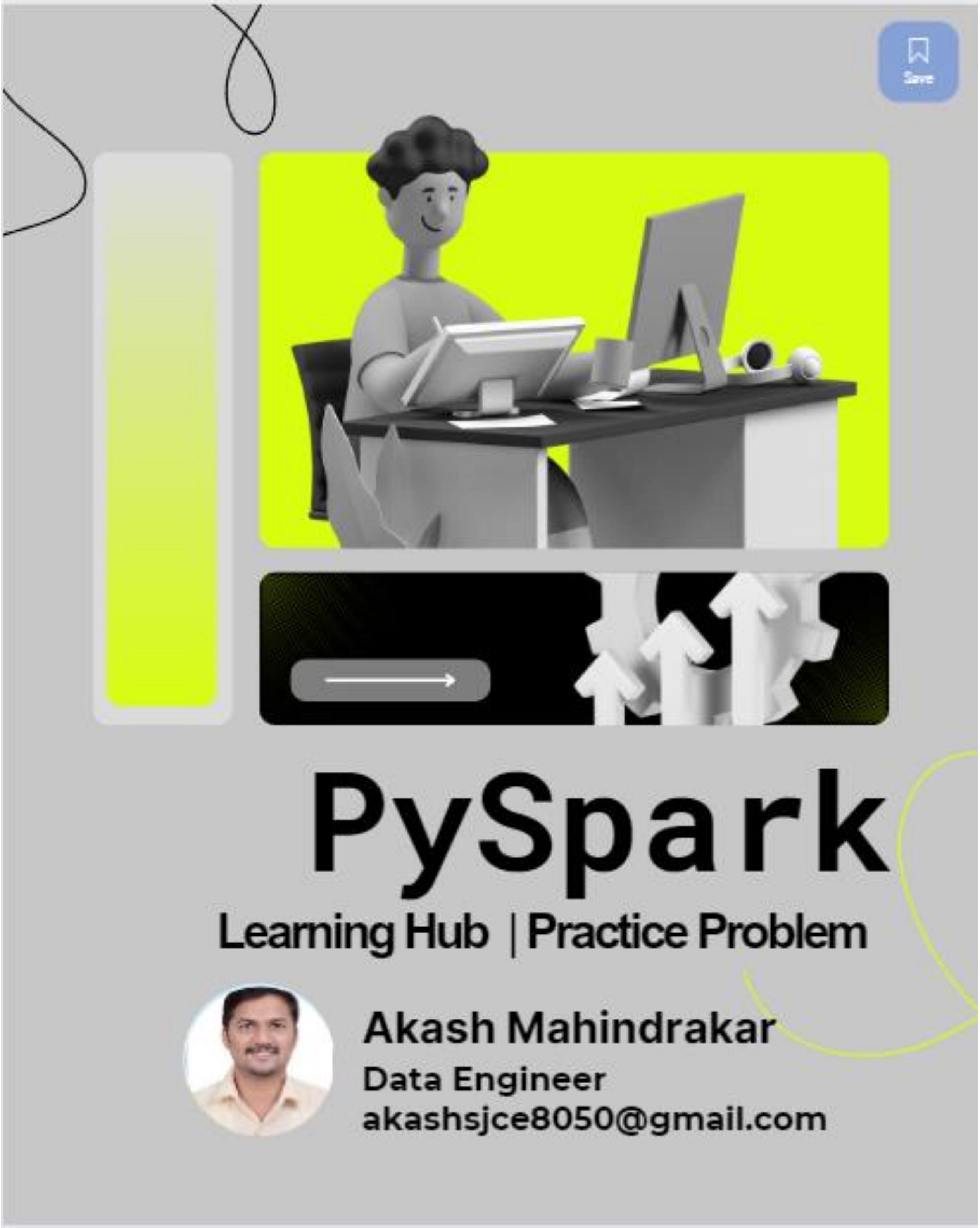
PYSPARK LEARNING HUB : DAY - 8



PySpark

Learning Hub | Practice Problem

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

WWW.LINKEDIN.COM/IN/AKASHMAHINDRAKAR

Step - 1 : Problem Statement

08_Game Play Analysis I

Write a solution to find the first login date for each player.

Return the result table in any order.

Difficult Level : EASY

DataFrame:

```
# Define the schema for the "Activity"
activity_schema = StructType([
        StructField("player_id", IntegerType(), True),
        StructField("device_id", IntegerType(), True),
        StructField("event_date", StringType(), True),
        StructField("games_played", IntegerType(), True)
])

# Define data for the "Activity"
activity_data = [
        (1, 2, '2016-03-01', 5),
        (1, 2, '2016-05-02', 6),
        (2, 3, '2017-06-25', 1),
        (3, 1, '2016-03-02', 0),
        (3, 4, '2018-07-03', 5)
]
```

PYSPARK LEARNING HUB : DAY - 8

Step - 2 : Identifying The Input Data And Expected Output

INPUT

INPUT			
PLAYER_ID	DEVICE_ID	EVENT_DATE	GAMES_PLAYED
1	2	2016-03-01	5
1	2	2016-05-02	6
2	3	2017-06-25	1
3	1	2016-03-02	0
3	4	2018-07-03	5

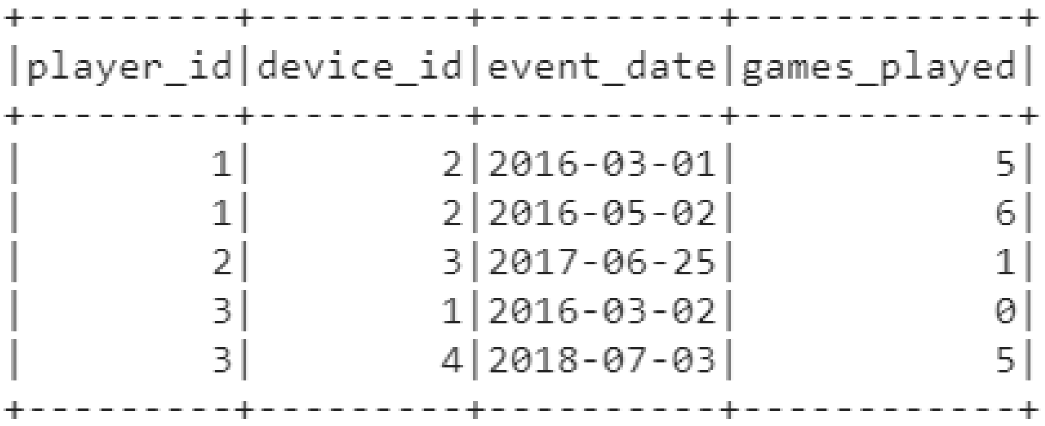
OUTPUT

OUTPUT	
PLAYER_ID	FISRT_LOGIN
1	2016-03-01
2	2017-06-25
3	2016-03-02

Step - 3 : Writing the pyspark code to solve

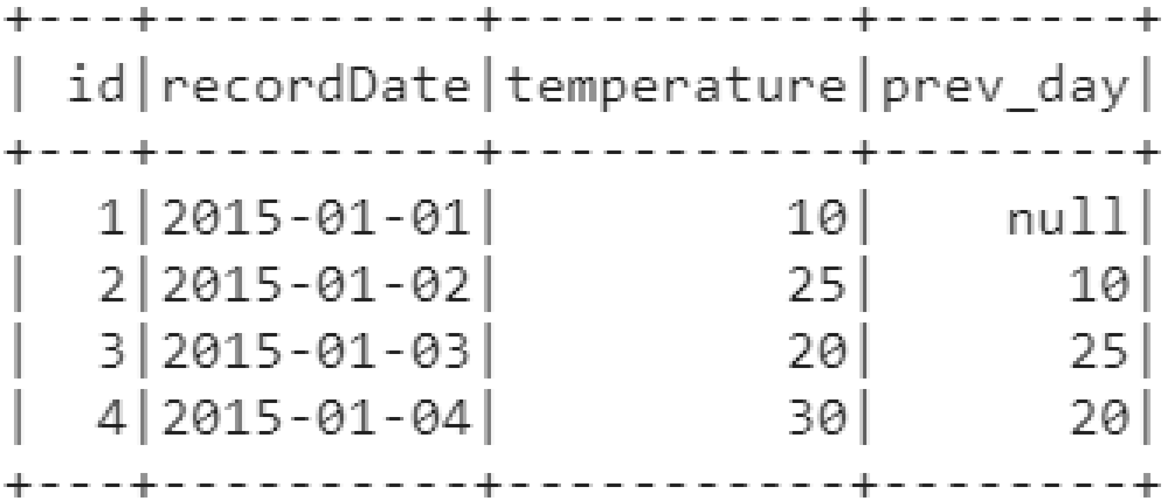
PYSPARK LEARNING HUB : DAY - 8

```
# 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 "Activity"
activity_schema = StructType([
        StructField("player_id", IntegerType(), True),
        StructField("device_id", IntegerType(), True),
        StructField("event_date", StringType(), True),
        StructField("games_played", IntegerType(), True)
])

# Define data for the "Activity"
activity_data = [
        (1, 2, '2016-03-01', 5),
        (1, 2, '2016-05-02', 6),
        (2, 3, '2017-06-25', 1),
        (3, 1, '2016-03-02', 0),
        (3, 4, '2018-07-03', 5)
]



# Create a PySpark DataFrame

activity_df=spark.createDataFrame(activity_data,activity_schema)
activity_df.show()
```

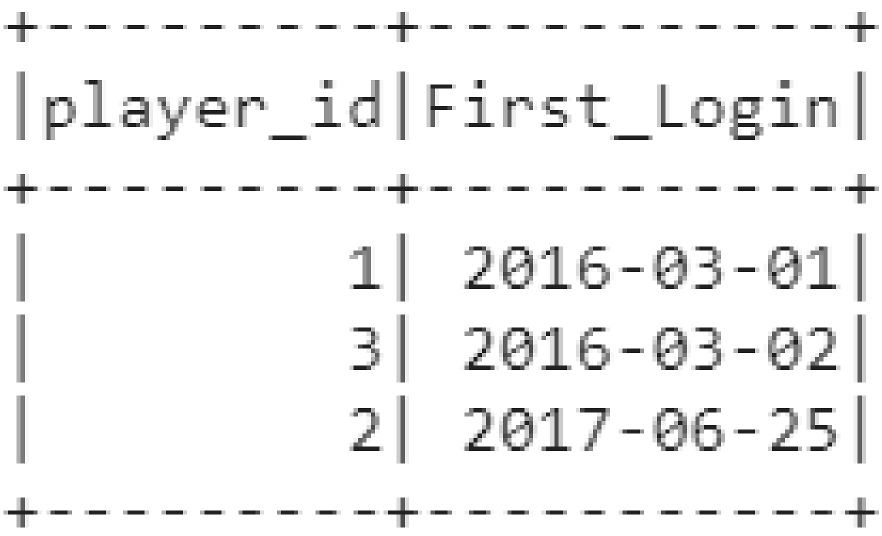


```
+---------+---------+----------+------------+
|player_id|device_id|event_date|games_played|
+---------+---------+----------+------------+
|        1|        2|2016-03-01|           5|
|        1|        2|2016-05-02|           6|
|        2|        3|2017-06-25|           1|
|        3|        1|2016-03-02|           0|
|        3|        4|2018-07-03|           5|
+---------+---------+----------+------------+
```

```
rank_df=activity_df.withColumn("RK",rank().over(Window.partitionBy(activity_df['player_id']).orderBy(activity_df['event_date'])))
rank_df.show()
```



```
+---+----------+-----------+--------+
| id|recordDate|temperature|prev_day|
+---+----------+-----------+--------+
|  1|2015-01-01|         10|    null|
|  2|2015-01-02|         25|      10|
|  3|2015-01-03|         20|      25|
|  4|2015-01-04|         30|      20|
+---+----------+-----------+--------+
rank_df.filter(rank_df["RK"] == 1).select("player_id",rank_df["event_date"].alias("First_Login")).show()
```



```
+---------+-----------+
|player_id|First_Login|
+---------+-----------+
|        1| 2016-03-01|
|        3| 2016-03-02|
|        2| 2017-06-25|
+---------+-----------+
```

Save

Was it helpful?

follow for more!

Akash Mahindrakar
Data Engineer
akashsjce8050@gmail.com

Comment

SHARE YOUR THOUGHTS IN COMMENT BELOW

Share