

# Java-Success.com

Prepare to fast-track, choose & go places with 800+ Java & Big Data Q&As with lots of code & diagrams.

[Home](#) [Why? ▾](#) [300+ Java FAQs ▾](#) [300+ Big Data FAQs ▾](#) [Courses ▾](#)[👤 Membership ▾](#) [Your Career ▾](#)[Home](#) > [bigdata-success.com](#) > [Tutorials - Big Data](#) > [TUT - Spark Scala on Zeppelin](#) >

04: Spark on Zeppelin – DataFrame joins in Scala

## 04: Spark on Zeppelin – DataFrame joins in Scala

 Posted on [July 27, 2018](#)

This tutorial extends the series: [Spark on Apache Zeppelin Tutorials](#).

### 1. Create “Orders” DataFrame

```
1 import org.apache.spark.sql._
2 import org.apache.spark.sql.types._
3
4 val orders = Seq (
5     Row ("order-1", "cust-234", 234.50),
6     Row ("order-2", "cust-234", 347.00),
7     Row ("order-3", "cust-337", 67.00),
8     Row ("order-4", "cust-425", 84.50),
9     Row ("order-5", "cust-456", 134.50)
```

### 300+ Java Interview FAQs

300+ Java FAQs



16+ Java Key Areas Q&amp;As



150+ Java Architect FAQs



80+ Java Code Quality Q&amp;As



150+ Java Coding Q&amp;As



### 300+ Big Data Interview FAQs

300+ Big Data FAQs



Tutorials - Big Data

TUT -  Starting Big Data

TUT - Starting Spark &amp; Scala

```

10 )
11
12 val orderSchema = List(
13   StructField("order_num", StringType, true),
14   StructField("customer_num", StringType, true),
15   StructField("amount", DoubleType, true)
16 )
17
18 val ordersDf = spark.createDataFrame(sc.parallelize(
19   ordersDf.show()
20 )
21

```

```

1
2 +-----+-----+-----+
3 |order_num|customer_num|amount|
4 +-----+-----+-----+
5 |  order-1|    cust-234| 234.5|
6 |  order-2|    cust-234| 347.0|
7 |  order-3|    cust-337|  67.0|
8 |  order-4|    cust-425|  84.5|
9 |  order-5|    cust-456| 134.5|
10 +-----+-----+-----+
11

```

## 2. Create "Customers" DataFrame

```

1 import org.apache.spark.sql._
2 import org.apache.spark.sql.types._
3
4 val customers = Seq (
5   Row ("cust-234", "John"),
6   Row ("cust-239", "Peter"),
7   Row ("cust-337", "Samuel"),
8   Row ("cust-425", "Jordan"),
9   Row ("cust-456", "George")
10 )
11
12 val customerSchema = List(
13   StructField("customer_num", StringType, true),
14   StructField("customer_name", StringType, true)
15 )
16
17 val customersDf = spark.createDataFrame(sc.parallelize(
18   customersDf.show()
19 )
20

```

```

1
2 +-----+-----+
3 |customer_num|customer_name|

```

TUT - Starting with Python

TUT - Kafka

TUT - Pig

TUT - Apache Storm

TUT - Spark Scala on Zeppelin

TUT - Cloudera

TUT - Cloudera on Docker

TUT - File Formats

TUT - Spark on Docker

TUT - Flume

TUT - Hadoop (HDFS)

TUT - HBase (NoSQL)

TUT - Hive (SQL)

TUT - Hadoop & Spark

TUT - MapReduce

TUT - Spark and Scala

TUT - Spark & Java

TUT - PySpark on Databricks

TUT - Zookeeper

## 800+ Java Interview Q&As

300+ Core Java Q&As



300+ Enterprise Java Q&As



150+ Java Frameworks Q&As



120+ Companion Tech Q&As



Tutorials - Enterprise Java



```

4 | +-----+-----+
5 | | cust-234 | John |
6 | | cust-239 | Peter |
7 | | cust-337 | Samuel |
8 | | cust-425 | Jordan |
9 | | cust-456 | George |
10 | +-----+-----+
11 |

```

You can perform a number of joins between DataFrames. Default is the inner join. Joins can be of: inner, cross, outer, full, full\_outer, left, left\_outer, right, right\_outer, left\_semi, left\_anti.

### 3. inner join of two DataFrames

Customers and their orders.

```

1 | val dfJoined = ordersDf.join(customersDf, ordersDf
2 |                           .select(customersDf.col("cu
3 | dfJoined.show()
4 |

```

```

1 |
2 | +-----+-----+-----+-----+
3 | |customer_num|customer_name|order_num|amount|
4 | +-----+-----+-----+-----+
5 | | cust-337 | Samuel | order-3 | 67.0 |
6 | | cust-234 | John | order-1 | 234.5 |
7 | | cust-234 | John | order-2 | 347.0 |
8 | | cust-456 | George | order-5 | 134.5 |
9 | | cust-425 | Jordan | order-4 | 84.5 |
10 | +-----+-----+-----+-----+
11 |

```

### 4. leftanti join

Customers who do not have any orders.

```

1 | val dfJoined = customersDf.join(ordersDf, ordersDf
2 | dfJoined.show()
3 |

```

```

1
2 +-----+-----+
3 |customer_num|customer_name|
4 +-----+-----+
5 |    cust-239|      Peter|
6 +-----+-----+
7

```

## 5. left join

```

1 val dfJoined = customersDf.join(ordersDf, ordersDf
2
3 dfJoined.show()

```

```

1
2 +-----+-----+-----+-----+
3 |customer_num|customer_name|order_num|customer_num|
4 +-----+-----+-----+-----+
5 |    cust-337|      Samuel|   order-3|    cust-337|
6 |    cust-234|       John|   order-1|    cust-234|
7 |    cust-234|       John|   order-2|    cust-234|
8 |    cust-239|      Peter|      null|      null|
9 |    cust-456|     George|   order-5|    cust-456|
10 |    cust-425|     Jordan|   order-4|    cust-425|
11 +-----+-----+-----+-----+
12

```

## 5. leftsemi join

Customers who have orders.

```

1 val dfJoined = customersDf.join(ordersDf, ordersDf
2
3 dfJoined.show()

```

```

1
2 +-----+-----+
3 |customer_num|customer_name|
4 +-----+-----+
5 |    cust-337|      Samuel|
6 |    cust-234|       John|
7 |    cust-456|     George|
8 |    cust-425|     Jordan|
9 +-----+-----+
10

```

[◀ 03: Spark on Zeppelin – DataFrame Operations in Scala](#)[05: Spark on Zeppelin – semi-structured log file ▶](#)

## Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech Pty Ltd. The EmpoweringTech Pty Ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech Pty Ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. [Privacy Policy](#)

© 2022 [java-success.com](https://www.java-success.com)