How can I use the PySpark “when otherwise” function to mimic SQL’s “case when” statement in my data analysis?

Authored by

stats writer

June 24, 2024

The PySpark library offers a powerful "when otherwise" function that can be used to mimic SQL's "case when" statement in data analysis. This function allows users to specify different conditions and corresponding actions, similar to the "case when" statement in SQL. By utilizing this function, users can efficiently handle data transformations and manipulations, making their data analysis process more efficient and effective. Additionally, the "when otherwise" function allows for easier implementation of complex logic and conditional statements, providing a more robust and versatile tool for data analysis.

PySpark When Otherwise and SQL Case When on DataFrame with Examples - Similar to SQL and programming languages, PySpark supports a way to check multiple conditions in sequence and returns a value when the first condition met by using **SQL like case when** and **when().otherwise()** expressions, these works similar to "`Switch`" and "`if then else`" statements.

PySpark When Otherwise - `when()` is a SQL function that returns a `Column` type and otherwise() is a function of Column, if otherwise() is not used, it returns a None/NULL value.

PySpark SQL Case When - This is similar to SQL expression, Usage: `CASE WHEN cond1 THEN result WHEN cond2 THEN result... ELSE result END`.

First, let's create a DataFrame

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

columns =
df = spark.createDataFrame(data = data, schema = columns)
df.show()
+-------+------+------+
| name|gender|salary|
+-------+------+------+
| James| M| 60000|
|Michael| M| 70000|
| Robert| null|400000|
| Maria| F|500000|
| Jen| | null|
+-------+------+------+
```

1. Using when()otherwise() on PySpark DataFrame.

PySpark `when()` is SQL function, in order to use this first you should import and this returns a Column type, `otherwise()` is a function of `Column`, when otherwise() not used and none of the conditions met it assigns None (Null) value. Usage would be like `when(condition).otherwise(default)`.

when() function take 2 parameters, first param takes a condition and second takes a literal value or Column, if condition evaluates to true then it returns a value from second param.

The below code snippet replaces the value of gender with a new derived value, when conditions not matched, we are assigning "Unknown" as value, for null assigning empty.

```
from pyspark.sql.functions import when
df2 = df.withColumn("new_gender", when(df.gender == "M","Male")
.when(df.gender == "F","Female")
.when(df.gender.isNull() ,"")
.otherwise(df.gender))
df2.show()
+-------+------+------+----------+
| name|gender|salary|new_gender|
+-------+------+------+----------+
| James| M| 60000| Male|
|Michael| M| 70000| Male|
| Robert| null|400000| |
| Maria| F|500000| Female|
| Jen| | null| |
+-------+------+------+----------+
```

Using with select()

```
df2=df.select(col("*"),when(df.gender == "M","Male")
.when(df.gender == "F","Female")
.when(df.gender.isNull() ,"")
.otherwise(df.gender).alias("new_gender"))
```

This yields same output as above.

2. PySpark SQL Case When on DataFrame.

If you have a SQL background you might have familiar with Case When statement that is used to execute a sequence of conditions and returns a value when the first condition met, similar to SWITH and IF THEN ELSE statements. Similarly, PySpark SQL Case When statement can be used on DataFrame, below are some of the examples of using with withColumn(), select(), selectExpr() utilizing expr() function.

Syntax of SQL CASE WHEN ELSE END

```
CASE
WHEN condition1 THEN result_value1
WHEN condition2 THEN result_value2
-----
-----
ELSE result
END;
```

2.1 Using Case When Else on DataFrame using withColumn() & select()

Below example uses PySpark SQL expr() Function to express SQL like expressions.

```
from pyspark.sql.functions import expr, col
```

```
#Using Case When on withColumn()
df3 = df.withColumn("new_gender", expr("CASE WHEN gender = 'M' THEN 'Male' " +
"WHEN gender = 'F' THEN 'Female' WHEN gender IS NULL THEN "" +
"ELSE gender END"))
df3.show(truncate=False)
+-------+------+------+----------+
|name |gender|salary|new_gender|
+-------+------+------+----------+
|James |M |60000 |Male |
|Michael|M |70000 |Male |
|Robert |null |400000| |
|Maria |F |500000|Female |
|Jen | |null | |
+-------+------+------+----------+
```

#Using Case When on select()
df4 = df.select(col("*"), expr("CASE WHEN gender = 'M' THEN 'Male' " +
"WHEN gender = 'F' THEN 'Female' WHEN gender IS NULL THEN "" +
"ELSE gender END").alias("new_gender"))

2.2 Using Case When on SQL Expression

You can also use Case When with SQL statement after creating a temporary view. This returns a similar output as above.

```
df.createOrReplaceTempView("EMP")
spark.sql("select name, CASE WHEN gender = 'M' THEN 'Male' " +
"WHEN gender = 'F' THEN 'Female' WHEN gender IS NULL THEN ''" +
"ELSE gender END as new_gender from EMP").show()
```

2.3. Multiple Conditions using & and | operator

We often need to check with multiple conditions, below is an example of using PySpark When Otherwise with multiple conditions by using and (&) or (|) operators. To explain this I will use a new set of data to make it simple.

```
df5.withColumn("new_column", when((col("code") == "a") | (col("code") == "d"),
"A")
.when((col("code") == "b") & (col("amt") == "4"), "B")
.otherwise("A1")).show()
```

Output:

```
+---+----+---+----------+
| id|code|amt|new_column|
+---+----+---+----------+
| 66| a| 4| A|
| 67| a| 0| A|
| 70| b| 4| B|
| 71| d| 4| A|
+---+----+---+----------+
```

Complete Example - PySpark When Otherwise | SQL Case When

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

columns =
df = spark.createDataFrame(data = data, schema = columns)
df.show()

#Using When otherwise
from pyspark.sql.functions import when,col
df2 = df.withColumn("new_gender", when(df.gender == "M","Male")
.when(df.gender == "F","Female")
.when(df.gender.isNull() ,"")
.otherwise(df.gender))
df2.show()

df2=df.select(col("*"),when(df.gender == "M","Male")
.when(df.gender == "F","Female")
.when(df.gender.isNull() ,"")
.otherwise(df.gender).alias("new_gender"))
df2.show()
# Using SQL Case When
from pyspark.sql.functions import expr
df3 = df.withColumn("new_gender", expr("CASE WHEN gender = 'M' THEN 'Male' " +
"WHEN gender = 'F' THEN 'Female' WHEN gender IS NULL THEN "" +
"ELSE gender END"))
df3.show()

df4 = df.select(col("*"), expr("CASE WHEN gender = 'M' THEN 'Male' " +
"WHEN gender = 'F' THEN 'Female' WHEN gender IS NULL THEN "" +
"ELSE gender END").alias("new_gender"))

df.createOrReplaceTempView("EMP")
spark.sql("select name, CASE WHEN gender = 'M' THEN 'Male' " +
"WHEN gender = 'F' THEN 'Female' WHEN gender IS NULL THEN "" +
"ELSE gender END as new_gender from EMP").show()
```

Conclusion:

In this article, you have learned how to use Pyspark SQL "`case when`" and "`when otherwise`" on Dataframe by leveraging example like checking with NUll/None, applying with multiple conditions using AND (&), OR (|) logical operators. I hope you like this article.

Happy Learning !!

References

Related Articles