How can I use the unionByName() function in PySpark to combine two datasets while preserving the column names?

Authored by

stats writer

June 24, 2024

The unionByName() function in PySpark allows users to combine two datasets while retaining the original column names. This function is useful when working with datasets that have similar or identical column names, as it avoids conflicts and preserves the integrity of the data. By using unionByName(), users can easily merge two datasets without having to manually rename columns, making the data integration process more efficient and accurate. This function is particularly helpful for data analysts and data scientists working with large and complex datasets in PySpark.

The `pyspark.sql.DataFrame.unionByName()` to merge/union two DataFrames with column names. In PySpark you can easily achieve this using unionByName() transformation, this function also takes param `allowMissingColumns` with the value `True` if you have a different number of columns on two DataFrames.

1. Syntax of unionByName()

Following is the syntax of the unionByName()

```
# unionByName() Syntax
unionByName(df, allowMissingColumns=True)
```

2. Difference between PySpark unionByName() vs union()

The difference between `unionByName()` function and `union()` is that this function resolves columns by name (not by position). In other words, unionByName() is used to merge two DataFrames by column names instead of by position.

`unionByName`() also provides an argument allowMissingColumns to specify if you have a different column counts. In case you are using an older than Spark 3.1 version, use the below approach to merge DataFrames with different column names.

PySpark Merge DataFrames with Different Columns (Python Example)

3. PySpark unionByName() Usage with Examples

PySpark unionByName() is used to union two DataFrames when you have column names in a different order or even if you have missing columns in any DataFrme, in other words, this function resolves columns by name (not by position). First, let's create DataFrames with the different number of columns.

```
from pyspark.sql import SparkSession
```

```
spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()
```

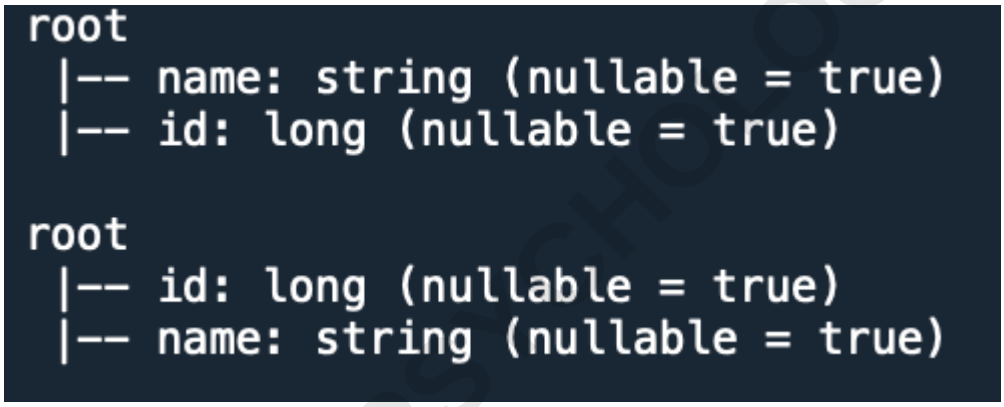
Create DataFrame df1 with columns name, and id
data =

df1 = spark.createDataFrame(data = data, schema=)
df1.printSchema()

Create DataFrame df2 with columns name and id
data2=

df2 = spark.createDataFrame(data = data2, schema =)
df2.printSchema()

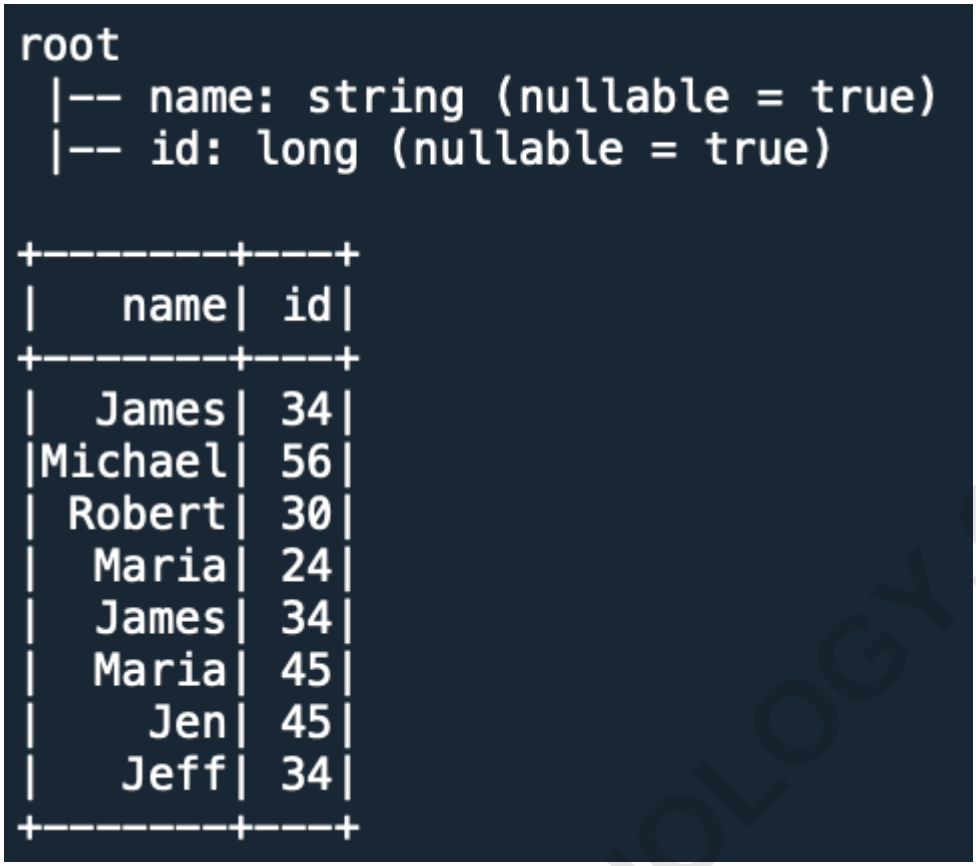
Yields below output.



Now let's use the PySpark unionByName() to union these two.

```
# unionByName() example
df3 = df1.unionByName(df2)
df3.printSchema
df3.show()
```

Yields below output.



4. Use unionByName() with Different Number of Columns

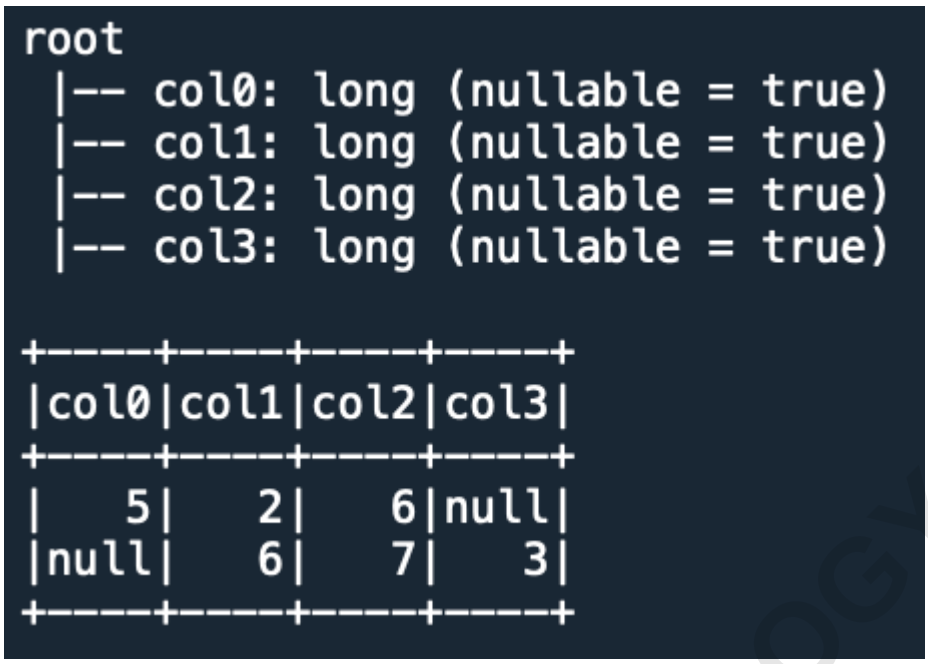
In the above example we have two DataFrames with the same column names but in different order. If you have a different number of columns then use `allowMissingColumns=True`. When using this, the result of the DataFrmae contains null values for the columns that are missing on the DataFrame.

Note that param allowMissingColumns is available since Spark 3.1 version.

```
# Create DataFrames with different column names
df1 = spark.createDataFrame(], )
df2 = spark.createDataFrame(], )
```

```
# Using allowMissingColumns
df3 = df1.unionByName(df2, allowMissingColumns=True)
df3.printSchema
df3.show()
```

Yields below output.



5. Complete Example of PySpark unionByName()

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('SparkByExamples.com').getOrCreate()
```

Create DataFrame df1 with columns name, and id
data =

df1 = spark.createDataFrame(data = data, schema=)
df1.printSchema()

Create DataFrame df2 with columns name and id
data2=

df2 = spark.createDataFrame(data = data2, schema =)
df2.printSchema()

# Using unionByName()
df3 = df1.unionByName(df2)
df3.printSchema()
df3.show()

Using allowMissingColumns

```
df1 = spark.createDataFrame(], )
df2 = spark.createDataFrame(], )
df3 = df1.unionByName(df2, allowMissingColumns=True)
df3.printSchema()
df3.show()
```

6. Conclusion

In this article, you have learned what is PySpark unionByName() and how it is different from union(). unionByName() is used to merge or union two DataFrames with different column names and a different number of columns.

Happy Learning !!

Related Articles