



Monitor data quality at scale

Get instant data observability

[Home](#) > [spark with scala](#) > [How to Replace a String in Spark DataFrame | Spark Scenario Based Question](#)

How to Replace a String in Spark DataFrame | Spark Scenario Based Question

[Azarudeen Shahul](#) 7:22 PM



Monitor data quality at scale

Get instant data observability

Stay on top of data quality. Detect data drift and data outages before they hurt business.

lightup.ai

OPEN

In this tutorial, we will see how to solve the problem statement and get required output as shown in the below picture. We will learn, how to replace a character or String in Spark Dataframe using both PySpark and Spark with Scala as a programming language. We use Databricks community Edition for our demo. Let us move on to the problem statement.

Problem Statement:

Consider we have a dataframe with columns as shown in the below figure (Input_DF). Our requirement is to replace the string value *Checking* in column called *Card_type* to *Cash*. The output that we are expected to workout is shown in the below figure for your reference.

[Twitter](#)

Subscribe to



Azarudeen

You

Follow on FB



Learn to Spark



Learn to Spark

How to Read Data Explained using


<https://youtu.be/...>



Apach

Read Data

11 DI



Spark Scenario Question

Spark SQL Functions to Replace

Input_DF:

Customer_No	Card_type	Date	Category	Transaction Type	Amount
1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34
1000210	Silver Card	3/19/2018	Restaurants	debit	36.48
1000210	Checking	3/19/2018	Utilities	debit	35
1000210	Platinum Card	3/20/2018	Shopping	debit	14.97
1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55
1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91
1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91

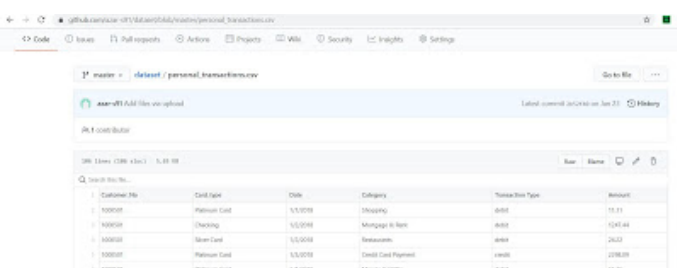
Question:
Consider a Spark Dataframe as shown above, Need to replace a string in column Card_type from Checking -> Cash as shown in the result above.

Output_DF:

Customer_No	Card_type	Date	Category	Transaction Type	Amount
1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34
1000210	Silver Card	3/19/2018	Restaurants	debit	36.48
1000210	Cash	3/19/2018	Utilities	debit	35
1000210	Platinum Card	3/20/2018	Shopping	debit	14.97
1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55
1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91
1000210	Cash	3/23/2018	Credit Card Payment	debit	559.91

Dataset:

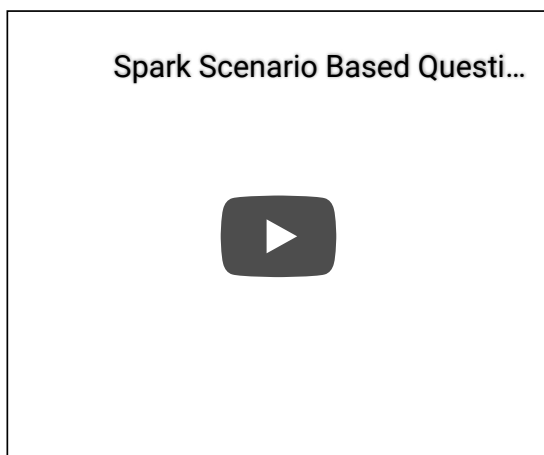
Dataset can be downloaded from the given Github link. Click on the link [personal_transaction.csv](#) to download dataset. It is a simple CSV file with transaction details in it.



Customer_No	Card_type	Date	Category	Transaction Type	Amount
1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34
1000210	Silver Card	3/19/2018	Restaurants	debit	36.48
1000210	Checking	3/19/2018	Utilities	debit	35
1000210	Platinum Card	3/20/2018	Shopping	debit	14.97
1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55
1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91
1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91

Solution with Demo:

We have different ways to achieve our expected output. We will see all the approach one by one to get the final required output. Before proceeding with the answer I would recommend you to give it a try on your own to solve this problem.



For more videos on Spark Scenario Based Interview Question, please do subscribe to my YouTube channel.

Twitter

Subscribe to



Azarudee

You

Follow on FB



Learn to Spark



Learn to Spark

How to Read Data Explained using

<https://youtu.k>



Apach

Read Data

11 DI

Solution - Using PySpark:

```
trans_path="dbfs:/FileStore/shared_uploads/personal_transactions.csv"
```

```
df1 = spark.read.format("csv").option("header",true).load(trans_path)
```

```
display(df1)
```

Out[]:

	Customer_No	Card_type	Date	Category	Transaction Type	Amount
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48
3	1000210	Checking	3/19/2018	Utilities	debit	35
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91

Method 1: Using na.replace

We can use `na.replace` to replace a string in any column of the Spark dataframe.

```
na_replace_df=df1.na.replace("Checking","Cash")
```

```
na_replace_df.show()
```

Out[]:

	Customer_No	Card_type	Date	Category	Transaction Type	Amount
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48
3	1000210	Cash	3/19/2018	Utilities	debit	35
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91
7	1000210	Cash	3/23/2018	Credit Card Payment	debit	559.91

From the above output we can observe that the highlighted value *Checking* is replaced with *Cash*.

Method 2: Using regular expression replace

The most common method that one uses to replace a string in Spark Dataframe is by using Regular expression `Regex_replace` function. The Code Snippet to achieve this, as follows.

```
#import the required function
```

```
from pyspark.sql.functions import regexp_replace
```

```
reg_df=df1.withColumn("card_type_rep",regexp_replace("Card_type","Checking",  
"Cash"))
```

```
reg_df.show()
```

Out[]:

Twitter

Subscribe to



Azarudee

You

Follow on FB



Learn



Learn

ಸುಮಾರು

How to Read
Explained usi

<https://youtu.k>



Apach

Read Data

11DI

	Customer_No	Card_type	Date	Category	Transaction Type	Amount	card_type_repl
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34	Platinum Card
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48	Silver Card
3	1000210	Checking	3/19/2018	Utilities	debit	35	Cash
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97	Platinum Card
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55	Silver Card
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91	Platinum Card
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91	Cash

Method 3: Using Case When

The traditional method that fits in to solve many problem is to simply write case when condition. The Code Snippet to replace the string using case when is given below.

```
from pyspark.sql.functions import when,col,lit
```

```
when_df=df1.withColumn("card_type_repl",when(col("Card_type").rlike("Checking"),lit("Ca  
sh")).otherwise(col("Card_type")))
```

```
when_df.show()
```

Out[]:

	Customer_No	Card_type	Date	Category	Transaction Type	Amount	card_type_repl
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34	Platinum Card
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48	Silver Card
3	1000210	Checking	3/19/2018	Utilities	debit	35	Cash
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97	Platinum Card
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55	Silver Card
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91	Platinum Card
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91	Cash

Pulsefire Haste gaming mouse

Show off your level and ranking with HyperX pro gaming gear.

HyperX

Solution - Using Spark With Scala:

Let us see how we can approach this problem using Spark with Scala. There will be a minimal syntax change with respect to the above pyspark code for implementing using Scala. As a first step let us read the csv file that we have

```
val trans_path="dbfs:/FileStore/shared_uploads/personal_transactions.csv"
```

```
val df1 = spark.read.format("csv").option("header","true").load(trans_path)
```

```
df1.show()
```

Out[]:

Twitter

Subscribe to



Azarudee

You

Follow on FB



Learn

ಪುಸ್ತಕ



Learn

ಸುಮಾರು

How to Read
Explained usi

<https://youtu.k>



Apach

Read Data
11 DI

	Customer_No ▲	Card_type ▲	Date ▲	Category ▲	Transaction Type ▲	Amount ▲
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48
3	1000210	Checking	3/19/2018	Utilities	debit	35
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91

Method 1: Using Regexp Replace

```
import org.apache.spark.sql.functions.{regexp_replace,lit}

val
reg_df=df1.withColumn("card_type_repl",regexp_replace($"Card_type",lit("Checking"),lit("
Cash")))

reg_df.show()
```

Out]:

	Customer_No ▲	Card_type ▲	Date ▲	Category ▲	Transaction Type ▲	Amount ▲	card_type_repl
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34	Platinum Card
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48	Silver Card
3	1000210	Checking	3/19/2018	Utilities	debit	35	Cash
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97	Platinum Card
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55	Silver Card
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91	Platinum Card
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91	Cash

Method 2: Using Case When

Case When Statement is similar to the Pyspark approach that we saw above with some minor change to the syntax. The code snippet as follows.

```
import org.apache.spark.sql.functions.{when,lit,col}

val
when_df=df1.withColumn("card_type_repl",when(col("Card_type").rlike("Checking"),lit("Ca
sh")).otherwise(col("Card_type")))

when_df.show()
```

Out]:

	Customer_No ▲	Card_type ▲	Date ▲	Category ▲	Transaction Type ▲	Amount ▲	card_type_repl
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34	Platinum Card
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48	Silver Card
3	1000210	Checking	3/19/2018	Utilities	debit	35	Cash
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97	Platinum Card
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55	Silver Card
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91	Platinum Card
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91	Cash

Method 3: Using UDF

We don't have na.replace function in Scala. We can write our own custom function to replace the character in the dataframe using native Scala functions. The code snippet for UDF is given below

```
val replace = udf((data: String , rep : String, withrep:String)=>data.replaceAll(rep, withrep))
```

Twitter

Subscribe to



Azarudee

You

Follow on FB



Learn Spark



Learn Spark

How to Read
Explained usi

<https://youtu.k>



Apach

Read Data

11 DI

```
val
udf_df=df1.withColumn("card_type_repl",replace($"Card_type",lit("Checking"),lit("Cash")))

udf_df.show()
```

Out[]:

	Customer_No	Card_type	Date	Category	Transaction Type	Amount	card_type_repl
1	1000210	Platinum Card	3/17/2018	Fast Food	debit	23.34	Platinum Card
2	1000210	Silver Card	3/19/2018	Restaurants	debit	36.48	Silver Card
3	1000210	Checking	3/19/2018	Utilities	debit	35	Cash
4	1000210	Platinum Card	3/20/2018	Shopping	debit	14.97	Platinum Card
5	1000210	Silver Card	3/22/2018	Gas & Fuel	debit	30.55	Silver Card
6	1000210	Platinum Card	3/23/2018	Credit Card Payment	credit	559.91	Platinum Card
7	1000210	Checking	3/23/2018	Credit Card Payment	debit	559.91	Cash

Hope you understood the concept of replacing the string in Spark Dataframe. Kindly try all this method in your own setup, let me know if you face any issues, I am happy to help you.

Happy Learning !!!

Pulsefire Haste gaming mouse

Show off your level and ranking with HyperX pro gaming gear.

HyperX

Twitter

Subscribe to

Tags: add column to Dataframe Apache Spark Databricks pyspark regex replace replace string Spark Interview Question Spark Scenario Based Spark sql Spark with Python spark with scala

Facebook Twitter Google+ Pinterest LinkedIn WhatsApp Email

Post a Comment

1 Comments



VIC

July 28, 2021 at 9:31 AM

Great explanation!

Reply Delete

Replies

Reply

Add comment

Enter your comment...

Azarudee You

Follow on FB

Learn Spark

Learn Spark

How to Read Explained using

<https://youtu.k>

Apache

Read Data

11 DI

Pages

[Home](#)
[Disclaimer](#)
[Contact Us](#)

[Privacy Policy](#)
[Terms and Conditions](#)
[About Us](#)

Most Viewed Post



How to Transform Rows and Column using Apache Spark
🕒 11:30 PM



How to cast string datatype to date timestamp in Spark
🕒 4:13 AM



Setup HBase in Windows 10 | Install HBase in Standalone Mode
🕒 6:09 AM

Featured Post

 **DOWNLOAD FILES**

Apache Spark With Databricks
Download File to Local

How to Download Files from Databricks to Local

Apache Spark With Databricks | How to Download Data From Databricks to Local System

👤 Azarudeen Shahul 🕒 10:00 PM

In this tutorial, we will learn a trick in databricks on How to download the output r...