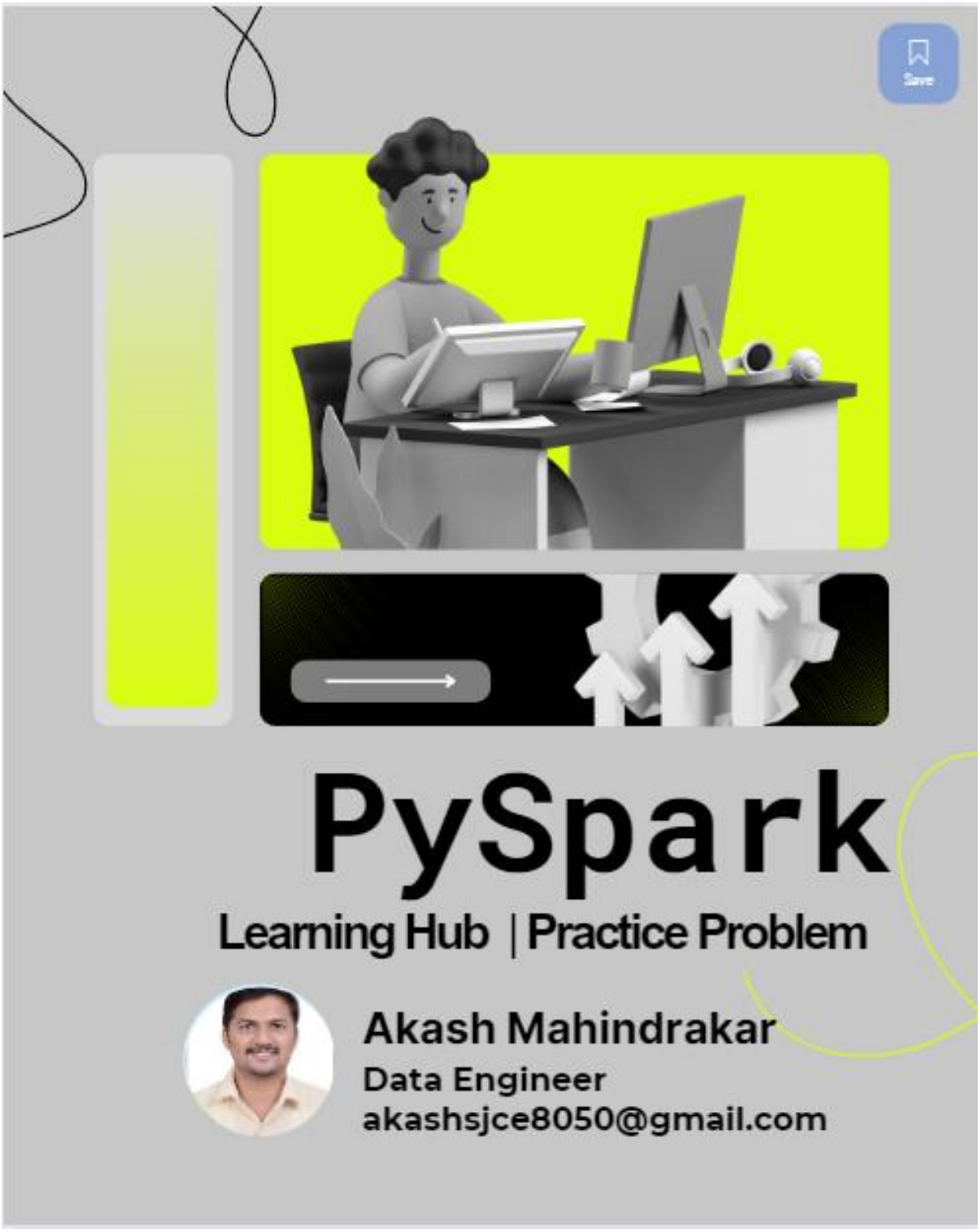
PYSPARK LEARNING HUB : DAY - 10



Step - 1 : Problem Statement

10_Employee Bonus

Write a solution to report the name and bonus amount of each employee with a bonus less than 1000.

Return the result table in any order

Difficult Level : EASY

DataFrame:

```
# Define the schema for the "Employee"
employee_schema = StructType([
        StructField("empId", IntegerType(), True),
        StructField("name", StringType(), True),
        StructField("supervisor", IntegerType(), True),
        StructField("salary", IntegerType(), True)
])

# Define data for the "Employee"
employee_data = [
        (3, 'Brad', None, 4000),
        (1, 'John', 3, 1000),
        (2, 'Dan', 3, 2000),
        (4, 'Thomas', 3, 4000)
]

# Define the schema for the "Bonus"
bonus_schema = StructType([
        StructField("empId", IntegerType(), True),
        StructField("bonus", IntegerType(), True)
])
```

```
# Define data for the "Bonus"
bonus_data = [
        (2, 500),
        (4, 2000)
]
```

Step - 2 : Identifying The Input Data And Expected Output

INPUT

INPUT-1 EMPLOYEE			
EMPID	NAME	SUPERVISOR	SALARY
3	Brad		4,000
1	John	3	1,000
2	Dan	3	2,000
4	Thomas	3	4,000

INPUT-2 BONUS	
EMPID	BONUS
2	500
4	2,000

OUTPUT

OUTPUT	
NAME	BONUS
Brad	
John	
Dan	500

Step - 3 : Writing the pyspark code to solve

```
# Creating Spark Session
from pyspark.sql import SparkSession
from pyspark.sql.types import
StructType,StructField,IntegerType,StringType

#creating spark session
spark = SparkSession. \
builder. \
config('spark.shuffle.useOldFetchProtocol', 'true'). \
config('spark.ui.port','0'). \
config("spark.sql.warehouse.dir", "/user/itv008042/warehouse"). \
enableHiveSupport(). \
master('yarn'). \
getOrCreate()

# Define the schema for the "Employee"
employee_schema = StructType([
        StructField("empId", IntegerType(), True),
        StructField("name", StringType(), True),
        StructField("supervisor", IntegerType(), True),
        StructField("salary", IntegerType(), True)
])

# Define data for the "Employee"
employee_data = [
        (3, 'Brad', None, 4000),
        (1, 'John', 3, 1000),
        (2, 'Dan', 3, 2000),
        (4, 'Thomas', 3, 4000)
]


# Define the schema for the "Bonus"
```

```
bonus_schema = StructType([
        StructField("empId", IntegerType(), True),
        StructField("bonus", IntegerType(), True)
])

# Define data for the "Bonus"
bonus_data = [
        (2, 500),
        (4, 2000)
]

# Create a PySpark DataFrame

emp_df =
spark.createDataFrame(employee_data,employee_schema)
bonus_df = spark.createDataFrame(bonus_data,bonus_schema)
emp_df.show()
bonus_df.show()
```

```
+-----+------+----------+------+
|empId|  name|supervisor|salary|
+-----+------+----------+------+
|    3|  Brad|      null|  4000|
|    1|  John|         3|  1000|
|    2|   Dan|         3|  2000|
|    4|Thomas|         3|  4000|
+-----+------+----------+------+

+-----+-----+
|empId|bonus|
+-----+-----+
|    2|  500|
|    4| 2000|
+-----+-----+
```

```
join_df=emp_df.join(bonus_df,emp_df.empId==bonus_df.empId,"left")
join_df.show()
```

```
+-----+------+----------+------+-----+-----+
|empId|  name|supervisor|salary|empId|bonus|
+-----+------+----------+------+-----+-----+
|    1|  John|         3|  1000| null| null|
|    3|  Brad|      null|  4000| null| null|
|    4|Thomas|         3|  4000|    4| 2000|
|    2|   Dan|         3|  2000|    2|  500|
+-----+------+----------+------+-----+-----+
```

```
join_df.filter( (join_df.bonus < 1000) | col("bonus").isNull() ).show()
```

```
+-----+----+----------+------+-----+-----+
|empId|name|supervisor|salary|empId|bonus|
+-----+----+----------+------+-----+-----+
|    1|John|         3|  1000| null| null|
|    3|Brad|      null|  4000| null| null|
|    2| Dan|         3|  2000|    2|  500|
+-----+----+----------+------+-----+-----+
```

PYSPARK LEARNING HUB : DAY - 10

Save

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

Comment

SHARE YOUR THOUGHTS IN COMMENT BELOW

Share