How can I use the substring() function in Pyspark to extract a specific portion of a column’s data?

Authored by

stats writer

June 24, 2024

The substring() function in Pyspark allows you to extract a specific portion of a column's data by specifying the starting and ending positions of the desired substring. This function is useful for manipulating and analyzing a large dataset, as it allows you to extract relevant information from a column and create new columns based on the extracted data. By utilizing the substring() function, you can efficiently extract and manipulate data in Pyspark, making it easier to perform various data analysis tasks.

The PySpark `substring()` function extracts a portion of a string column in a DataFrame. It takes three parameters: the column containing the string, the starting index of the substring (1-based), and optionally, the length of the substring. If the length is not specified, the function extracts from the starting index to the end of the string.

This function is useful for text manipulation tasks such as extracting substrings based on position within a string column. It operates similarly to the `SUBSTRING()` function in SQL and enables efficient string processing within PySpark DataFrames.

In this tutorial, I have explained with an example of getting substring of a column using `substring()` from pyspark.sql.functions and using `substr()` from `pyspark.sql.Column` type.

PySpark substring()

The `substring()` function is from `pyspark.sql.functions` module hence, to use this function, first you need to import this. Following is the syntax.

```
#Syntax
substring(str, pos, len)
```

Here,

The following example demonstrates using substring() with withColumn().

```
# Imports
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, substring
```

spark=SparkSession.builder.appName("stringoperations").getOrCreate()

Create Sample Data
data =
columns=

```
df=spark.createDataFrame(data,columns)

# Using substring()
df.withColumn('year', substring('date', 1,4))
.withColumn('month', substring('date', 5,2))
.withColumn('day', substring('date', 7,2))
df.printSchema()
df.show(truncate=False)
```

In the example above, we've created a DataFrame with two columns: `id` and `date`. The `date` column is formatted as "year month day". We used the `substring()` function on the `date` column to extract the year, month, and day as separate substrings. Below is the output.

```
# Output
+---+--------+----+-----+---+
|id |date    |year|month|day|
+---+--------+----+-----+---+
|1  |20200828|2020|08   |28 |
|2  |20180525|2018|05   |25 |
+---+--------+----+-----+---+
```

2. Using substring() with select()

In Pyspark, we can also get the substring() of a column using select(). The above example can be written as follows.

```
# substring() with select()
df.select('date', substring('date', 1,4).alias('year'),
substring('date', 5,2).alias('month'),
substring('date', 7,2).alias('day'))
```

3. Using substring() with selectExpr()

Sample example using `selectExpr()` to get a substring of column(date) as year, month, day. Below is the code that gives the same output as above.

```
# substring() with selectExpr()
```

```
df.selectExpr('date', 'substring(date, 1,4) as year',
'substring(date, 5,2) as month',
'substring(date, 7,2) as day')
```

4. Using substr() from Column type

Below is the example of getting substring using `substr()` function from `pyspark.sql.Column` type in Pyspark.

```
# Using substr()
df3=df.withColumn('year', col('date').substr(1, 4))
.withColumn('month',col('date').substr(5, 2))
.withColumn('day', col('date').substr(7, 2))
```

The above example gives output the same as the above-mentioned examples.

Complete Example of PySpark substring()

```
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, substring
spark=SparkSession.builder.appName("stringoperations").getOrCreate()
data =
columns=
df=spark.createDataFrame(data,columns)
```

#Using SQL function substring()
df.withColumn('year', substring('date', 1,4))
.withColumn('month', substring('date', 5,2))
.withColumn('day', substring('date', 7,2))
df.printSchema()
df.show(truncate=False)

#Using select
df1=df.select('date', substring('date', 1,4).alias('year'),
substring('date', 5,2).alias('month'),
substring('date', 7,2).alias('day'))

#Using with selectExpr

```
df2=df.selectExpr('date', 'substring(date, 1,4) as year',
'substring(date, 5,2) as month',
'substring(date, 7,2) as day')

#Using substr from Column type
df3=df.withColumn('year', col('date').substr(1, 4))
.withColumn('month',col('date').substr(5, 2))
.withColumn('day', col('date').substr(7, 2))

df3.show()
```

Conclusion

In this session, we have learned different ways of getting substring of a column in PySpark DataFarme. I hope you liked it! Keep practicing. And do comment in the comment section for any kind of questions!!

Related Articles