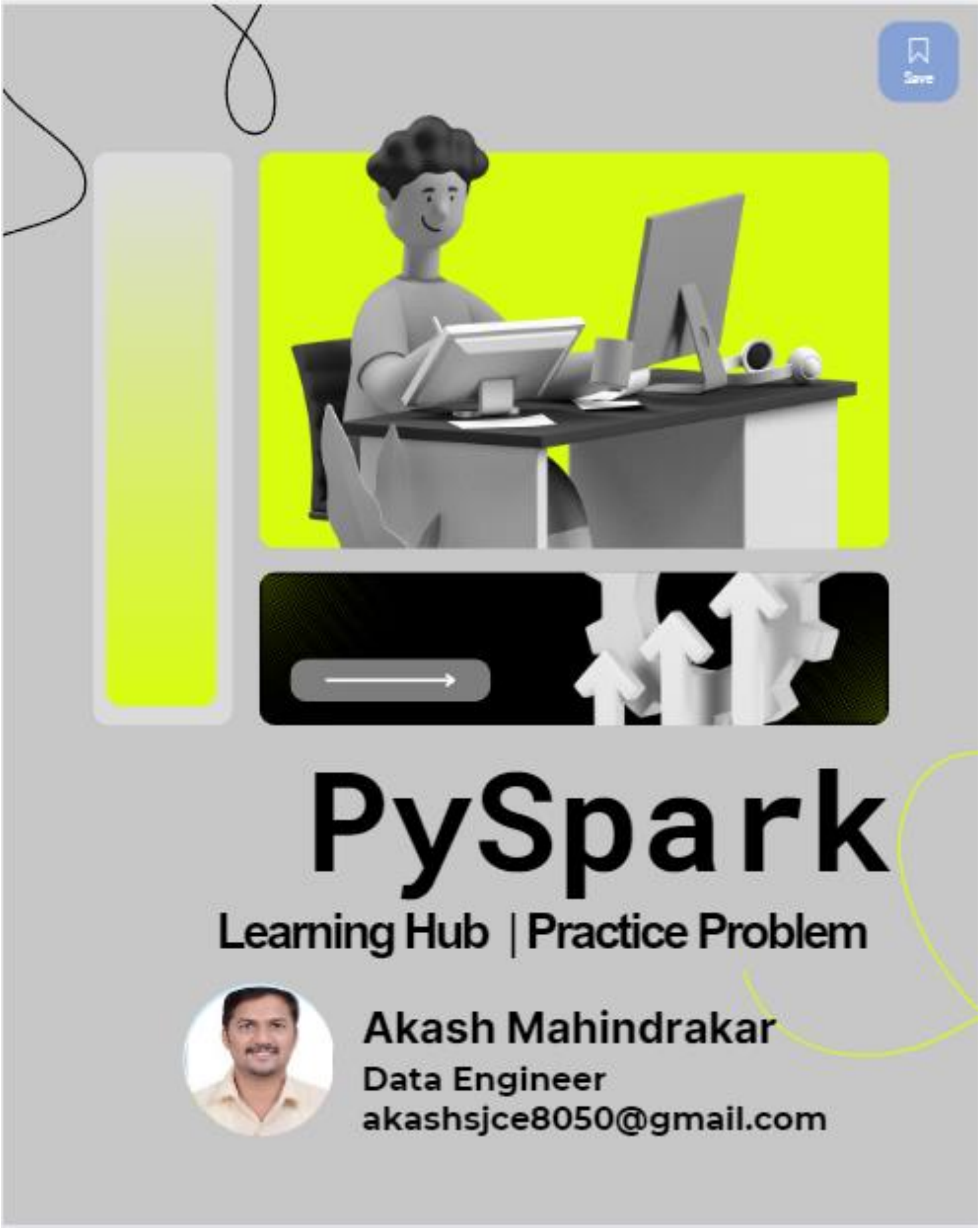
PYSPARK LEARNING HUB : DAY - 25



PySpark

Learning Hub | Practice Problem

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

WWW.LINKEDIN.COM/IN/AKASHMAHINDRAKAR

Step - 1 : Problem Statement

25_operator in pyspark

Write a pyspark code perform below function

- Select first name from "EmployeeDetail" table prifixed with "Hello "
- Get employee details from "EmployeeDetail" table whose Salary greater than 600000
- Get employee details from "EmployeeDetail" table whose Salary less than 700000
- Get employee details from "EmployeeDetail" table whose Salary between 500000 than 600000
- Select second highest salary from "EmployeeDetail" table

Difficult Level : EASY

DataFrame:

```
data = [
        [1, "Vikas", "Ahlawat", 600000.0, "2013-02-15 11:16:28.290", "IT", "Male"],
        [2, "nikita", "Jain", 530000.0, "2014-01-09 17:31:07.793", "HR", "Female"],
        [3, "Ashish", "Kumar", 1000000.0, "2014-01-09 10:05:07.793", "IT", "Male"],
        [4, "Nikhil", "Sharma", 480000.0, "2014-01-09 09:00:07.793", "HR", "Male"],
        [5, "anish", "kadian", 500000.0, "2014-01-09 09:31:07.793", "Payroll", "Male"],
]
```

```
# Create a schema for the DataFrame
```

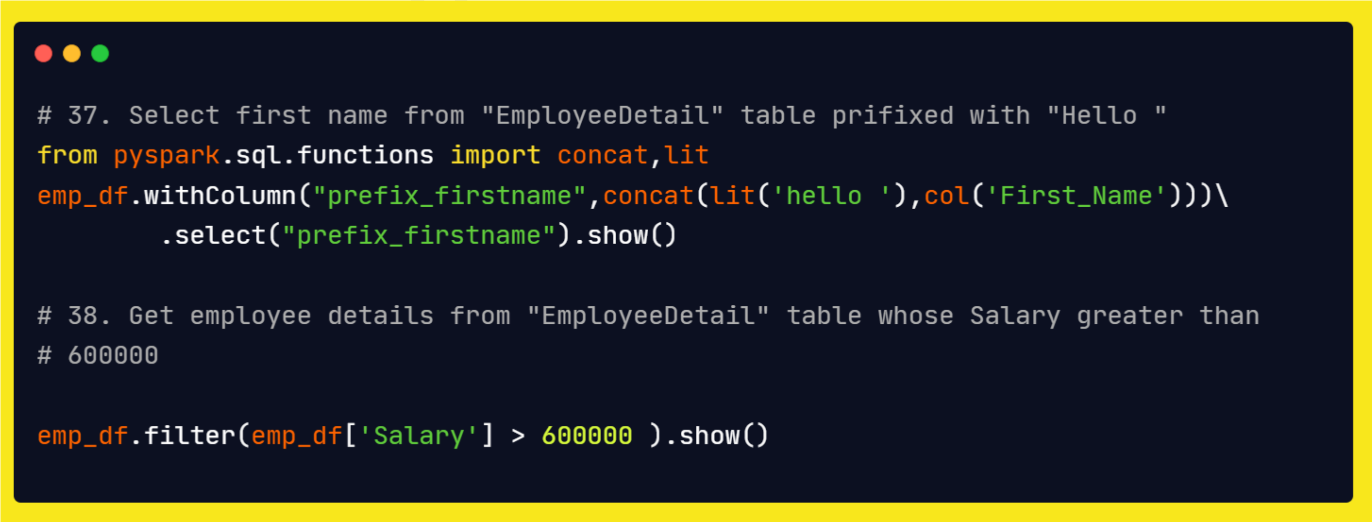
```
schema = StructType([
        StructField("EmployeeID", IntegerType(), True),
        StructField("First_Name", StringType(), True),
        StructField("Last_Name", StringType(), True),
        StructField("Salary", DoubleType(), True),
        StructField("Joining_Date", StringType(), True),
        StructField("Department", StringType(), True),
        StructField("Gender", StringType(), True)
])
```

Step - 2 : Writing the pyspark code to solve the

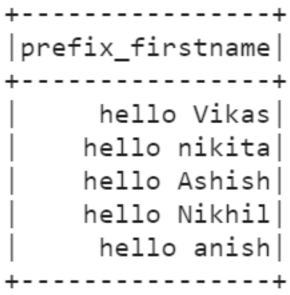
```
# Creating Spark Session
from pyspark.sql import SparkSession
from pyspark.sql.types import
StructType,StructField,IntegerType,StringType

#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()

# Create a list of rows from the image
data = [
        [1, "Vikas", "Ahlawat", 600000.0, "2013-02-15 11:16:28.290", "IT", "Male"],
        [2, "nikita", "Jain", 530000.0, "2014-01-09 17:31:07.793", "HR", "Female"],
        [3, "Ashish", "Kumar", 1000000.0, "2014-01-09 10:05:07.793", "IT", "Male"],
        [4, "Nikhil", "Sharma", 480000.0, "2014-01-09 09:00:07.793", "HR", "Male"],
        [5, "anish", "kadian", 500000.0, "2014-01-09 09:31:07.793", "Payroll", "Male"],
]

# Create a schema for the DataFrame
```

```
schema = StructType([
        StructField("EmployeeID", IntegerType(), True),
        StructField("First_Name", StringType(), True),
        StructField("Last_Name", StringType(), True),
        StructField("Salary", DoubleType(), True),
        StructField("Joining_Date", StringType(), True),
        StructField("Department", StringType(), True),
        StructField("Gender", StringType(), True)
])

emp_df=spark.createDataFrame(data,schema)
```

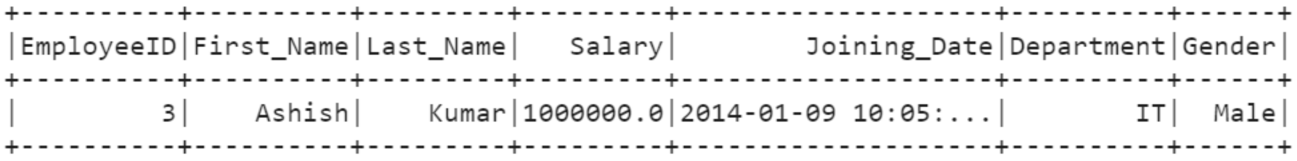
```
+----------+----------+---------+---------+--------------------+----------+------+
|EmployeeID|First_Name|Last_Name|   Salary|        Joining_Date|Department|Gender|
+----------+----------+---------+---------+--------------------+----------+------+
|         1|     Vikas|  Ahlawat| 600000.0|2013-02-15 11:16:...|        IT|  Male|
|         2|    nikita|     Jain| 530000.0|2014-01-09 17:31:...|        HR|Female|
|         3|    Ashish|    Kumar|1000000.0|2014-01-09 10:05:...|        IT|  Male|
|         4|    Nikhil|   Sharma| 480000.0|2014-01-09 09:00:...|        HR|  Male|
|         5|     anish|   kadian| 500000.0|2014-01-09 09:31:...|   Payroll|  Male|
+----------+----------+---------+---------+--------------------+----------+------+
```



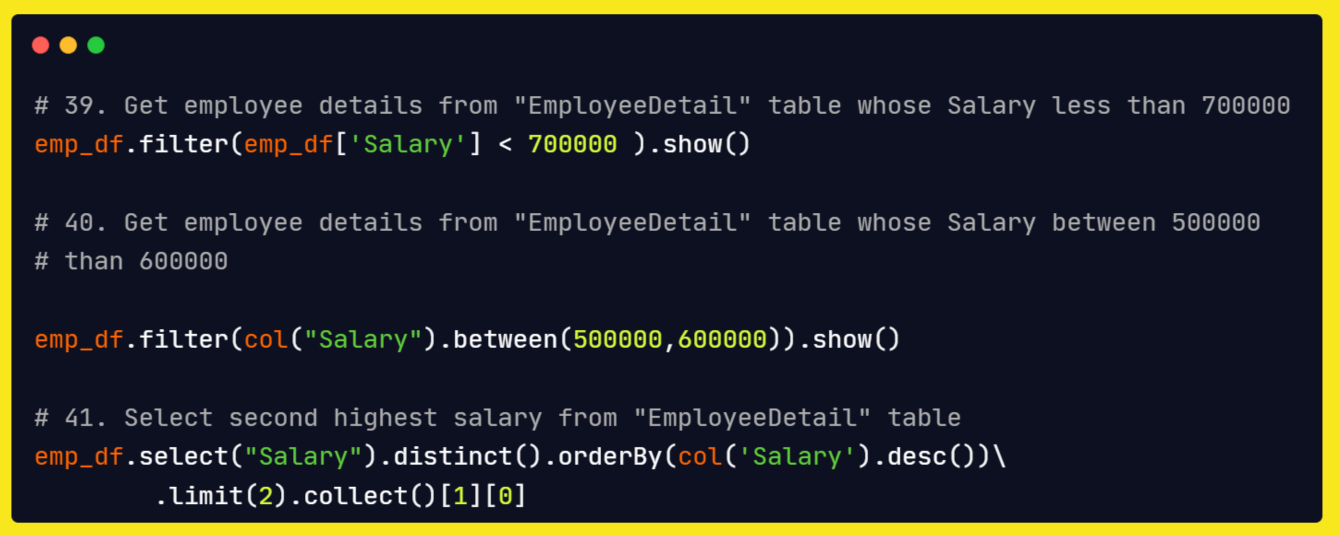
```
# 37. Select first name from "EmployeeDetail" table prifixed with "Hello "
from pyspark.sql.functions import concat,lit
emp_df.withColumn("prefix_firstname",concat(lit('hello '),col('First_Name')))\
        .select("prefix_firstname").show()

# 38. Get employee details from "EmployeeDetail" table whose Salary greater than
# 600000

emp_df.filter(emp_df['Salary'] > 600000 ).show()
```



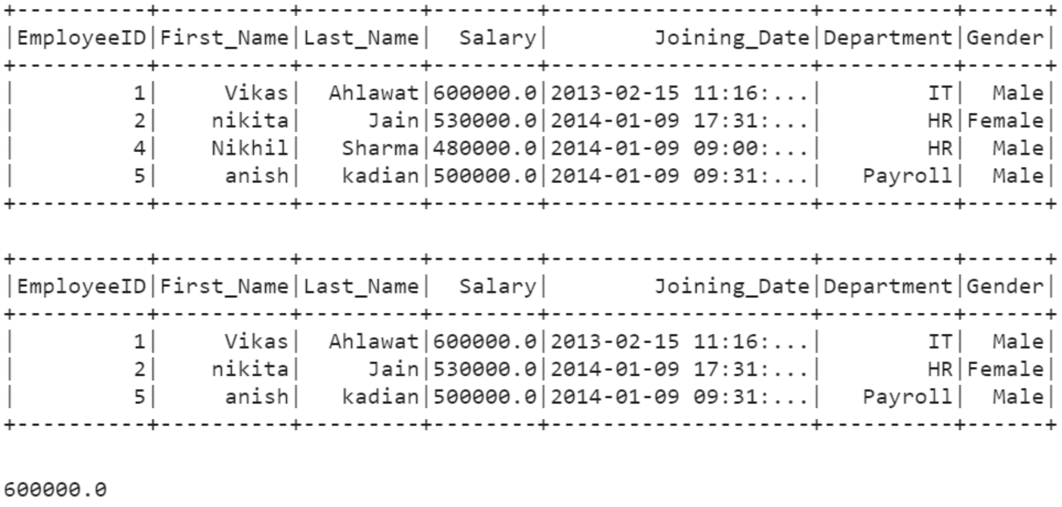
```
+----------------+
|prefix_firstname|
+----------------+
|     hello Vikas|
|    hello nikita|
|    hello Ashish|
|    hello Nikhil|
|     hello anish|
+----------------+
```



```
+----------+----------+---------+---------+--------------------+----------+------+
|EmployeeID|First_Name|Last_Name|   Salary|        Joining_Date|Department|Gender|
+----------+----------+---------+---------+--------------------+----------+------+
|         3|    Ashish|    Kumar|1000000.0|2014-01-09 10:05:...|        IT|  Male|
+----------+----------+---------+---------+--------------------+----------+------+
```



```
# 39. Get employee details from "EmployeeDetail" table whose Salary less than 700000
emp_df.filter(emp_df['Salary'] < 700000 ).show()

# 40. Get employee details from "EmployeeDetail" table whose Salary between 500000
# than 600000

emp_df.filter(col("Salary").between(500000,600000)).show()

# 41. Select second highest salary from "EmployeeDetail" table
emp_df.select("Salary").distinct().orderBy(col('Salary').desc())\
        .limit(2).collect()[1][0]
```



```
+----------+----------+---------+--------+--------------------+----------+------+
|EmployeeID|First_Name|Last_Name|  Salary|        Joining_Date|Department|Gender|
+----------+----------+---------+--------+--------------------+----------+------+
|         1|     Vikas|  Ahlawat|600000.0|2013-02-15 11:16:...|        IT|  Male|
|         2|    nikita|     Jain|530000.0|2014-01-09 17:31:...|        HR|Female|
|         4|    Nikhil|   Sharma|480000.0|2014-01-09 09:00:...|        HR|  Male|
|         5|     anish|   kadian|500000.0|2014-01-09 09:31:...|   Payroll|  Male|
+----------+----------+---------+--------+--------------------+----------+------+

+----------+----------+---------+--------+--------------------+----------+------+
|EmployeeID|First_Name|Last_Name|  Salary|        Joining_Date|Department|Gender|
+----------+----------+---------+--------+--------------------+----------+------+
|         1|     Vikas|  Ahlawat|600000.0|2013-02-15 11:16:...|        IT|  Male|
|         2|    nikita|     Jain|530000.0|2014-01-09 17:31:...|        HR|Female|
|         5|     anish|   kadian|500000.0|2014-01-09 09:31:...|   Payroll|  Male|
+----------+----------+---------+--------+--------------------+----------+------+

600000.0
```

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

SHARE YOUR THOUGHTS
IN COMMENT BELOW