

How to Explode an Array into Rows in PySpark

Authored by
stats writer

February 4, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Explode an Array into Rows in PySpark*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129351>

PySpark is a powerful tool that allows users to efficiently process and analyze large datasets using Python. One useful feature of PySpark is the ability to explode an array into rows, which allows for easier manipulation and analysis of data contained within arrays. This can be achieved by using the "explode" function, which splits each element of the array into a separate row, thereby expanding the dataset. This feature is particularly helpful for working with complex data structures and performing various data transformations. Overall, utilizing PySpark to explode an array into rows can greatly enhance the efficiency and effectiveness of data analysis tasks.

PySpark: Explode Array into Rows

You can use the following syntax to explode a column that contains arrays in a PySpark DataFrame into multiple rows:

```
from pyspark.sql.functions import explode
```

```
#explode points column into rows
```

```
df_new = df.withColumn('points', explode(df.points))
```

This particular example explodes the arrays in the points column of a DataFrame into multiple rows.

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

Example: How to Explode Array into Rows in a PySpark DataFrame

Suppose we have the following PySpark DataFrame that contains information about points scored in three

different games by 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|position| points|
```

```
+----+-----+-----+
```

```
| A| Guard| |
```

```
| A| Forward||
```

```
| B| Guard| |
```

```
| B| Forward|
```

```
+----+-----+-----+
```

Notice that the points column currently contains arrays.

We can use the following syntax to explode the values from each of these arrays into their own rows:

```
from pyspark.sql.functions import explode
```

```
#explode points column into rows
```

```
df_new = df.withColumn('points', explode(df.points))
```

```
#view new DataFrame
```

```
df_new.show()
```

```
+----+-----+-----+
```

```
|team|position|points|
```

```
+----+-----+-----+
```

```
| A| Guard| 11|
```

```
| A| Guard| 8|
```

```
| A| Guard| 25|
```

```
| A| Forward| 14|
```

```
| A| Forward| 20|
```

```
| A| Forward| 22|
```

```
| B| Guard| 21|  
| B| Guard| 30|  
| B| Guard| 6|  
| B| Forward| 22|  
| B| Forward| 12|  
| B| Forward| 34|  
+----+-----+-----+
```

Notice that each of the values in the arrays from the points column have been exploded into their own rows.

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

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