Spark Merge Two DataFrames with Different Columns or Schema

In [1]:

```
import pyspark
from pyspark.sql import SparkSession
spark = SparkSession.builder \
    .appName("proj") \
    .getOrCreate()
```

In [2]:

```
spark
```

Out[2]:

SparkSession - in-memory
SparkContext

Spark UI
Version
`v3.3.1`
Master
`local[*]`
AppName
`proj`

Creating the DataFrames

In [25]:

```
data_1 = [{"Category": 'A', "ID": 1, "Value": 121.44, "License": True},
        {"Category": 'B', "ID": 2, "Value": 300.01, "License": False},
        {"Category": 'C', "ID": 3, "Value": 10.99, "License": None},
        {"Category": 'E', "ID": 4, "Value": 33.87, "License": True}
        ]
```

In [26]:

```
df_1 = spark.createDataFrame(data_1)
```

In [27]:

```
data_2 = [{"Category": 'A', "ID": 5, "Value": 222.44, "Age": 37},
        {"Category": 'B', "ID": 6, "Value": 500.01, "Age": 55},
        {"Category": 'C', "ID": 7, "Value": 40.99, "Age": 22},
        {"Category": 'E', "ID": 9, "Value": 30.87, "Age": 20}
        ]
```

In [28]:

```
df_2 = spark.createDataFrame(data_2)
```

In [29]:

```
list(df_2.schema)
```

Out[29]:

```
[StructField('Age', LongType(), True),
 StructField('Category', StringType(), True),
 StructField('ID', LongType(), True),
 StructField('Value', DoubleType(), True)]
```

Labelling Each DataFrame

In [33]:

```
from pyspark.sql import functions as F
```

In [34]:

```
df_1 = df_1.withColumn('Data', F.lit('Data_1'))
df_2 = df_2.withColumn('Data', F.lit('Data_2'))
```

Merge using unionByName

In [35]:

```
merged_df = df_1.unionByName(df_2, allowMissingColumns=True)
```

In [59]:

```
merged_df.show()
```

```
+--------+---+-------+------+------+----+
|Category| ID|License| Value|  Data| Age|
+--------+---+-------+------+------+----+
|       A|  1|   true|121.44|Data_1|null|
|       B|  2|  false|300.01|Data_1|null|
|       C|  3|   null| 10.99|Data_1|null|
|       E|  4|   true| 33.87|Data_1|null|
|       A|  5|   null|222.44|Data_2|  37|
|       B|  6|   null|500.01|Data_2|  55|
|       C|  7|   null| 40.99|Data_2|  22|
|       E|  9|   null| 30.87|Data_2|  20|
+--------+---+-------+------+------+----+
```

Doing it the old way since allowMissingColumns was only added in spark 3.1

In [67]:

```
for column in [column for column in df_2.columns if column not in df_1.columns]:
    df_1 = df_1.withColumn(column, F.lit(None))
for column in [column for column in df_1.columns if column not in df_2.columns]:
    df_2 = df_2.withColumn(column, F.lit(None))
```

In [65]:

```
merged_df = df_1.unionByName(df_2)
```

In [66]:

```
merged_df.show()
```

```
+--------+---+-------+------+------+----+
|Category| ID|License| Value|  Data| Age|
+--------+---+-------+------+------+----+
|       A|  1|   true|121.44|Data_1|null|
|       B|  2|  false|300.01|Data_1|null|
|       C|  3|   null| 10.99|Data_1|null|
|       E|  4|   true| 33.87|Data_1|null|
|       A|  5|   null|222.44|Data_2|  37|
|       B|  6|   null|500.01|Data_2|  55|
|       C|  7|   null| 40.99|Data_2|  22|
|       E|  9|   null| 30.87|Data_2|  20|
+--------+---+-------+------+------+----+
```