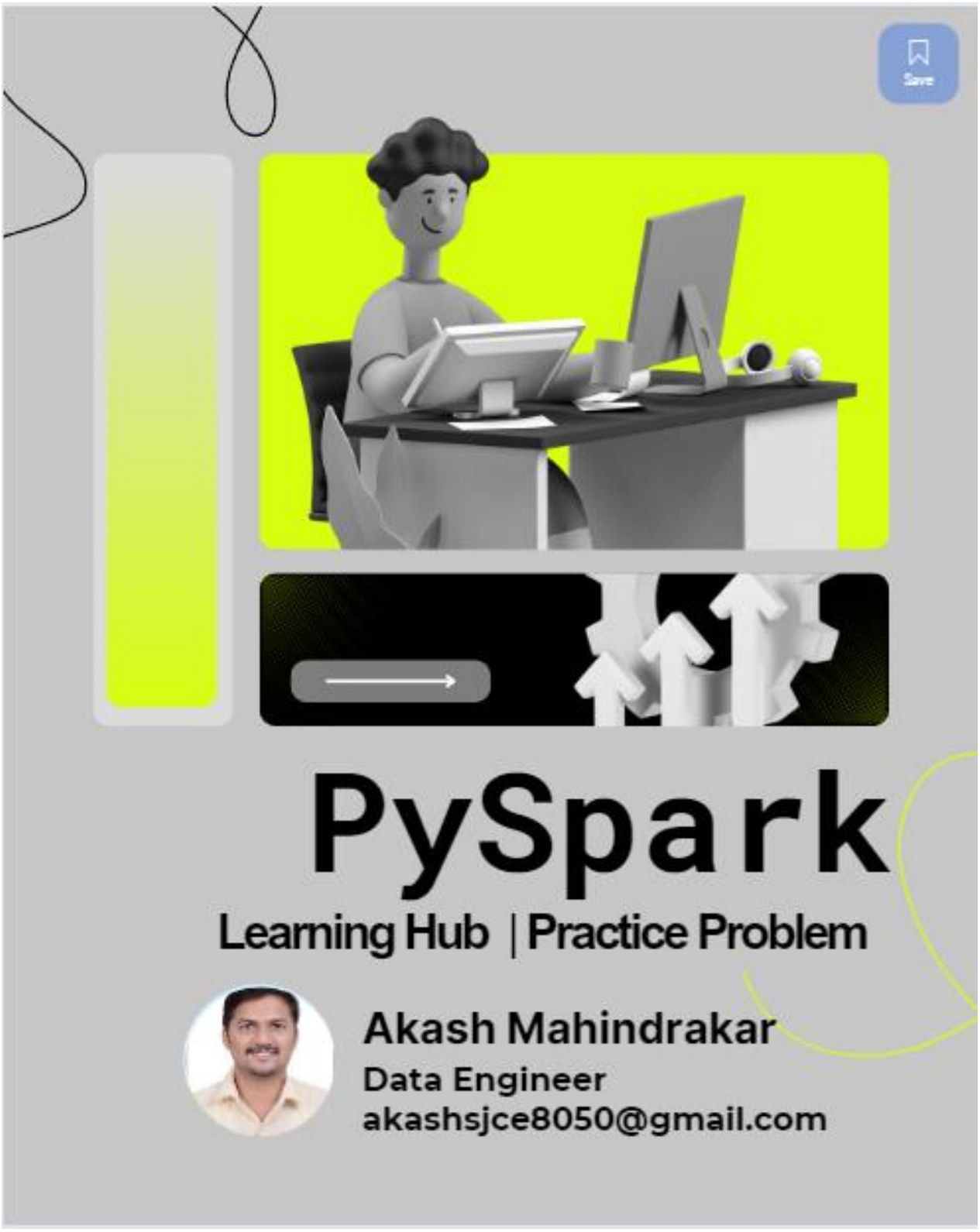
PYSPARK LEARNING HUB : DAY - 19



Step - 1 : Problem Statement

19_Select in pyspark

Write a pyspark code perform below function

- Get all employee detail from emp_df whose "Gender" end with 'le' and contain 4 letters. The Underscore(_) Wildcard Character represents any single character.
- Get all employee detail from EmployeeDetail table whose "FirstName" start with # 'A' and contain 5 letters.
- Get all unique "Department" from EmployeeDetail table.
- Get the 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

```
# 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|
+----------+----------+---------+---------+--------------------+----------+------+
```

Get all employee detail from emp_df whose "Gender" end with 'le'and contain 4 letters. The Underscore(_) Wildcard Character represents any single character.

```
# Get all employee detail from emp_df  whose "Gender" end with 'le'
# and contain 4 letters. The Underscore(_) Wildcard Character represents any single
# character.

emp_df.filter(emp_df["Gender"].like("__le")).show()
```

```
+----------+----------+---------+---------+--------------------+----------+------+
|EmployeeID|First_Name|Last_Name|   Salary|        Joining_Date|Department|Gender|
+----------+----------+---------+---------+--------------------+----------+------+
|         1|     Vikas|  Ahlawat| 600000.0|2013-02-15 11:16:...|        IT|  Male|
|         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|
+----------+----------+---------+---------+--------------------+----------+------+
```

 # Get all employee detail from EmployeeDetail table whose "FirstName" start with

 # 'A' and contain 5 letters.

```
# Get all employee detail from EmployeeDetail table whose "FirstName" start with
# 'A' and contain 5 letters.

emp_df.filter(emp_df["First_NamE"].like("a____")).show()
```

```
+----------+----------+---------+--------+--------------------+----------+------+
|EmployeeID|First_Name|Last_Name|  Salary|        Joining_Date|Department|Gender|
+----------+----------+---------+--------+--------------------+----------+------+
|         5|     anish|   kadian|500000.0|2014-01-09 09:31:...|   Payroll|  Male|
+----------+----------+---------+--------+--------------------+----------+------+
```

 # Get all unique "Department" from EmployeeDetail table.

```
# Get all unique "Department" from EmployeeDetail table.


emp_df.select("Department").distinct().show()
```

```
+-----------+
|Department|
+-----------+
|         HR|
|    Payroll|
|         IT|
+-----------+
```

 # Get the highest "Salary" from EmployeeDetail table.

```
# Get the highest "Salary" from EmployeeDetail table.


emp_df.agg(max("Salary")).show()
```

```
1000000.0
```

PYSPARK LEARNING HUB : DAY - 19

Save

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

Comment

SHARE YOUR THOUGHTS IN COMMENT BELOW

Share