How to Vertically Concatenate DataFrames in PySpark Using `union`

Authored by

stats writer

February 7, 2026

To vertically concatenate multiple DataFrames in PySpark, the "union" function can be used. This function combines two or more DataFrames by adding their rows together, resulting in a single DataFrame with all the rows from the original DataFrames. This method is useful for combining data from different sources or partitions into a single DataFrame for further analysis. It is an efficient and convenient way to handle large datasets in PySpark. Additionally, the "unionAll" function can be used to concatenate DataFrames without removing duplicate rows. Overall, using these functions allows for easy and efficient vertical concatenation of multiple DataFrames in PySpark.

Vertically Concatenate DataFrames in PySpark

You can use the following syntax to vertically concatenate multiple PySpark DataFrames:

```
from functools import reduce
from pyspark.sql import DataFrame

#specify DataFrames to concatenate
df_list =

#vertically concatenate all DataFrames in list
df_all = reduce(DataFrame.unionAll, df_list)
```

This particular example uses the reduce function along with the unionAll function to vertically concatenate the DataFrames named df1, df2 and df3 into one DataFrame called df_all.

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

Example: How to Vertically Concatenate DataFrames in PySpark

Suppose we have three PySpark DataFrames that each contain information about points scored by basketball players on various teams:

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

#define data
data1 = ,
,
]

data2 = ,
,
,
]

data3 = ,
,
,
]
```

```
#define column names
columns =

#create dataframes using data and column names
df1 = spark.createDataFrame(data1, columns)
df2 = spark.createDataFrame(data2, columns)
df3 = spark.createDataFrame(data3, columns)

#view dataframes
df1.show()
df2.show()
df3.show()

+------+------+
| team|points|
+------+------+
| Mavs| 18|
| Nets| 33|
|Lakers| 12|
+------+------+

+-------+------+
| team|points|
+-------+------+
| Kings| 15|
```

| Hawks| 19|
|Wizards| 24|
| Magic| 28|
+-------+------+

+-------+------+
| team|points|
+-------+------+
|Celtics| 25|
| Spurs| 29|
|Rockets| 14|
| Heat| 30|
+-------+------+

Suppose we would like to vertically concatenate each of the three DataFrames into one DataFrame.

We can use the following syntax to do so:

```
from functools import reduce
from pyspark.sql import DataFrame

#specify DataFrames to concatenate
df_list =
```

```
#vertically concatenate all DataFrames in list
df_all = reduce(DataFrame.unionAll, df_list)

#view resulting DataFrame
df_all.show()

+-------+------+
| team|points|
+-------+------+
| Mavs| 18|
| Nets| 33|
| Lakers| 12|
| Kings| 15|
| Hawks| 19|
|Wizards| 24|
| Magic| 28|
|Celtics| 25|
| Spurs| 29|
|Rockets| 14|
| Heat| 30|
+-------+------+
```

The new DataFrame named df_all contains the data from all three DataFrames concatenated vertically.

Note: You can find the complete documentation for the PySpark concat function .

The following tutorials explain how to perform other common tasks in PySpark: