How can I convert a timestamp to a date using the to_date() function in PySpark?

Authored by

stats writer

June 24, 2024

The to_date() function in PySpark is used to convert a timestamp to a date format. This function takes in a timestamp column and returns a new column containing the corresponding date values. It follows the standard SQL date format of YYYY-MM-DD. This conversion is useful in cases where only the date component is needed for further analysis or visualization. By using the to_date() function, timestamps can easily be converted to a more user-friendly date format in PySpark.

PySpark functions provide to_date() function to convert timestamp to date (DateType), this ideally achieved by just truncating the time part from the Timestamp column. In this tutorial, I will show you a PySpark example of how to convert timestamp to date on DataFrame & SQL.

to_date() - function formats Timestamp to Date.

```
Syntax: to_date(timestamp_column)
Syntax: to_date(timestamp_column,format)
```

PySpark timestamp (`TimestampType`) consists of value in the format `yyyy-MM-dd HH:mm:ss.SSSS` and Date (`DateType`) format would be `yyyy-MM-dd`. Use to_date() function to truncate time from Timestamp or to convert the timestamp to date on DataFrame column.

```
df=spark.createDataFrame(
data = ,
schema=)
df.printSchema()
```

#Displays
root
|-- id: string (nullable = true)
|-- input_timestamp: string (nullable = true)

Using to_date() - Convert Timestamp String to Date

In this example, we will use `to_date()` function to convert `TimestampType` (or string) column to `DateType` column. The input to this function should be timestamp column or string in TimestampType format and it returns just date in DateType column.

```
from pyspark.sql.functions import *
```

#Timestamp String to DateType

df.withColumn("date_type",to_date("input_timestamp"))
.show(truncate=False)

#Timestamp Type to DateType
df.withColumn("date_type",to_date(current_timestamp()))
.show(truncate=False)

#Above Both examples display
+---+----------------------+----------+
|id |input_timestamp |date_type |
+---+----------------------+----------+
|1 |2019-06-24 12:01:19.000|2019-06-24|
+---+----------------------+----------+

#Custom Timestamp format to DateType
df.select(to_date(lit('06-24-2019 12:01:19.000'),'MM-dd-yyyy HH:mm:ss.SSSS'))
.show()

#Displays
+---+
|to_date('06-24-2019 12:01:19.000', 'MM-dd-yyyy HH:mm:ss.SSSS')|
+---+
| 2019-06-24|
+---+

Convert TimestampType (timestamp) to DateType (date)

This example converts the PySpark TimestampType column to DateType.

```
#Timestamp type to DateType
df.withColumn("ts",to_timestamp(col("input_timestamp")))
.withColumn("datetype",to_date(col("ts")))
.show(truncate=False)
```

#Displays
+---+----------------------+-------------------+----------+
|id |input_timestamp |ts |datetype |
+---+----------------------+-------------------+----------+
|1 |2019-06-24 12:01:19.000|2019-06-24 12:01:19|2019-06-24|
+---+----------------------+-------------------+----------+

Using Column cast() Function

Here is another way to convert TimestampType (timestamp string) to DateType using `cast` function.

```
# Using Cast to convert Timestamp String to DateType
df.withColumn('date_type', col('input_timestamp').cast('date'))
.show(truncate=False)
```

# Using Cast to convert TimestampType to DateType
df.withColumn('date_type', to_timestamp('input_timestamp').cast('date'))
.show(truncate=False)

#Displays
+---+----------------------+----------+
|id |input_timestamp |date_type |
+---+----------------------+----------+
|1 |2019-06-24 12:01:19.000|2019-06-24|
+---+----------------------+----------+

PySpark SQL - Convert Timestamp to Date

Following are similar examples using with PySpark SQL. If you are from an SQL background these come in handy.

```
#SQL TimestampType to DateType
spark.sql("select to_date(current_timestamp) as date_type")
```

#SQL CAST TimestampType to DateType
spark.sql("select date(to_timestamp('2019-06-24 12:01:19.000')) as date_type")

#SQL CAST timestamp string to DateType
spark.sql("select date('2019-06-24 12:01:19.000') as date_type")

#SQL Timestamp String (default format) to DateType
spark.sql("select to_date('2019-06-24 12:01:19.000') as date_type")

#SQL Custom Timeformat to DateType
spark.sql("select to_date('06-24-2019 12:01:19.000','MM-dd-yyyy HH:mm:ss.SSSS') as date_type")

Complete code

```
from pyspark.sql import SparkSession

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

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

from pyspark.sql.functions import *

# Using Cast to convert Timestamp String to DateType
df.withColumn('date_type', col('input_timestamp').cast('date'))
.show(truncate=False)

# Using Cast to convert TimestampType to DateType
df.withColumn('date_type', to_timestamp('input_timestamp').cast('date'))
.show(truncate=False)

df.select(to_date(lit('06-24-2019 12:01:19.000'),'MM-dd-yyyy HH:mm:ss.SSSS'))
.show()

#Timestamp String to DateType
df.withColumn("date_type",to_date("input_timestamp"))
.show(truncate=False)

#Timestamp Type to DateType
df.withColumn("date_type",to_date(current_timestamp()))
.show(truncate=False)

df.withColumn("ts",to_timestamp(col("input_timestamp")))
.withColumn("datetype",to_date(col("ts")))
.show(truncate=False)

#SQL TimestampType to DateType
spark.sql("select to_date(current_timestamp) as date_type")
#SQL CAST TimestampType to DateType
```

```
spark.sql("select date(to_timestamp('2019-06-24 12:01:19.000')) as date_type")
#SQL CAST timestamp string to DateType
spark.sql("select date('2019-06-24 12:01:19.000') as date_type")
#SQL Timestamp String (default format) to DateType
spark.sql("select to_date('2019-06-24 12:01:19.000') as date_type")
#SQL Custom Timeformat to DateType
spark.sql("select to_date('06-24-2019 12:01:19.000','MM-dd-yyyy HH:mm:ss.SSSS') as date_type")
```

In this example, you have learned how to cast the timestamp to date column using `to_date()` and `cast` functions.

Happy Learning !!

Related Articles: