davis	Finance	2020-11-15	67000
1007	James Miller	IT	2019-08-14	65000

EmployeeID	Name	Department	JoiningDate	Salary
1001	John Doe	HR	2021-01-15	55000
1002	Jane Smith	IT	2020-03-10	62000
1005	David Wilson	IT	2021-06-25	58000
1006	Linda Davis	Finance	2020-11-15	67000

3. Data Aggregation:

```
Just now (1s)                                                    6

# Find the average salary by Department
average_salary_by_department = df_cleaned.groupby('Department').agg(avg('Salary').alias('AverageSalary'))
print("Average Salary by Department:\n")
average_salary_by_department.show()

# Count the number of employees in each Department
employee_count_by_department = df_cleaned.groupby('Department').count()
print("Employee Count by Department:\n")
employee_count_by_department.show()
```

Output Terminal Debug Console

```
Average Salary by Department:

+----------+-------------+
|Department|AverageSalary|
+----------+-------------+
|        HR|      55000.0|
|   Finance|      67000.0|
|        IT|      60000.0|
+----------+-------------+

Employee Count by Department:

+----------+-----+
|Department|count|
+----------+-----+
|        HR|    1|
|   Finance|    1|
|        IT|    2|
```

4. Write the Data to CSV:

Save the cleaned data (from the previous steps) to a new CSV file.

```
Just now (1s)                                                    6

# Save the cleaned data to a new CSV file
df_cleaned.write.mode("overwrite").csv("dbfs:/path/to/cleaned_employee_data.csv", header=True)

▸ (1) Spark Jobs
```