How to Add a Column in PySpark Based on Multiple AND Conditions Using `when()`

Authored by

stats writer

January 18, 2026

You can use the following syntax to use the **when** function in PySpark with **and** conditions:

import pyspark.sql.functions as F

df_new = df.withColumn('B10', F.when((df.team=='B') & (df.points>10), 1).otherwise(0))

This particular example creates a new column named **B10** that returns the following values:

1 if the **team** column is B *and* the **points** column is greater than 10
0 otherwise

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

Example: How to Use When with AND Condition in PySpark

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|position|points|

```
+----+--------+------+
| A| Guard| 11|
| A| Guard| 8|
| A| Forward| 22|
| A| Forward| 22|
| B| Guard| 14|
| B| Guard| 14|
| B| Forward| 13|
| B| Forward| 7|
+----+--------+------+
```

We can use the following syntax to create a new column named **B10** that returns **1** if the **team** is A *and* the **points** is greater than 10, or **0** otherwise:

```
import pyspark.sql.functions as F

#create new DataFrame
df_new = df.withColumn('B10', F.when((df.team=='B') & (df.points>10), 1).otherwise(0))

#view new DataFrame
df_new.show()
```

```
+----+--------+------+---+
|team|position|points|B10|
+----+--------+------+---+
| A| Guard| 11| 0|
| A| Guard| 8| 0|
| A| Forward| 22| 0|
| A| Forward| 22| 0|
| B| Guard| 14| 1|
| B| Guard| 14| 1|
| B| Forward| 13| 1|
| B| Forward| 7| 0|
+----+--------+------+---+
```

Notice that three values in the **B10** column contain **1** because only those three rows met the two conditions that we specified.

Also note that you could return 'Yes' or 'No' instead of **1** and **0** by using the following syntax:

import pyspark.sql.functions as F

#create new DataFrame
df_new = df.withColumn('B10', F.when((df.team=='B') & (df.points>10), 'Yes').otherwise('No'))

#view new DataFrame
df_new.show()

```
+----+--------+------+---+
|team|position|points|B10|
+----+--------+------+---+
| A| Guard| 11| No|
| A| Guard| 8| No|
| A| Forward| 22| No|
| A| Forward| 22| No|
| B| Guard| 14|Yes|
| B| Guard| 14|Yes|
| B| Forward| 13|Yes|
| B| Forward| 7| No|
+----+--------+------+---+
```

The new **B10** column now returns 'Yes' or 'No' instead of **1** or **0**.

Feel free to return whatever values you'd like by specifying them in the **when** and **otherwise** functions.

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