

Azure DataBricks Assignment – 12th September

SaiPrabath Chowdary S

Assignment 1

```
11:36 AM (3s) 2: loading csv

# Loading Data
data = "file:/Workspace/Shared/employee_data.csv"
employee_df = spark.read.csv(data, header=True, inferSchema=True)
employee_df.show(10)

(3) Spark Jobs
  employee_df: pyspark.sql.dataframe.DataFrame = [EmployeeID: integer, 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|
+-----+-----+-----+-----+-----+
```

```
11:40 AM (1s) 3: cleaning data

# Remove rows where the Salary is less than 55,000
cleaned_df = employee_df.filter(employee_df.Salary >= 55000)
cleaned_df.show()

# Filter the employees who joined after the year 2020
from pyspark.sql.functions import year
filtered_df = employee_df.filter(year(cleaned_df.JoiningDate) > 2020)
filtered_df.show()

(2) Spark Jobs
  cleaned_df: pyspark.sql.dataframe.DataFrame = [EmployeeID: integer, Name: string ... 3 more fields]
  filtered_df: pyspark.sql.dataframe.DataFrame = [EmployeeID: integer, 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|
|    1005| David Wilson|      IT| 2021-06-25| 58000|
|    1008|Barbara Moore|      HR| 2021-03-29| 53000|
+-----+-----+-----+-----+-----+
```



11:40 AM (1s)

4: Data Aggregation

```
from pyspark.sql.functions import avg, count

# Find the average salary by Department
avg_salary_df = employee_df.groupBy("Department").agg(avg("Salary").alias("Average Salary"))
avg_salary_df.show()

# Count the number of employees in each Department
employee_count_df = employee_df.groupBy("Department").agg(count("EmployeeID").alias("Employee Count"))
employee_count_df.show()
```

▶ (4) Spark Jobs

- ▶ avg_salary_df: pyspark.sql.dataframe.DataFrame = [Department: string, Average Salary: double]
- ▶ employee_count_df: pyspark.sql.dataframe.DataFrame = [Department: string, Employee Count: long]

```
+-----+-----+
|Department|  Average Salary|
+-----+-----+
|      HR|         54000.0|
|  Finance|         68500.0|
|      IT|61666.666666666664|
+-----+-----+
```

```
+-----+-----+
|Department|Employee Count|
+-----+-----+
|      HR|             3|
|  Finance|             2|
|      IT|             3|
+-----+-----+
```



12:50 PM (3s)

5: loading data

Python



```
# Save the cleaned data to a new CSV file
filtered_df.write.csv("file:/Workspace/Shared/filtered_employee_data.csv", header=True)
```

▶ (1) Spark Jobs