In [1]:

```
from pyspark.sql import SparkSession
import pyspark.sql.functions as F
from pyspark.sql.types import *

spark = SparkSession\
    .builder\
    .appName("chapter-08-join")\
    .getOrCreate()
```

In [2]:

```
person = spark.createDataFrame([
    (0, "Bill Chambers", 0, [100]),
    (1, "Matei Zaharia", 1, [500, 250, 100]),
    (2, "Michael Armbrust", 1, [250, 100])])\
  .toDF("id", "name", "graduate_program", "spark_status")

graduateProgram = spark.createDataFrame([
    (0, "Masters", "School of Information", "UC Berkeley"),
    (2, "Masters", "EECS", "UC Berkeley"),
    (1, "Ph.D.", "EECS", "UC Berkeley")])\
  .toDF("id", "degree", "department", "school")

sparkStatus = spark.createDataFrame([
    (500, "Vice President"),
    (250, "PMC Member"),
    (100, "Contributor")])\
  .toDF("id", "status")
```

In [4]:

```
person.show()
```

```
+---+----------------+----------------+---------------+
| id|            name|graduate_program|   spark_status|
+---+----------------+----------------+---------------+
|  0|   Bill Chambers|               0|          [100]|
|  1|   Matei Zaharia|               1|[500, 250, 100]|
|  2|Michael Armbrust|               1|     [250, 100]|
+---+----------------+----------------+---------------+
```

In [5]:

```
graduateProgram.show()
```

```
+---+-------+--------------------+-----------+
| id| degree|          department|     school|
+---+-------+--------------------+-----------+
|  0|Masters|School of Informa...|UC Berkeley|
|  2|Masters|                EECS|UC Berkeley|
|  1|  Ph.D.|                EECS|UC Berkeley|
+---+-------+--------------------+-----------+
```

In [6]:

```
sparkStatus.show()
```

```
+---+--------------+
| id|        status|
+---+--------------+
|500|Vice President|
|250|    PMC Member|
|100|   Contributor|
+---+--------------+
```

In [7]:

```
# COMMAND ----------

joinExpression = person["graduate_program"] == graduateProgram['id']
```

In [8]:

```
# COMMAND ----------

# wrongJoinExpression = person["name"] == graduateProgram["school"]
```

In [9]:

```
# COMMAND ----------

joinType = "inner"
```

In [10]:

```
# COMMAND ----------

gradProgram2 = graduateProgram.union(spark.createDataFrame([
    (0, "Masters", "Duplicated Row", "Duplicated School")]))
```

In [11]:

```
gradProgram2.createOrReplaceTempView("gradProgram2")
```

In [12]:

```
# COMMAND ----------

from pyspark.sql.functions import expr

person.withColumnRenamed("id", "personId")\
  .join(sparkStatus, expr("array_contains(spark_status, id)")).show()


# COMMAND ----------
```

```
+--------+----------------+----------------+---------------+---+------
--------+
|personId|            name|graduate_program|   spark_status| id|
status|
+--------+----------------+----------------+---------------+---+------
--------+
|       0|   Bill Chambers|               0|          [100]|100|   Con
tributor|
|       1|   Matei Zaharia|               1|[500, 250, 100]|500|Vice P
resident|
|       1|   Matei Zaharia|               1|[500, 250, 100]|250|    PM
C Member|
|       1|   Matei Zaharia|               1|[500, 250, 100]|100|   Con
tributor|
|       2|Michael Armbrust|               1|     [250, 100]|250|    PM
C Member|
|       2|Michael Armbrust|               1|     [250, 100]|100|   Con
tributor|
+--------+----------------+----------------+---------------+---+------
--------+
```

In [ ]: