Lectura: Esquema Definido por el Usuario (UDS) para DSL y SQL

Tiempo estimado necesario: 10 minutos

¿Cómo definir y aplicar un esquema definido por el usuario en PySpark?

En esta lectura, aprenderás cómo definir y aplicar un esquema definido por el usuario en PySpark.

Spark proporciona un marco de procesamiento de datos estructurados que puede definir y aplicar esquemas para diversas fuentes de datos, incluidos los archivos CSV. Veamos los pasos para definir y usar un esquema definido por el usuario para un archivo CSV en PySpark:

Paso 1:

Importar las bibliotecas requeridas.

```
from pyspark.sql.types import StructType, IntegerType, FloatType, StringType, StructField
```

Paso 2:

Define el esquema.

Entender los datos antes de definir un esquema es un paso importante.

Veamos el enfoque paso a paso para entender los datos y definir un esquema apropiado para un archivo de entrada dado:

1. **Explorar los datos:** Comprender los diferentes tipos de datos presentes en cada columna.
2. **Tipos de datos de las columnas:** Determinar los tipos de datos apropiados para cada columna según los valores observados.
3. **Definir el esquema:** Utiliza la clase 'StructType' en Spark y crea un 'StructField' para cada columna, mencionando el nombre de la columna, el tipo de dato y otras propiedades.

Ejemplo:

```
schema = StructType([
    StructField("Emp_Id", StringType(), False),
    StructField("Emp_Name", StringType(), False),
    StructField("Department", StringType(), False),
    StructField("Salary", IntegerType(), False),
    StructField("Phone", IntegerType(), True),
])
```

'False' indica que los valores nulos **NO** están permitidos para la columna.

El esquema definido arriba se puede utilizar para los datos del archivo CSV a continuación:

Nombre del archivo: employee.csv

```
emp_id,emp_name,dept,salary,phone
A101,jhon,computer science,1000,+1 (701) 846 958
A102,Peter,Electronics,2000,
A103,Micheal,IT,2500,
```

Paso 3: Lee el archivo de entrada con un esquema definido por el usuario.

```
#create a dataframe on top a csv file
df = (spark.read
```

```
  .format("csv")
  .schema(schema)
  .option("header", "true")
  .load("employee.csv")
)
# display the dataframe content
df.show()
```

Paso 4: Usa el `printSchema()` método en Spark para mostrar el esquema de un DataFrame y asegurar que el esquema se aplique correctamente a los datos.

```
df.printSchema()
```

A través de los cuatro pasos anteriores, has adquirido la capacidad de establecer un esquema para un archivo CSV. Además, has utilizado este esquema definido por el usuario (UDF) para leer el archivo CSV, exhibir su contenido y mostrar el esquema en sí.

Autor(es)

- Raghul Ramesh