Master PySpark: From Zero to Big Data Hero!!

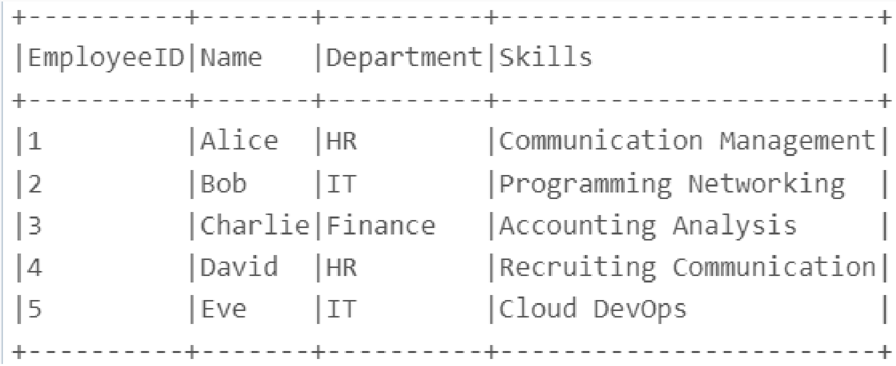
Split Function In Dataframe

Let's create a PySpark DataFrame for employee data, which will include columns such as EmployeeID, Name, Department, and Skills.

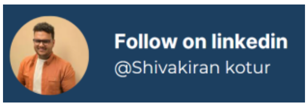
I'll demonstrate the usage of the split, explode, and other relevant PySpark functions with the employee data, along with notes for each operation.

Sample Data Creation for Employee Data

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import split, explode, size,
array_contains, col
# Sample employee data
data = [
    (1, "Alice", "HR", "Communication Management"),
    (2, "Bob", "IT", "Programming Networking"),
    (3, "Charlie", "Finance", "Accounting Analysis"),
    (4, "David", "HR", "Recruiting Communication"),
    (5, "Eve", "IT", "Cloud DevOps")
]

# Define the schema
columns = ["EmployeeID", "Name", "Department", "Skills"]

# Create DataFrame
df = spark.createDataFrame(data, columns)

# Display the original DataFrame
df.show(truncate=False)
```



```
+----------+-------+----------+------------------------+
|EmployeeID|Name   |Department|Skills                  |
+----------+-------+----------+------------------------+
|1         |Alice  |HR        |Communication Management|
|2         |Bob    |IT        |Programming Networking  |
|3         |Charlie|Finance   |Accounting Analysis     |
|4         |David  |HR        |Recruiting Communication|
|5         |Eve    |IT        |Cloud DevOps            |
+----------+-------+----------+------------------------+
```



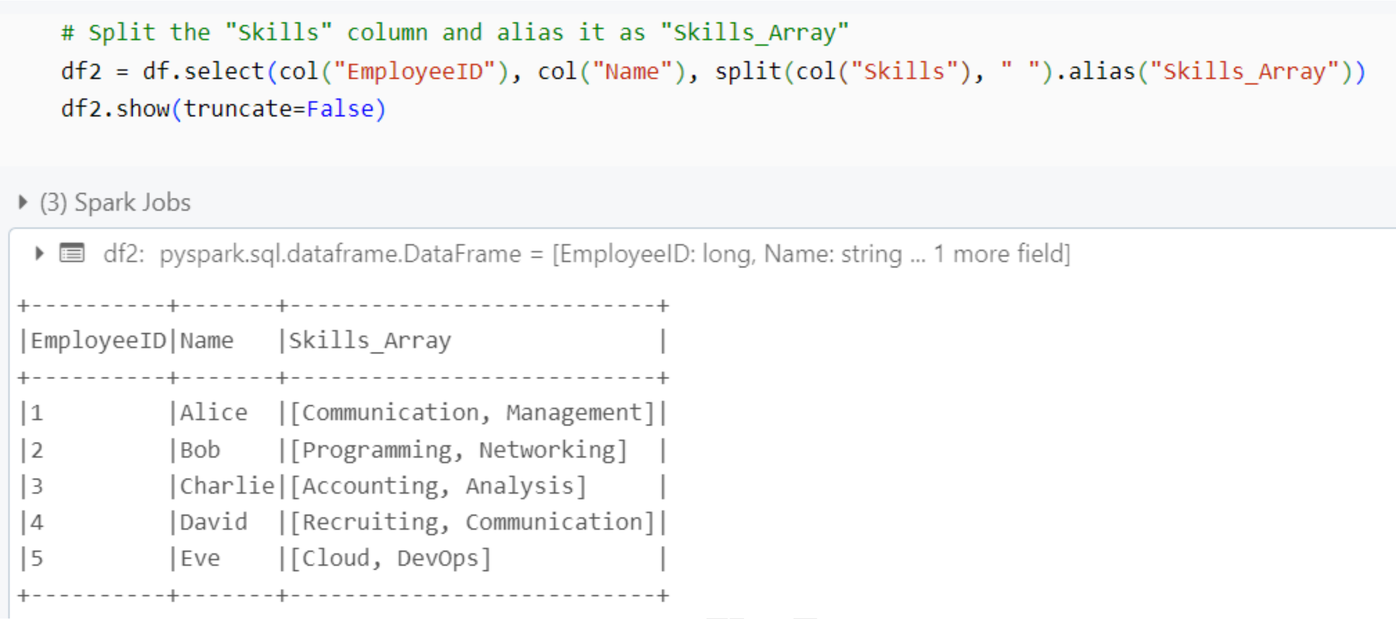
Follow me on LinkedIn – Shivakiran kotur

Notes with Examples

1. Split the "Skills" column:

We will split the Skills column into an array, where each skill is separated by a space.

python



```python
# Split the "Skills" column and alias it as "Skills_Array"
df2 = df.select(col("EmployeeID"), col("Name"), split(col("Skills"), " ").alias("Skills_Array"))
df2.show(truncate=False)
```

(3) Spark Jobs

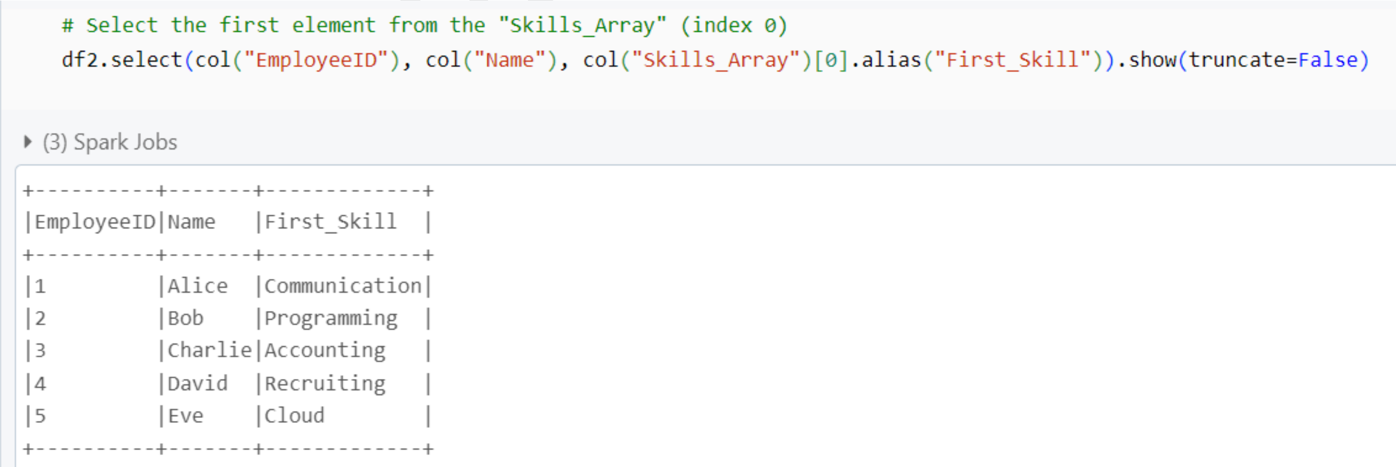
df2: pyspark.sql.dataframe.DataFrame = [EmployeeID: long, Name: string ... 1 more field]

```
+----------+-------+-------------------------+
|EmployeeID|Name   |Skills_Array             |
+----------+-------+-------------------------+
|1         |Alice  |[Communication, Management]|
|2         |Bob    |[Programming, Networking]  |
|3         |Charlie|[Accounting, Analysis]     |
|4         |David  |[Recruiting, Communication]|
|5         |Eve    |[Cloud, DevOps]            |
+----------+-------+-------------------------+
```

Note: This splits the Skills column into an array of skills based on the space separator. The alias("Skills_Array") gives the resulting array a meaningful name.

2. Select the first skill from the "Skills_Array":

You can select specific elements from an array using index notation. In this case, we'll select the first skill from the Skills_Array.

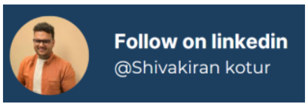


```python
# Select the first element from the "Skills_Array" (index 0)
df2.select(col("EmployeeID"), col("Name"), col("Skills_Array")[0].alias("First_Skill")).show(truncate=False)
```

(3) Spark Jobs

```
+----------+-------+-------------+
|EmployeeID|Name   |First_Skill  |
+----------+-------+-------------+
|1         |Alice  |Communication|
|2         |Bob    |Programming  |
|3         |Charlie|Accounting   |
|4         |David  |Recruiting   |
|5         |Eve    |Cloud        |
+----------+-------+-------------+
```

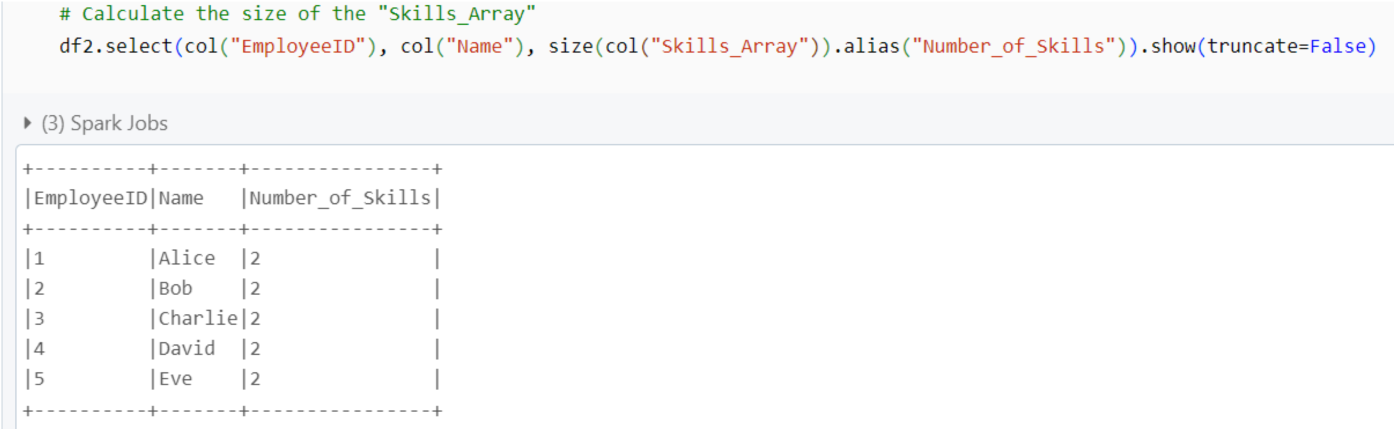
Note: The array index starts from 0, so Skills_Array[0] gives the first skill for each employee.



Follow me on LinkedIn – Shivakiran kotur

3. Calculate the size of the "Skills_Array":

We can calculate how many skills each employee has by using the size() function.



```
# Calculate the size of the "Skills_Array"
df2.select(col("EmployeeID"), col("Name"), size(col("Skills_Array")).alias("Number_of_Skills")).show(truncate=False)
```

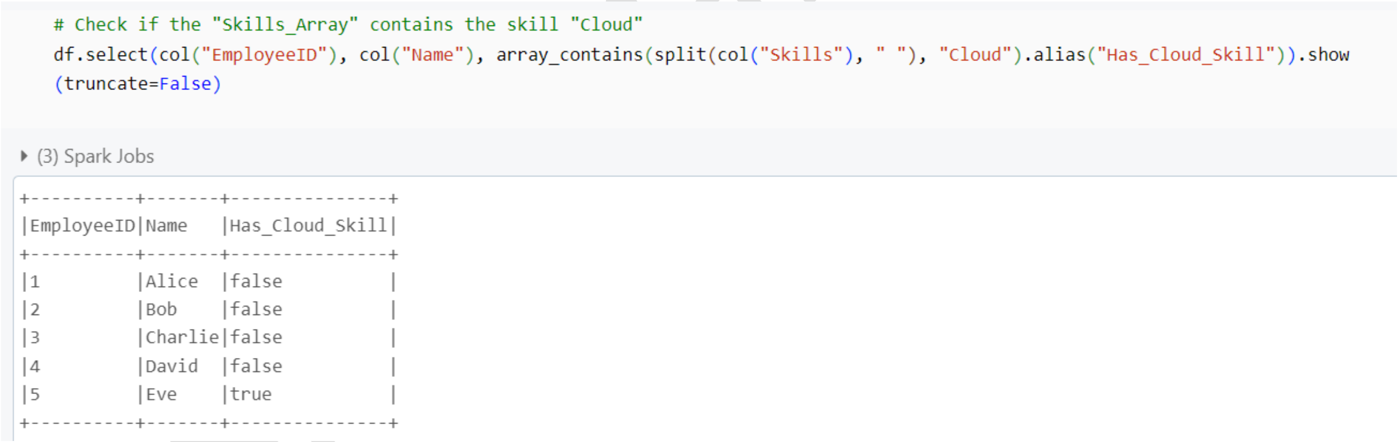
▸ (3) Spark Jobs

```
+----------+-------+----------------+
|EmployeeID|Name   |Number_of_Skills|
+----------+-------+----------------+
|1         |Alice  |2               |
|2         |Bob    |2               |
|3         |Charlie|2               |
|4         |David  |2               |
|5         |Eve    |2               |
+----------+-------+----------------+
```

Note: The size() function returns the number of elements (skills) in the Skills_Array.

4. Check if the array contains a specific skill:

We can check if a particular skill (e.g., "Cloud") is present in the employee's skillset using the array_contains() function.

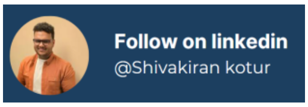


```
# Check if the "Skills_Array" contains the skill "Cloud"
df.select(col("EmployeeID"), col("Name"), array_contains(split(col("Skills"), " "), "Cloud").alias("Has_Cloud_Skill")).show
(truncate=False)
```

▸ (3) Spark Jobs

```
+----------+-------+---------------+
|EmployeeID|Name   |Has_Cloud_Skill|
+----------+-------+---------------+
|1         |Alice  |false          |
|2         |Bob    |false          |
|3         |Charlie|false          |
|4         |David  |false          |
|5         |Eve    |true           |
+----------+-------+---------------+
```

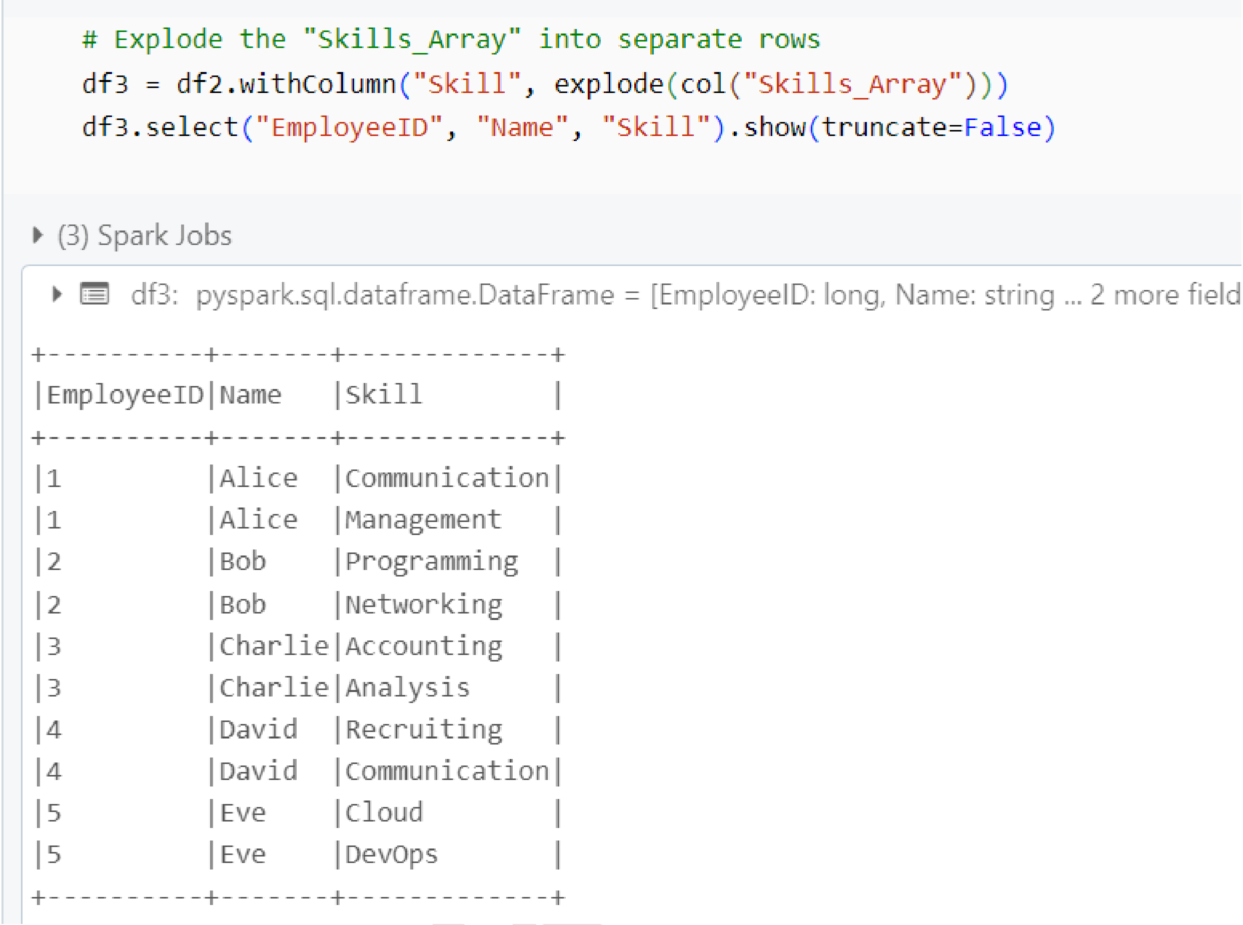
Note: This returns a boolean indicating whether the array contains the specified skill, "Cloud", for each employee.



Follow me on LinkedIn – Shivakiran kotur

5. Use the explode function to transform array elements into individual rows:

The explode() function can be used to flatten the array into individual rows, where each skill becomes a separate row for the employee.



```python
# Explode the "Skills_Array" into separate rows
df3 = df2.withColumn("Skill", explode(col("Skills_Array")))
df3.select("EmployeeID", "Name", "Skill").show(truncate=False)
```

▸ (3) Spark Jobs

▸ df3: pyspark.sql.dataframe.DataFrame = [EmployeeID: long, Name: string ... 2 more field

```
+----------+-------+-------------+
|EmployeeID|Name   |Skill        |
+----------+-------+-------------+
|1         |Alice  |Communication|
|1         |Alice  |Management   |
|2         |Bob    |Programming  |
|2         |Bob    |Networking   |
|3         |Charlie|Accounting   |
|3         |Charlie|Analysis     |
|4         |David  |Recruiting   |
|4         |David  |Communication|
|5         |Eve    |Cloud        |
|5         |Eve    |DevOps       |
+----------+-------+-------------+
```

Note: The explode() function takes an array column and creates a new row for each element of the array. Here, each employee will have multiple rows, one for each skill.

Summary of Key Functions:

- **split()**: This splits a column's string value into an array based on a specified delimiter (in this case, a space).
- **explode()**: Converts an array column into multiple rows, one for each element in the array.
- **size()**: Returns the number of elements in an array.
- **array_contains()**: Checks if a specific value exists in the array.
- **selectExpr()**: Allows you to use SQL expressions (like array[0]) to select array elements.