How to Replicate Rows in a PySpark DataFrame

Authored by

stats writer

February 4, 2026

Replicating rows in a PySpark DataFrame refers to the process of creating multiple copies of a single row within the DataFrame. This can be achieved by using the "repeat" function in PySpark, which allows for the duplication of a specific row based on a given number. This function is particularly useful when working with large datasets, as it allows for the creation of multiple identical rows without the need for manual data entry. By replicating rows in a PySpark DataFrame, users can easily generate larger datasets for testing or analysis purposes, saving time and effort in the process.

Replicate Rows in a PySpark DataFrame

You can use the following syntax to replicate each row in a PySpark DataFrame a certain number of times:

```
from pyspark.sql.functions import expr

df_new = df.withColumn('team', expr('explode(array_repeat(team, 3))'))
```

This particular example replicates each row in the DataFrame 3 times.

Note: We used the team column within the array_repeat function but you can use any column name that exists in the DataFrame and the result will be the same.

The following example shows how to use this syntax in practice.

Example: How to Replicate Rows in a PySpark DataFrame

Suppose we have the following PySpark DataFrame that contains information about various basketball players:

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.getOrCreate()

#define data
data = ,
,
,
]

#define column names
columns =

#create dataframe using data and column names
df = spark.createDataFrame(data, columns)

#view dataframe
df.show()

+----+----------+------+-------+
|team|conference|points|assists|
+----+----------+------+-------+
```

```
| A| East| 11| 4|
| A| West| 15| 7|
| B| West| 6| 12|
| C| East| 5| 2|
+----+----------+------+-------+
```

The PySpark DataFrame currently has 4 total rows.

We can use the following syntax to replicate each row 3 times so the resulting DataFrame will have a total of 12 rows:

```
from pyspark.sql.functions import expr

#replicate each row in DataFrame 3 times
df_new = df.withColumn('team',
expr('explode(array_repeat(team, 3))'))

#view new DataFrame
df_new.show()

+----+----------+------+-------+
|team|conference|points|assists|
+----+----------+------+-------+
| A| East| 11| 4|
```

```
| A| East| 11| 4|
| A| East| 11| 4|
| A| West| 15| 7|
| A| West| 15| 7|
| A| West| 15| 7|
| B| West| 6| 12|
| B| West| 6| 12|
| B| West| 6| 12|
| C| East| 5| 2|
| C| East| 5| 2|
| C| East| 5| 2|
+----+----------+------+-------+
```

Notice that each row in the original DataFrame has been replicated 3 times.

Note that we used the array_repeat function to create an array containing the team column repeated 3 times, then we used the explode function to return a new row for each element in the array.

The end result is that each row is repeated 3 times.

The following tutorials explain how to perform other

common tasks in PySpark: