```
# Databricks notebook source
import pyspark
from pyspark.sql import SparkSession
from pyspark.sql.types import StructType,StructField, StringType,
IntegerType, ArrayType
from pyspark.sql.functions import col,array_contains

spark =
SparkSession.builder.appName('SparkByExamples.com').getOrCreate()

arrayStructureData = [
        (("James","","Smith"),["Java","Scala","C++"],"OH","M"),
        (("Anna","Rose",""),["Spark","Java","C++"],"NY","F"),
        (("Julia","","Williams"),["CSharp","VB"],"OH","F"),
        (("Maria","Anne","Jones"),["CSharp","VB"],"NY","M"),
        (("Jen","Mary","Brown"),["CSharp","VB"],"NY","M"),
        (("Mike","Mary","Williams"),["Python","VB"],"OH","M")
        ]

arrayStructureSchema = StructType([
        StructField('name', StructType([
             StructField('firstname', StringType(), True),
             StructField('middlename', StringType(), True),
             StructField('lastname', StringType(), True)
             ])),
         StructField('languages', ArrayType(StringType()), True),
         StructField('state', StringType(), True),
         StructField('gender', StringType(), True)
         ])


df = spark.createDataFrame(data = arrayStructureData, schema =
arrayStructureSchema)
df.printSchema()
df.show(truncate=False)

#Equals
df.filter(df.state == "OH") \
    .show(truncate=False)

#Not equals
df.filter(~(df.state == "OH")) \
    .show(truncate=False)
df.filter(df.state != "OH") \
    .show(truncate=False)

df.filter(col("state") == "OH") \
    .show(truncate=False)

df.filter("gender  == 'M'") \
    .show(truncate=False)

df.filter("gender  <> 'M'") \
    .show(truncate=False)

#IS IN
li=["OH","CA","DE"]
df.filter(df.state.isin(li)).show()
#IS NOT IN
df.filter(~df.state.isin(li)).show()

df.filter( (df.state  == "OH") & (df.gender  == "M") ) \
    .show(truncate=False)

df.filter(array_contains(df.languages,"Java")) \
    .show(truncate=False)

df.filter(df.name.lastname == "Williams") \
    .show(truncate=False)

df.filter(df.state.startswith("N")).show()
df.filter(df.state.endswith("H")).show()
df.filter(df.state.like("N%")).show()


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