



**PySpark**  
Learning Hub | Practice Problem



**Akash Mahindrakar**  
Data Engineer  
akashsjce8050@gmail.com

## Step - 1 : Problem Statement

### 30\_Join\_in\_pyspark

Write a pyspark code perform below function

- 56. Write a pyspark code to find out the employeename who has not assigned any project, and display "-No Project Assigned"( tables :- [EmployeeDetail],[ProjectDetail]).
- 57. Write a pyspark code to find out the project name which is not assigned to any employee( tables :- [EmployeeDetail],[ProjectDetail]).

**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"],  
]
```

## PYSPARK LEARNING HUB : DAY - 30

# 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)
])
```

```
pro_schema = StructType([
    StructField("Project_DetailID", IntegerType(), True),
    StructField("Employee_DetailID", IntegerType(), True),
    StructField("Project_Name", StringType(), True)
])
```

# Create the data as a list of tuples

```
pro_data = [
    (1, 1, "Task Track"),
    (2, 1, "CLP"),
    (3, 1, "Survey Management"),
    (4, 2, "HR Management"),
    (5, 3, "Task Track"),
    (6, 3, "GRS"),
    (7, 3, "DDS"),
    (8, 4, "HR Management"),
    (9, 6, "GL Management")
]
```

# PYSPARK LEARNING HUB : DAY - 30

## Step - 2 : Writing the pyspark code to solve the

```
# import packages
from pyspark.sql import SparkSession
from pyspark.sql.types import
StructType, StructField, IntegerType, StringType, DoubleType, TimestampType
from pyspark.sql.functions import col

#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()
```

```
pro_schema = StructType([
    StructField("Project_DetailID", IntegerType(), True),
    StructField("Employee_DetailID", IntegerType(), True),
    StructField("Project_Name", StringType(), True)
])
# Create the data as a list of tuples
pro_data = [
    (1, 1, "Task Track"),
    (2, 1, "CLP"),
    (3, 1, "Survey Management"),
    (4, 2, "HR Management"),
    (5, 3, "Task Track"),
    (6, 3, "GRS"),
    (7, 3, "DDS"),
    (8, 4, "HR Management"),
    (9, 6, "GL Management")
]
pro_df=spark.createDataFrame(pro_data,pro_schema)
pro_df.show()
```

## PYSPARK LEARNING HUB : DAY - 30

Project_DetailID	Employee_DetailID	Project_Name
1	1	Task Track
2	1	CLP
3	1	Survey Management
4	2	HR Management
5	3	Task Track
6	3	GRS
7	3	DDS
8	4	HR Management
9	6	GL Management

```
# Create a list of rows from the image
emp_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
emp_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(emp_data,emp_schema)
emp_df.show()
```

## PYSPARK LEARNING HUB : DAY - 30

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

```
# 56. Write a query to find out the employee name who has not assigned any project,
# and display "-No Project Assigned" ( tables :- [EmployeeDetail],[ProjectDetail]).
from pyspark.sql.functions import lower,coalesce,lit

emp_df.join(pro_df,emp_df['EmployeeID'] = pro_df['Employee_DetailID'] ,"left")\
    .filter(col("Project_Name").isNull())\
    .select("First_Name",coalesce(col("Project_Name"), lit("-No Project
Assigned"))).alias("Project_Name"))\
    .orderBy(lower(col("First_Name")))\
    .show()
```

First_Name	Project_Name
anish	-No Project Assigned



```
# 57. Write a query to find out the project name which is not assigned to any employee(  
# tables :- [EmployeeDetail],[ProjectDetail]).
```

```
from pyspark.sql.functions import col  
result_df=emp_df.join(pro_df,emp_df["EmployeeID"]=pro_df["Employee_DetailID"],"right")\  
    .select("EmployeeID","Project_Name")  
  
result_df.filter(col("EmployeeID").isNull()).select("Project_Name").show()
```

```
+-----+  
| Project_Name |  
+-----+  
| GL Management |  
+-----+
```



Save

**Was it  
helpful?**  
follow for more!



**Akash Mahindrakar**

Data Engineer

akashsjce8050@gmail.com



Comment

**SHARE YOUR THOUGHTS  
IN COMMENT BELOW**



Share