Home (/) About (https://sparkbyexamples.com/about-sparkbyexamples/) | { One stop for all Spark Examples }

Spark by {Examples} (https://sparkbyexamples.com/)

Spark ⌄ (https://sparkbyexamples.com/)

PySpark (https://sparkbyexamples.com/pyspark-tutorial/)

Hive (https://sparkbyexamples.com/apache-hive-tutorial/)

HBase (https://sparkbyexamples.com/apache-hbase-tutorial/)

Kafka (https://sparkbyexamples.com/apache-kafka-tutorials-with-examples/)

FAQ's (https://sparkbyexamples.com/spark-questions/)

More ⌄ (https://sparkbyexamples.com/)

Spark Groupby Example with DataFrame

NNK (https://sparkbyexamples.com/author/admin/) - Apache Spark (https://sparkbyexamples.com/category/spark/)

Similar to SQL "GROUP BY" clause, Spark groupBy() function is used to collect the identical data into groups on DataFrame/Dataset and perform aggregate functions on the grouped data. In this article, I will explain several groupBy() examples with the Scala language.

The same approach can be used with the Pyspark (Spark with Python).

Syntax:

Spark Tutorial

- Spark – Installation on Windows (https://sparkbyexamples.com/spark/apache-spark-installation-on-windows/)
- Spark – Installation on Linux | Ubuntu (https://sparkbyexamples.com/spark/spark-installation-on-linux-ubuntu/)
- Spark – Cluster Setup with Hadoop Yarn (https://sparkbyexamples.com/spark/spark-setup-on-hadoop-yarn/)
- Spark – Web/Application UI (https://sparkbyexamples.com/spark/spark-web-ui-understanding/)
- Spark – Setup with Scala and IntelliJ (https://sparkbyexamples.com/spark/spark-setup-run-with-scala-intellij/)
- Spark – How to Run Examples From this Site on IntelliJ IDEA (https://sparkbyexamples.com/spark/how-to-run-spark-examples-from-intellij/)
- Spark – SparkSession (https://sparkbyexamples.com/spark/sparksession-explained-with-examples/)
- Spark – SparkContext (https://sparkbyexamples.com/spark/spark-sparkcontext/)

Spark RDD Tutorial

- Spark RDD – Parallelize (https://sparkbyexamples.com/apache-spark-rdd/how-to-create-an-rdd-using-parallelize/)
- Spark RDD – Read text file (https://sparkbyexamples.com/apache-spark-rdd/spark-read-multiple-text-files-into-a-single-rdd/)

[Spark RDD – Read CSV (https://sparkbyexamples.com/apache-spark-rdd/spark-load-csv-file-into-rdd/)](https://sparkbyexamples.com/apache-spark-rdd/spark-load-csv-file-into-rdd/)

[Spark RDD – Create RDD (https://sparkbyexamples.com/apache-spark-rdd/different-ways-to-create-spark-rdd/)](https://sparkbyexamples.com/apache-spark-rdd/different-ways-to-create-spark-rdd/)

[Spark RDD – Create Empty RDD (https://sparkbyexamples.com/apache-spark-rdd/spark-how-to-create-an-empty-rdd/)](https://sparkbyexamples.com/apache-spark-rdd/spark-how-to-create-an-empty-rdd/)

[Spark RDD – Transformations (https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-transformations/)](https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-transformations/)

[Spark RDD – Actions (https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-actions/)](https://sparkbyexamples.com/apache-spark-rdd/spark-rdd-actions/)

[Spark RDD – Pair Functions (https://sparkbyexamples.com/apache-spark-rdd/spark-pair-rdd-functions/)](https://sparkbyexamples.com/apache-spark-rdd/spark-pair-rdd-functions/)

[Spark RDD – Repartition and Coalesce (https://sparkbyexamples.com/spark/spark-repartition-vs-coalesce/)](https://sparkbyexamples.com/spark/spark-repartition-vs-coalesce/)

[Spark RDD – Shuffle Partitions (https://sparkbyexamples.com/spark/spark-shuffle-partitions/)](https://sparkbyexamples.com/spark/spark-shuffle-partitions/)

[Spark RDD – Cache vs Persist (https://sparkbyexamples.com/spark/spark-difference-between-cache-and-persist/)](https://sparkbyexamples.com/spark/spark-difference-between-cache-and-persist/)

[Spark RDD – Persistance Storage Levels (https://sparkbyexamples.com/spark/spark-persistence-storage-levels/)](https://sparkbyexamples.com/spark/spark-persistence-storage-levels/)

[Spark RDD – Broadcast Variables (https://sparkbyexamples.com/spark/spark-broadcast-variables/)](https://sparkbyexamples.com/spark/spark-broadcast-variables/)

[Spark RDD – Accumulator Variables (https://sparkbyexamples.com/spark/spark-accumulators/)](https://sparkbyexamples.com/spark/spark-accumulators/)

[Spark RDD – Convert RDD to DataFrame (https://sparkbyexamples.com/](https://sparkbyexamples.com/)

```
groupBy(col1 : scala.Predef.Str
      org.apache.spark.sql.Rela
```

When we perform `groupBy()` on Spark Dataframe, it returns `RelationalGroupedDataset` object which contains below aggregate functions.

`count()` - Returns the count of rows for each group.

`mean()` - Returns the mean of values for each group.

`max()` - Returns the maximum of values for each group.

`min()` - Returns the minimum of values for each group.

`sum()` - Returns the total for values for each group.

`avg()` - Returns the average for values for each group.

`agg()` - Using [agg()](agg()) function, we can calculate more than one aggregate at a time.

apache-spark-rdd/convert-spark-rdd-to-dataframe-dataset/)

Spark SQL Tutorial

Spark SQL – Create DataFrame (https://sparkbyexamples.com/spark/different-ways-to-create-a-spark-dataframe/)

Spark SQL – Select Columns (https://sparkbyexamples.com/spark/spark-select-columns-from-dataframe/)

Spark SQL – Add and Update Column (withColumn) (https://sparkbyexamples.com/spark/spark-dataframe-withcolumn/)

Spark SQL – Rename Nested Column (https://sparkbyexamples.com/spark/rename-a-column-on-spark-dataframes/)

Spark SQL – Drop column (https://sparkbyexamples.com/spark/spark-drop-column-from-dataframe-dataset/)

Spark SQL – Where | Filter (https://sparkbyexamples.com/spark/spark-dataframe-where-filter/)

Spark SQL – When Otherwise (https://sparkbyexamples.com/spark/spark-case-when-otherwise-example/)

Spark SQL – Collect data to Driver (https://sparkbyexamples.com/spark/spark-dataframe-collect/)

Spark SQL – Distinct (https://sparkbyexamples.com/spark/spark-remove-duplicate-rows/)

Spark SQL- Pivot Table DataFrame (https://sparkbyexamples.com/spark/how-to-pivot-table-and-unpivot-a-spark-dataframe/)

Spark SQL – Data Types (https://sparkbyexamples.com/spark/spark-sql-dataframe-data-types/)

pivot() - This function is used to Pivot the DataFrame which I will not be covered in this article as I already have a dedicated article for Pivot & Unvot DataFrame (https://sparkbyexamples.com/spark/how-to-pivot-table-and-unpivot-a-spark-dataframe/).

Preparing Data & DataFrame

Before we start, let's create the DataFrame (https://sparkbyexamples.com/spark/different-ways-to-create-a-spark-dataframe/) from a sequence of the data to work with. This DataFrame contains columns "employee_name", "department", "state", "salary", "age" and "bonus" columns.

We will use this Spark DataFrame to run groupBy() on "department" columns and calculate aggregates like minimum, maximum, average, total salary for each group using min(), max() and sum() aggregate functions respectively. and finally, we will also see how to do group and aggregate on multiple columns.

```
import spark.implicits._
val simpleData = Seq(("James","
    ("Michael","Sales","NY",860
    ("Robert","Sales","CA",8100
    ("Maria","Finance","CA",900
    ("Raman","Finance","CA",990
    ("Scott","Finance","NY",830
    ("Jen","Finance","NY",79000
    ("Jeff","Marketing","CA",80
    ("Kumar","Marketing","NY",9
  )
val df = simpleData.toDF("emplo
df.show()
```

Yields below output.

- Spark SQL – StructType | StructField (https://sparkbyexamples.com/spark/spark-sql-structtype-on-dataframe/)
- Spark SQL – Schema (https://sparkbyexamples.com/spark/spark-schema-explained-with-examples/)
- Spark SQL – Groupby (https://sparkbyexamples.com/spark/using-groupby-on-dataframe/)
- Spark SQL – Sort DataFrame (https://sparkbyexamples.com/spark/spark-how-to-sort-dataframe-column-explained/)
- Spark SQL – Join Types (https://sparkbyexamples.com/spark/spark-sql-dataframe-join/)
- Spark SQL – Union and UnionAll (https://sparkbyexamples.com/spark/spark-dataframe-union-and-union-all/)
- Spark SQL – map() vs mapPartitions() (https://sparkbyexamples.com/spark/spark-map-vs-mappartitions-transformation/)
- Spark SQL – foreach() vs foreachPartition() (https://sparkbyexamples.com/spark/spark-foreachpartition-vs-foreach-explained/)
- Spark SQL – map() vs flatMap() (https://sparkbyexamples.com/spark/spark-map-vs-flatmap-with-examples/)
- Spark SQL – Persist and Cache (https://sparkbyexamples.com/spark/spark-dataframe-cache-and-persist-explained/)
- Spark SQL – UDF (User Defined Functions) (https://sparkbyexamples.com/spark/spark-sql-udf/)
- Spark SQL – Array (ArrayType) Column (https://sparkbyexamples.com/spark/spark-array-arraytype-dataframe-column/)

```
+-------------+----------+-----
|employee_name|department|state
+-------------+----------+-----
|        James|     Sales|   NY
|      Michael|     Sales|   NY
|       Robert|     Sales|   CA
|        Maria|   Finance|   CA
|        Raman|   Finance|   CA
|        Scott|   Finance|   NY
|          Jen|   Finance|   NY
|         Jeff| Marketing|   CA
|        Kumar| Marketing|   NY
+-------------+----------+-----
```

groupBy and aggregate on DataFrame columns

Let's do the `groupBy()` on `department` column of DataFrame and then find the sum of salary for each department using `sum()` aggregate function.

```
df.groupBy("department").sum("s
+----------+-----------+
|department|sum(salary)|
+----------+-----------+
|Sales     |257000     |
|Finance   |351000     |
|Marketing |171000     |
+----------+-----------+
```

Similarly, we can calculate the number of employee in each department using `count()`

```
df.groupBy("department").count(
```

Calculate the minimum salary of each department using `min()`

```
df.groupBy("department").min("s
```

[Spark SQL – Map (MapType) column (https://sparkbyexamples.com/spark/spark-dataframe-map-maptype-column/)](https://sparkbyexamples.com/spark/spark-dataframe-map-maptype-column/)

[Spark SQL – Flatten Nested Struct Column (https://sparkbyexamples.com/spark/spark-flatten-nested-struct-column/)](https://sparkbyexamples.com/spark/spark-flatten-nested-struct-column/)

[Spark SQL – Flatten Nested Array Column (https://sparkbyexamples.com/spark/spark-flatten-nested-array-column-to-single-column/)](https://sparkbyexamples.com/spark/spark-flatten-nested-array-column-to-single-column/)

[Spark SQL – Explode Array & Map Columns (https://sparkbyexamples.com/spark/explode-spark-array-and-map-dataframe-column/)](https://sparkbyexamples.com/spark/explode-spark-array-and-map-dataframe-column/)

[Spark SQL – Sampling (https://sparkbyexamples.com/spark/spark-sampling-with-examples/)](https://sparkbyexamples.com/spark/spark-sampling-with-examples/)

[Spark SQL – Partitioning (https://sparkbyexamples.com/spark/spark-partitioning-understanding/)](https://sparkbyexamples.com/spark/spark-partitioning-understanding/)

Spark SQL Functions

[Spark SQL String Functions (https://sparkbyexamples.com/spark/usage-of-spark-sql-string-functions/)](https://sparkbyexamples.com/spark/usage-of-spark-sql-string-functions/)

[Spark SQL Date and Timestamp Functions (https://sparkbyexamples.com/spark/spark-sql-date-and-time-functions/)](https://sparkbyexamples.com/spark/spark-sql-date-and-time-functions/)

[Spark SQL Array Functions (https://sparkbyexamples.com/spark/spark-sql-array-functions/)](https://sparkbyexamples.com/spark/spark-sql-array-functions/)

[Spark SQL Map Functions (https://sparkbyexamples.com/spark/spark-sql-map-functions/)](https://sparkbyexamples.com/spark/spark-sql-map-functions/)

[Spark SQL Sort Functions (https://sparkbyexamples.com/spark/spark-sql-sort-functions/)](https://sparkbyexamples.com/spark/spark-sql-sort-functions/)

[Spark SQL Aggregate Functions (https://sparkbyexamples.com/](https://sparkbyexamples.com/)

Calculate the maximin salary of each department using `max()`

```
df.groupBy("department").max("s
```

Calculate the average salary of each department using `avg()`

```
df.groupBy("department").avg( "
```

Calculate the mean salary of each department using `mean()`

```
df.groupBy("department").mean(
```

groupBy and aggregate on multiple DataFrame columns

Similarly, we can also run groupBy and aggregate on two or more DataFrame columns, below example does group by on `department`,`state` and does sum() on `salary` and `bonus` columns.

```
//GroupBy on multiple columns
df.groupBy("department","state"
    .sum("salary","bonus")
    .show(false)
```

This yields the below output.

spark/spark-sql-aggregate-functions/)

Spark SQL Window Functions (https://sparkbyexamples.com/spark/spark-sql-window-functions/)

Spark SQL JSON Functions (https://sparkbyexamples.com/spark/spark-most-used-json-functions-with-examples/)

Spark Data Source API

Spark – Read & Write CSV file (https://sparkbyexamples.com/spark/spark-read-csv-file-into-dataframe/)

Spark – Read and Write JSON file (https://sparkbyexamples.com/spark/spark-read-and-write-json-file/)

Spark – Read & Write Parquet file (https://sparkbyexamples.com/spark/spark-read-write-dataframe-parquet-example/)

Spark – Read & Write XML file (https://sparkbyexamples.com/spark/spark-read-write-xml/)

Spark – Read & Write Avro files (https://sparkbyexamples.com/spark/read-write-avro-file-spark-dataframe/)

Spark – Read & Write Avro files (Spark version 2.3.x or earlier) (https://sparkbyexamples.com/spark/using-avro-data-files-from-spark-sql-2-3-x/)

Spark – Read & Write HBase using “hbase-spark” Connector (https://sparkbyexamples.com/spark/spark-read-write-using-hbase-spark-connector/)

Spark – Read & Write from HBase using Hortonworks (https://sparkbyexamples.com/spark/create-spark-dataframe-from-hbase-using-hortonworks/)

Spark – Read & Write ORC file (https://sparkbyexamples.com/spark/spark-read-orc-file-into-dataframe/)

```
+----------+-----+-----------+-
|department|state|sum(salary)|s
+----------+-----+-----------+-
|Finance   |NY   |162000     |3
|Marketing |NY   |91000      |2
|Sales     |CA   |81000      |2
|Marketing |CA   |80000      |1
|Finance   |CA   |189000     |4
|Sales     |NY   |176000     |3
+----------+-----+-----------+-
```

similarly, we can run group by and aggregate on tow or more columns for other aggregate functions, please refer below source code for example.

Running more aggregates at a time

Using `agg()` aggregate function we can calculate many aggregations at a time on a single statement using Spark SQL aggregate (https://sparkbyexamples.com/spark/spark-sql-aggregate-functions/) functions sum() (https://sparkbyexamples.com/spark/spark-sql-aggregate-functions/#sum), avg() (https://sparkbyexamples.com/spark/spark-sql-aggregate-functions/#avg), min() (https://sparkbyexamples.com/spark/spark-sql-aggregate-functions/#min), max() (https://sparkbyexamples.com/spark/spark-sql-aggregate-functions/#max) mean() (https://sparkbyexamples.com/spark/spark-sql-aggregate-functions/#mean) e.t.c. In order to use these, we should import `"import org.apache.spark.sql.functions._"`

[Spark – Read Binary File (https://sparkbyexamples.com/spark/spark-read-binary-file-into-dataframe/)](https://sparkbyexamples.com/spark/spark-read-binary-file-into-dataframe/)

Spark Streaming & Kafka

[Spark Streaming – OutputModes (https://sparkbyexamples.com/spark/spark-streaming-outputmode/)](https://sparkbyexamples.com/spark/spark-streaming-outputmode/)

[Spark Streaming – Reading Files From Directory (https://sparkbyexamples.com/spark/spark-streaming-read-json-files-from-directory/)](https://sparkbyexamples.com/spark/spark-streaming-read-json-files-from-directory/)

[Spark Streaming – Reading Data From TCP Socket (https://sparkbyexamples.com/spark/spark-streaming-from-tcp-socket/)](https://sparkbyexamples.com/spark/spark-streaming-from-tcp-socket/)

[Spark Streaming – Processing Kafka Messages in JSON Format (https://sparkbyexamples.com/spark/spark-streaming-with-kafka/)](https://sparkbyexamples.com/spark/spark-streaming-with-kafka/)

[Spark Streaming – Processing Kafka messages in AVRO Format (https://sparkbyexamples.com/spark/spark-streaming-consume-and-produce-kafka-messages-in-avro-format/)](https://sparkbyexamples.com/spark/spark-streaming-consume-and-produce-kafka-messages-in-avro-format/)

[Spark SQL Batch – Consume & Produce Kafka Message (https://sparkbyexamples.com/spark/spark-batch-processing-produce-consume-kafka-topic/)](https://sparkbyexamples.com/spark/spark-batch-processing-produce-consume-kafka-topic/)

```
import org.apache.spark.sql.fun
df.groupBy("department")
    .agg(
      sum("salary").as("sum_sal
      avg("salary").as("avg_sal
      sum("bonus").as("sum_bonu
      max("bonus").as("max_bonu
    .show(false)
```

This example does group on `department` column and calculates `sum()` and `avg()` of salary for each department and calculates `sum()` and `max()` of bonus for each department.

```
+----------+----------+--------
|department|sum_salary|avg_sala
+----------+----------+--------
|Sales     |257000    |85666.66
|Finance   |351000    |87750.0
|Marketing |171000    |85500.0
+----------+----------+--------
```

Using filter on aggregate data

Similar to SQL "HAVING" clause, On Spark DataFrame we can use either [where() (https://sparkbyexamples.com/spark/working-with-spark-dataframe-filter/)](https://sparkbyexamples.com/spark/working-with-spark-dataframe-filter/) or [filter() (https://sparkbyexamples.com/spark/working-with-spark-dataframe-filter/)](https://sparkbyexamples.com/spark/working-with-spark-dataframe-filter/) function to filter the rows of aggregated a.

```
f.groupBy("department")
    .agg(
      sum("salary").as("sum_sal
      avg("salary").as("avg_sal
      sum("bonus").as("sum_bonu
      max("bonus").as("max_bonu
    .where(col("sum_bonus") >=
    .show(false)
```

s removes the sum of a bonus that less than 50000 and yields below out.

```
----------+----------+--------
department|sum_salary|avg_sala
----------+----------+--------
Sales     |257000    |85666.66
Finance   |351000    |87750.0
----------+----------+--------
```

urce code

```
package com.sparkbyexamples.spa

import org.apache.spark.sql.Spa
import org.apache.spark.sql.fun

object GroupbyExample extends A
  val spark: SparkSession = Spa
    .master("local[1]")
    .appName("SparkByExamples.c
    .getOrCreate()

  spark.sparkContext.setLogLeve
  import spark.implicits._

  val simpleData = Seq(("James"
    ("Michael","Sales","NY",860
    ("Robert","Sales","CA",8100
    ("Maria","Finance","CA",900
    ("Raman","Finance","CA",990
    ("Scott","Finance","NY",830
    ("Jen","Finance","NY",79000
    ("Jeff","Marketing","CA",80
    ("Kumar","Marketing","NY",9
  )
  val df = simpleData.toDF("emp
  df.show()

  //Group By on single column
  df.groupBy("department").coun
  df.groupBy("department").avg(
  df.groupBy("department").sum(
  df.groupBy("department").min(
  df.groupBy("department").max(
  df.groupBy("department").mean

  //GroupBy on multiple columns
  df.groupBy("department","stat
    .sum("salary","bonus")
    .show(false)
  df.groupBy("department","stat
    .avg("salary","bonus")
    .show(false)
  df.groupBy("department","stat
    .max("salary","bonus")
    .show(false)
  df.groupBy("department","stat
    .min("salary","bonus")
    .show(false)
  df.groupBy("department","stat
    .mean("salary","bonus")
    .show(false)

  //Running Filter
  df.groupBy("department","stat
    .sum("salary","bonus")
```

```
      .show(false)

    //using agg function
    df.groupBy("department")
      .agg(
        sum("salary").as("sum_sal
        avg("salary").as("avg_sal
        sum("bonus").as("sum_bonu
        max("bonus").as("max_bonu
      .show(false)

    df.groupBy("department")
      .agg(
        sum("salary").as("sum_sal
        avg("salary").as("avg_sal
        sum("bonus").as("sum_bonu
        stddev("bonus").as("stdde
      .where(col("sum_bonus") > 5
      .show(false)
  }
```

This example is also available at GitHub (https://github.com/spark-examples/spark-scala-examples/blob/master/src/main/scala/com/sparkbyexamples/spark/dataframe/GroupbyExample.scala) project for reference.

Conclusion

In this tutorial, you have learned how to use `groupBy()` and aggregate functions on Spark DataFrame and also learned how to run these on multiple columns and finally filtering data on the aggregated column.

Thanks for reading. If you like it, please do share the article by following the below social links and any comments or suggestions are welcome in the comments sections!

Happy Learning !!

Share this:

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=facebook&nb=1)

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=reddit&nb=1)

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=pinterest&nb=1)

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=tumblr&nb=1)

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=pocket&nb=1)

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=linkedin&nb=1)

(https://sparkbyexamples.com/spark/using-groupby-on-dataframe/?share=twitter&nb=1)

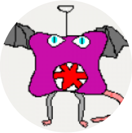
TAGS: AGG (HTTPS://SPARKBYEXAMPLES.COM/TAG/AGG/), GROUPBY (HTTPS://SPARKBYEXAMPLES.COM/TAG/GROUPBY/)

(https://sparkbyexamples.com/author/admin/)

NNK (Https://Sparkbyexamples.Com/Author/Admin/)

SparkByExamples.com is a Big Data and Spark examples community page, all examples are simple and easy to understand and well tested in our development environment Read more .. (https://sparkbyexamples.com/about-sparkbyexamples/)

› THIS POST HAS 3 COMMENTS



divkr 6 JAN 2021 REPLY

are you using python or scala for this tutorial ?

NNK 6 JAN 2021 REPLY

Examples on this page use Scala. If you are looking for GroupBy with Python (PySpark) see https://sparkbyexamples.com/pyspark/pyspark-groupby-explained-with-example/ (https://sparkbyexamples.com/pyspark/pyspark-groupby-explained-with-example/)

sri bhargavi

23 NOV 2020 REPLY

Hi, why we use agg without agg also can we perform agg functions rt??

Leave a Reply

Categories

Apache Hadoop (https://sparkbyexamples.com/category/hadoop/)

Apache Spark (https://sparkbyexamples.com/category/spark/)

Apache Spark Streaming (https://sparkbyexamples.com/category/spark/apache-spark-streaming/)

Apache Kafka (https://sparkbyexamples.com/category/kafka/)

Recent Posts

Spark regexp_replace() – Replace String Value (https://sparkbyexamples.com/spark/spark-regexp_replace-replace-string-value/)

How to Run a PySpark Script from Python? (https://sparkbyexamples.com/pyspark/run-pyspark-script-from-python-subprocess/)

Spark SQL like() Using Wildcard Example (https://sparkbyexamples.com/spark/spark-sql-like-using-wildcard-example/)

Apache HBase (https://sparkbyexamples.com/category/hbase/)

Apache Cassandra (https://sparkbyexamples.com/category/cassandra/)

Snowflake Database (https://sparkbyexamples.com/category/snowflake/)

H2O Sparkling Water (https://sparkbyexamples.com/category/h2o-sparkling-water/)

PySpark (https://sparkbyexamples.com/category/pyspark/)

Spark isin() & IS NOT IN Operator Example (https://sparkbyexamples.com/spark/spark-isin-is-not-in-operator-example/)

Spark – Get Size/Length of Array & Map Column (https://sparkbyexamples.com/spark/spark-get-size-length-of-array-map-column/)

Spark Using Length/Size Of a DataFrame Column (https://sparkbyexamples.com/spark/spark-using-length-size-of-a-dataframe-column/)

Spark rlike() Working with Regex Matching Examples (https://sparkbyexamples.com/spark/spark-rlike-regex-matching-examples/)

Spark Check String Column Has Numeric Values (https://sparkbyexamples.com/spark/spark-check-string-column-has-numeric-values/)

Spark Check Column Data Type is Integer or String (https://sparkbyexamples.com/spark/spark-check-column-data-type-is-integer-or-string/)

(https://twitter.com/sparkbyexamples)

(https://www.facebook.com/sparkbyexamples/)

(https://www.linkedin.com/in/n-nk-b860a8193/)

(https://github.com/spark-examples/)