What are the available date and timestamp functions in PySpark SQL?

Authored by

stats writer

June 24, 2024

PySpark SQL is a powerful tool for working with structured data in the Apache Spark framework. It provides a wide range of date and timestamp functions that allow users to manipulate and analyze temporal data with ease. Some of the available functions include date and time extraction, formatting, conversion, and arithmetic operations. These functions enable users to query, filter, and transform data based on date and time values, making it easier to perform complex analyses and derive meaningful insights from temporal data. With a comprehensive set of date and timestamp functions, PySpark SQL offers a flexible and efficient solution for managing and processing temporal data in a variety of applications.

PySpark Date and Timestamp Functions are supported on DataFrame and SQL queries and they work similarly to traditional SQL, Date and Time are very important if you are using PySpark for ETL. Most of all these functions accept input as, Date type, Timestamp type, or String. If a String used, it should be in a default format that can be cast to date.

PySpark SQL provides several Date & Timestamp functions hence keep an eye on and understand these. Always you should choose these functions instead of writing your own functions (UDF) as these functions are compile-time safe, handles null, and perform better when compared to PySpark UDF. If your PySpark application is critical on performance try to avoid using custom UDF at all costs as these are not guarantee performance.

For readable purposes, I've grouped these functions into the following groups.

Before you use any examples below, make sure you Create PySpark Sparksession and import SQL functions.

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


PySpark SQL Date Functions

Below are some of the PySpark SQL Date functions, these functions operate on the just Date.

The default format of the PySpark Date is `yyyy-MM-dd`.

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
current_date()	Returns the current date as a `DateType` object. This function does not take any arguments and simply returns the current date based on the system clock where the PySpark application is running.
date_format()	It is used to format a date or timestamp column in a DataFrame to a specified date or time format pattern.

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
to_date()	It converts a string column representing a date or timestamp into a date type column in a DataFrame
add_months()	It is used to add or subtract a specified number of months to a date or timestamp column in a DataFrame. It takes two arguments: the column representing the date or timestamp, and the number of months to add or subtract.
date_add(column, days) date_sub(column, days)	date_add() is used to add a specified number of days to a date column date_sub() is used to subtract a specified number of days from a date column
datediff(end, start)	It is used to calculate the difference in days between two date columns in a DataFrame. It takes two arguments: the two date columns to calculate the difference between.
months_between(end, start)	It is used to calculate the difference in months between two date or timestamp columns in a DataFrame. It takes two arguments: the two date or timestamp columns to calculate the difference between them. If both inputs share the same day of the month or are both the last day of their respective months, a whole number is returned. Otherwise, the difference is computed under the assumption of 31 days per month.
months_between(end, start, roundOff)	The result is rounded off to 8 digits when `roundOff` is set to true, it is not rounded otherwise.
next_day(column, dayOfWeek)	It is used to find the first occurrence of a specified day of the week that comes after a given date. It takes two arguments, the date column representing the reference date, and the day of the week specified as a string (e.g., 'Monday', 'Tuesday', etc.).
trunc(column, format)	Truncate a date or timestamp column in a DataFrame to a specified level of granularity. For example, `trunc(df, 'month')` would truncate the dates in the "date_column" of DataFrame "df" to the first day of the month, effectively removing the day component and retaining only the month and year.
date_trunc(format, timestamp)	Truncate a timestamp column in a DataFrame to a specified level of granularity, while preserving the timestamp type. It takes two arguments: the column representing the timestamp to be truncated, and the level of granularity to which the timestamp should be truncated.
year(column)	Returns the year from a given date or timestamp.
quarter(column)	Returns the quarter as an integer from a given date or timestamp.
month(column)	Returns the month as an integer from a given date or timestamp
dayofweek(column)	Extract the day of the week from a date or timestamp column in a DataFrame. Monday is represented by 1, Tuesday by 2, and so on until Sunday, which is represented by 7.
dayofmonth(column)	Extracts the day of the month from a given date or timestamp.
dayofyear(column)	Extracts the day of the year from a given date or timestamp.

PYSPARK DATE FUNCTION	DATE FUNCTION DESCRIPTION
weekofyear(column)	Extract the week number from a date or timestamp column in a DataFrame.
last_day(column)	Return the last day of the month for a given date or timestamp column.The result is a date column where each date corresponds to the last day of the month for the original dates in the specified column.
from_unixtime(column)	Convert a Unix timestamp (represented as the number of seconds since the Unix epoch) to a timestamp column
unix_timestamp()	It is used to convert a string representing a date or timestamp to a Unix timestamp (i.e., the number of seconds since the Unix epoch). It takes two arguments: the column containing the string representation of the date or timestamp, and the format string specifying the format of the input string.

PySpark SQL Timestamp Functions

Below are some of the PySpark SQL Timestamp functions, these functions operate on both date and timestamp values.

The default format of the Spark Timestamp is `yyyy-MM-dd HH:mm:ss.SSSS`

PYSPARK TIMESTAMP FUNCTION SIGNATURE	TIMESTAMP FUNCTION DESCRIPTION
current_timestamp()	It is used to retrieve the current timestamp at the time of execution within the PySpark application. It does not require any arguments. When called, it returns the current timestamp with timezone information as a timestamp type.
hour(column)	Return the hours from a timestamp column
minute(column)	Return the hours from a timestamp column
second(column)	Return the seconds from a timestamp column
to_timestamp(column) to_timestamp(column, fmt)	Convert a string column representing a date or timestamp to a timestamp column in a DataFrame. It takes two arguments: the column containing the string representation of the date or timestamp, and the format string specifying the format of the input string.

Date and Timestamp Window Functions

Below are PySpark Data and Timestamp window functions.