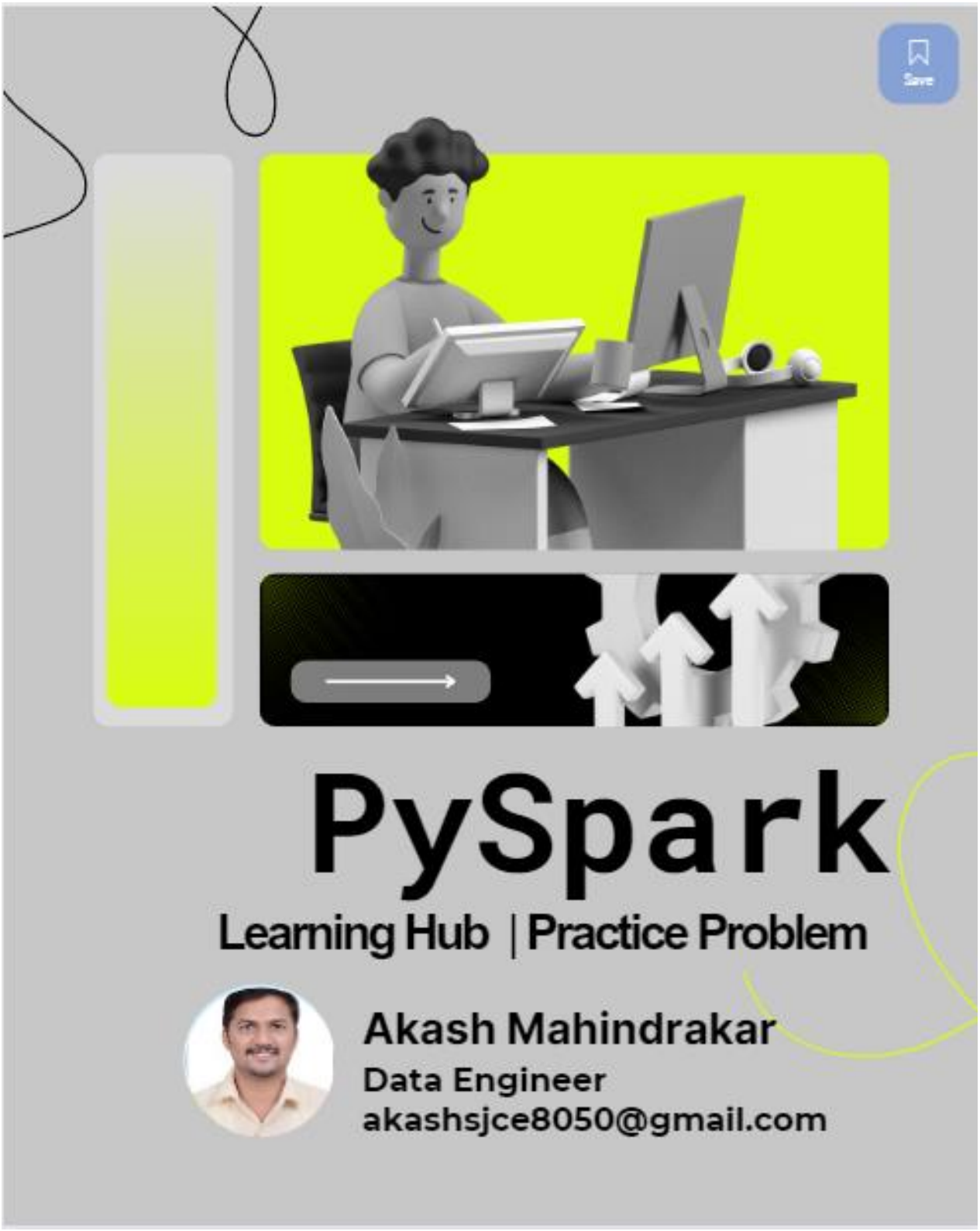
PYSPARK LEARNING HUB : DAY - 4



Step - 1 : Problem Statement

04_Employees Earning More Than Their Managers

Write a Pyspark program to find Employees Earning More Than Their Managers

Difficult Level : EASY

DataFrame:

```
# Define the schema for the "employees"
employees_schema = StructType([
        StructField("id", IntegerType(), True),
        StructField("name", StringType(), True),
        StructField("salary", IntegerType(), True),
        StructField("managerId", IntegerType(), True)
])

# Define data for the "employees"
employees_data = [
        (1, 'Joe', 70000, 3),
        (2, 'Henry', 80000, 4),
        (3, 'Sam', 60000, None),
        (4, 'Max', 90000, None)
]
```

Step - 2 : Identifying The Input Data And Expected Output

INPUT

INPUT			
ID	NAME	SALARY	MANAGERID
1	Joe	70,000	3
2	Henry	80,000	4
3	Sam	60,000	
4	Max	90,000	

OUTPUT

OUTPUT
NAME
Joe

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
from pyspark.sql.functions import when
from pyspark.sql import functions as F
from pyspark.sql.window import Window

#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 "employees"
employees_schema = StructType([
        StructField("id", IntegerType(), True),
        StructField("name", StringType(), True),
        StructField("salary", IntegerType(), True),
        StructField("managerId", IntegerType(), True)
])

# Define data for the "employees"
employees_data = [
        (1, 'Joe', 70000, 3),
        (2, 'Henry', 80000, 4),
        (3, 'Sam', 60000, None),
        (4, 'Max', 90000, None)
]
```

```
# Create a PySpark DataFrame
emp_df=spark.createDataFrame(employees_data,employees_schema)
emp_df.show()


 emp_df1 = emp_df.alias("e1")
 emp_df2 = emp_df.alias("e2")

 self_joined_df = emp_df1.join(emp_df2, col("e1.id") ==
 col("e2.managerId"),
 "inner")            .select(col("e2.name"),col("e2.salary"),col("e1.salary").alias("msal"))


self_joined_df.filter(self_joined_df.salary>self_joined_df.msal).select("name").show()
```

```
+----+
|name|
+----+
| Joe|
+----+
```

Save

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

Comment

SHARE YOUR THOUGHTS IN COMMENT BELOW

Share