ARRAY_TYPE COLUMNS IN PYSPARK

ArrayType Columns in pySpark

Cmd 2

```
from pyspark.sql.types import StructType,StructField,StringType,ArrayType,IntegerType,MapType
from pyspark.sql import SparkSession
from pyspark.sql.functions import col,explode,array_contains

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

data = [("Alice", [25, 30, 28]),
        ("Bob", [22, 24]),
        ("Carol", [29])]

schema=StructType([
        StructField('name',StringType()),
        StructField('numbers',ArrayType(IntegerType()))
        ])
df = spark.createDataFrame(data, schema)
df.withColumn('firstNumbers',col("numbers")[1]).show()
```

▸ (3) Spark Jobs

▸ df: pyspark.sql.dataframe.DataFrame = [name: string, numbers: array]

```
+-----+------------+------------+
| name|     numbers|firstNumbers|
+-----+------------+------------+
|Alice|[25, 30, 28]|          30|
|  Bob|    [22, 24]|          24|
|Carol|        [29]|        null|
+-----+------------+------------+
```

EXPLODE FUNCTION

It is to create a new row for each element in the given array column.

explode function

Cmd 4

```
df.select(col("name"), explode(col("numbers")).alias("Age")).show()
```

▸ (3) Spark Jobs

```
+-----+---+
| name|Age|
+-----+---+
|Alice| 25|
|Alice| 30|
|Alice| 28|
|  Bob| 22|
|  Bob| 24|
|Carol| 29|
+-----+---+
```

USAGE OF ARRAY_CONTAINS

array_contains

Cmd 6

```
df.withColumn("Has age 25",array_contains(col("numbers"),25)).show()
```

(3) Spark Jobs

```
+-----+------------+----------+
| name|     numbers|Has age 25|
+-----+------------+----------+
|Alice|[25, 30, 28]|      true|
|  Bob|    [22, 24]|     false|
|Carol|        [29]|     false|
+-----+------------+----------+
```

MAP_TYPE COLUMN IN PYSPARK

It is used to represent map key-value pair.

MapType column

Cmd 8

```
data = [("Alice", {"age": 25, "city": "New York"}),
        ("Bob", {"age": 22, "city": "San Francisco"}),
        ("Carol", {"age": 29, "city": "Seattle"})]


schema=StructType([
        StructField('name',StringType()),
        StructField('info',MapType(StringType(),StringType()))
        ])

df1=spark.createDataFrame(data,schema)
df1.show(truncate=False)
```

(3) Spark Jobs

df1: pyspark.sql.dataframe.DataFrame = [name: string, info: map]

```
+-----+-----------------------------------+
|name |info                               |
+-----+-----------------------------------+
|Alice|{city -> New York, age -> 25}      |
|Bob  |{city -> San Francisco, age -> 22}|
|Carol|{city -> Seattle, age -> 29}       |
+-----+-----------------------------------+
```

```
df2=df1.withColumn("age",df1.info["age"])
display(df2)
```

(3) Spark Jobs

df2: pyspark.sql.dataframe.DataFrame = [name: string, info: map ... 1 more field]

Table +

	name	info	age
1	Alice	{"city": "New York", "age": "25"}	25
2	Bob	{"city": "San Francisco", "age": "22"}	22
3	Carol	{"city": "Seattle", "age": "29"}	29