

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

PySpark Tutorial

PySpark Tutorial For Beginners

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

PySpark – Features

(https://sparkbyexamples.com/pyspark-tutorial/#features)

PySpark – Advantages

(https://sparkbyexamples.com/pyspark-tutorial/#advantages)

PySpark – Modules & Packages

(https://sparkbyexamples.com/pyspark-tutorial/#modules-packages)

PySpark – Cluster Managers

(https://sparkbyexamples.com/pyspark-tutorial/#cluster-manager)

PySpark – Install on Windows

(https://sparkbyexamples.com/pyspark-tutorial/#pyspark-installation)

PySpark – Web/Application UI

(https://sparkbyexamples.com/spark/spark-web-ui-understanding/)

PySpark – SparkSession

(https://sparkbyexamples.com/pyspark/pyspark-what-is-sparksession/)

PySpark – RDD

(https://sparkbyexamples.com/pyspark-rdd)

PySpark – Parallelize

(https://sparkbyexamples.com/pyspark/pyspark-parallelize-create-rdd/)

PySpark – repartition() vs coalesce()

(https://sparkbyexamples.com/pyspark/pyspark-repartition-vs-coalesce/)

PySpark – Broadcast Variables

(https://sparkbyexamples.com/pyspark/pyspark-broadcast-variables/)

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

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

#1 Screen Recorder & Editor

Show Off Your Product, Teach A Course, Train Coworkers & More. Buy Camtasia® Today.

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

PySpark Window Functions

👤 NNK

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


📁 PySpark

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

🔍

PySpark Window functions are used to calculate results such as the rank, row number e.t.c over a range of input rows.

In this article, I've explained the concept of window functions, syntax, and finally how to use them with PySpark SQL and PySpark DataFrame API. These come in handy when we need to make aggregate operations in a specific window frame on DataFrame columns.



When possible try to leverage standard library as they are little bit more compile-time safety, handles null and perform better when compared to

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

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

More ✓ (https://sparkbyexamples.com/)

🔍

[PySpark – Accumulator](https://sparkbyexamples.com/pyspark/pyspark-accumulator-with-example/)
(<https://sparkbyexamples.com/pyspark/pyspark-accumulator-with-example/>).

PySpark DataFrame

[PySpark – Create a DataFrame](https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/)
(<https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/>).

[PySpark – Create an empty DataFrame](https://sparkbyexamples.com/pyspark/pyspark-create-an-empty-dataframe/)
(<https://sparkbyexamples.com/pyspark/pyspark-create-an-empty-dataframe/>).

[PySpark – Convert RDD to DataFrame](https://sparkbyexamples.com/pyspark/convert-pyspark-rdd-to-dataframe/)
(<https://sparkbyexamples.com/pyspark/convert-pyspark-rdd-to-dataframe/>).

[PySpark – Convert DataFrame to Pandas](https://sparkbyexamples.com/pyspark/convert-pyspark-dataframe-to-pandas/)
(<https://sparkbyexamples.com/pyspark/convert-pyspark-dataframe-to-pandas/>).

[PySpark – show\(\)](https://sparkbyexamples.com/pyspark/pyspark-show-display-dataframe-contents-in-table/)
(<https://sparkbyexamples.com/pyspark/pyspark-show-display-dataframe-contents-in-table/>).

[PySpark – StructType & StructField](https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/)
(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/>).

[PySpark – Row Class](https://sparkbyexamples.com/pyspark/pyspark-row-using-rdd-dataframe/)
(<https://sparkbyexamples.com/pyspark/pyspark-row-using-rdd-dataframe/>).

[PySpark – Column Class](https://sparkbyexamples.com/pyspark/pyspark-column-functions/)
(<https://sparkbyexamples.com/pyspark/pyspark-column-functions/>).

[PySpark – select\(\)](https://sparkbyexamples.com/pyspark/select-columns-from-pyspark-dataframe/)
(<https://sparkbyexamples.com/pyspark/select-columns-from-pyspark-dataframe/>).

[PySpark – collect\(\)](https://sparkbyexamples.com/pyspark/pyspark-collect/)
(<https://sparkbyexamples.com/pyspark/pyspark-collect/>).

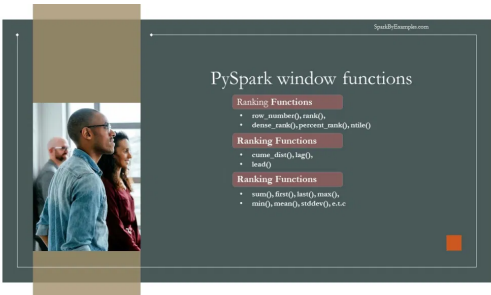
[PySpark – withColumn\(\)](https://sparkbyexamples.com/pyspark/pyspark-withcolumn/)
(<https://sparkbyexamples.com/pyspark/pyspark-withcolumn/>).

UDF's. If your application is critical on performance try to avoid using custom UDF at all costs as these are not guarantee on performance.

1. Window Functions

PySpark Window functions operate on a group of rows (like frame, partition) and return a single value for every input row. PySpark SQL supports three kinds of window functions:

- [ranking functions](#)
- [analytic functions](#)
- [aggregate functions](#)



PySpark Window Functions

The below table defines Ranking and Analytic functions and for aggregate functions, we can use any existing [aggregate functions](#) (<https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/>) as a window function.

To perform an operation on a group first, we need to partition the data using Window.partitionBy() , and for row number and rank function we need to additionally order by on partition data using orderBy clause.

LinkedIn ad
made easy

[PySpark – withColumnRenamed\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-rename-dataframe-column/>).

[PySpark – where\(\) & filter\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-where-filter/>).

[PySpark – drop\(\) & dropDuplicates\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-distinct-to-drop-duplicates/>).

[PySpark – orderBy\(\) and sort\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-orderby-and-sort-explained/>).

[PySpark – groupBy\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-groupby-explained-with-example/>).

[PySpark – join\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-join-explained-with-examples/>).

[PySpark – union\(\) & unionAll\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-union-and-unionall/>).

[PySpark – unionByName\(\).](#)
(<https://sparkbyexamples.com/spark/spark-merge-two-dataframes-with-different-columns/>).

[PySpark – UDF \(User Defined Function\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-udf-user-defined-function/>).

[PySpark – map\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-map-transformation/>).

[PySpark – flatMap\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-flatmap-transformation/>).

[pyspark – foreach\(\).](#)
(<https://sparkbyexamples.com/pyspark/pyspark-loop-iterate-through-rows-in-dataframe/#use-foreach-loop-through-dataframe>).

[PySpark – sample\(\) vs sampleBy\(\).](#)

Click on each link to know more about these functions along with the Scala examples.

WINDOW FUNCTIONS USAGE & SYNTAX	PYSPARK WINDOW FUNCTIONS DESCRIPTION
row_number(): Column	Returns a sequential number starting from 1 within a window partition
rank(): Column	Returns the rank of rows within a window partition, with gaps.
percent_rank(): Column	Returns the percentile rank of rows within a window partition.
dense_rank(): Column	Returns the rank of rows within a window partition without any gaps. Where as Rank() returns rank with gaps.
ntile(n: Int): Column	Returns the ntile id in a window partition
cume_dist(): Column	Returns the cumulative distribution of values within a window partition
lag(e: Column, offset: Int): Column lag(columnName: String, offset: Int): Column lag(columnName: String, offset: Int, defaultValue: Any): Column	returns the value that is `offset` rows before the current row, and `null` if there is less than `offset` rows before the current row.

[\(https://sparkbyexamples.com/pyspark/pyspark-sampling-example/\)](https://sparkbyexamples.com/pyspark/pyspark-sampling-example/)

[PySpark – fillna\(\) & fill\(\). \(https://sparkbyexamples.com/pyspark/pyspark-fillna-fill-replace-null-values/\)](https://sparkbyexamples.com/pyspark/pyspark-fillna-fill-replace-null-values/)

[PySpark – pivot\(\)_\(Row to Column\). \(https://sparkbyexamples.com/pyspark/pyspark-pivot-and-unpivot-dataframe/\)](https://sparkbyexamples.com/pyspark/pyspark-pivot-and-unpivot-dataframe/)

[PySpark – partitionBy\(\). \(https://sparkbyexamples.com/pyspark/pyspark-partitionby-example/\)](https://sparkbyexamples.com/pyspark/pyspark-partitionby-example/)

[PySpark – ArrayType Column \(Array\). \(https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/)

[PySpark – MapType \(Map/Dict\). \(https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/)

PySpark SQL Functions

[PySpark – Aggregate Functions \(https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/\)](https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/)

[PySpark – Window Functions \(https://sparkbyexamples.com/pyspark/pyspark-window-functions/\)](https://sparkbyexamples.com/pyspark/pyspark-window-functions/)

[PySpark – Date and Timestamp Functions \(https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/\)](https://sparkbyexamples.com/pyspark/pyspark-sql-date-and-timestamp-functions/)

[PySpark – JSON Functions \(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/\)](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/)

PySpark Datasources

[PySpark – Read & Write CSV File \(https://sparkbyexamples.com/pyspark/pyspark-read-csv-file-into-dataframe/\)](https://sparkbyexamples.com/pyspark/pyspark-read-csv-file-into-dataframe/)

[PySpark – Read & Write Parquet File \(https://sparkbyexamples.com/pyspark/pyspark-read-write-parquet-file/\)](https://sparkbyexamples.com/pyspark/pyspark-read-write-parquet-file/)

WINDOW FUNCTIONS USAGE & SYNTAX	PYSPARK WINDOW FUNCTIONS DESCRIPTION
lead(columnName: String, offset: Int): Column lead(columnName: String, offset: Int): Column lead(columnName: String, offset: Int, defaultValue: Any): Column	returns the value that is `offset` rows after the current row, and `null` if there is less than `offset` rows after the current row.

Before we start with an example, first let's [create a PySpark DataFrame \(https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/\)](https://sparkbyexamples.com/pyspark/different-ways-to-create-dataframe-in-pyspark/) to work with.

```
spark = SparkSession.builder.appName("PySpark SQL")\
    .master("local[*]")\
    .getOrCreate()

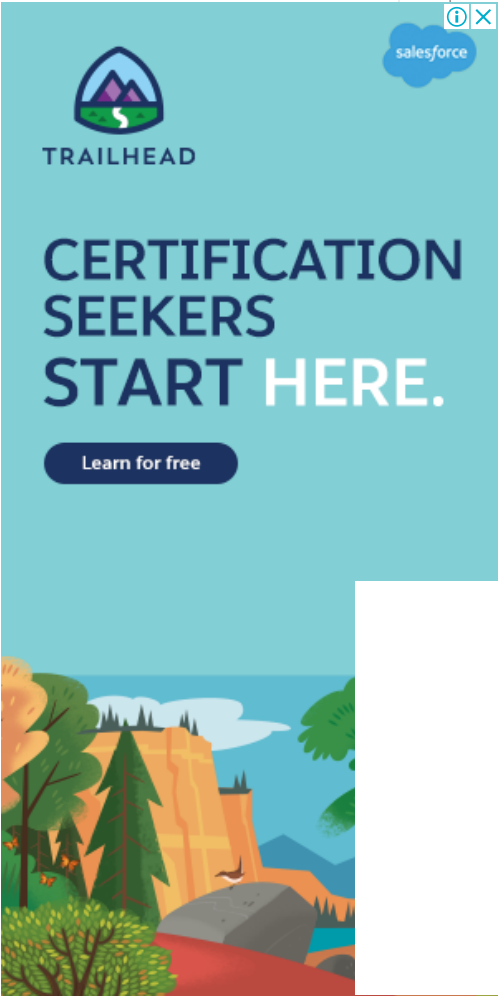
simpleData = (("James", "Sales", 4600),
              ("Michael", "Sales", 4600),
              ("Robert", "Sales", 4100),
              ("Maria", "Finance", 3000),
              ("James", "Sales", 3000),
              ("Scott", "Finance", 3300),
              ("Jen", "Finance", 3900),
              ("Jeff", "Marketing", 3000),
              ("Kumar", "Marketing", 2000),
              ("Saif", "Sales", 4100) \
              )

columns= ["employee_name", "dept", "salary"]
df = spark.createDataFrame(data=simpleData, schema=columns)
df.printSchema()
df.show(truncate=False)
```

Yields below output

[pyspark/pyspark-read-and-write-parquet-file/](#)

[PySpark – Read & Write JSON file](#)
(<https://sparkbyexamples.com/pyspark/pyspark-read-json-file-into-dataframe/>)



PySpark Built-In Functions

[PySpark – when\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-when-otherwise/>)

[PySpark – expr\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-sql-expr-expression-function/>)

```
root
|-- employee_name: string (nullable = true)
|-- department: string (nullable = true)
|-- salary: long (nullable = true)

+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|James        |Sales     |3000   |
|Michael      |Sales     |4600   |
|Robert       |Sales     |4100   |
|Maria        |Finance   |3000   |
|James        |Sales     |3000   |
|Scott        |Finance   |3300   |
|Jen          |Finance   |3900   |
|Jeff         |Marketing |3000   |
|Kumar        |Marketing |2000   |
|Saif         |Sales     |4100   |
+-----+-----+-----+
```

PySpark Window

Ranking functions

row_number Window

function

row_number() window function is used to give the sequential row number starting from 1 to the result of each window partition.

LinkedIn ads made easy

LinkedIn Marketing



[PySpark – lit\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/\)](https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/)

[PySpark – split\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-convert-string-to-array-column/\)](https://sparkbyexamples.com/pyspark/pyspark-convert-string-to-array-column/)

[PySpark – concat_ws\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-convert-array-column-to-string-column/\)](https://sparkbyexamples.com/pyspark/pyspark-convert-array-column-to-string-column/)

[Pyspark – substring\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-substring-from-a-column/\)](https://sparkbyexamples.com/pyspark/pyspark-substring-from-a-column/)

[PySpark – translate\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#translate-replace-character-by-character\)](https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#translate-replace-character-by-character)

[PySpark – regexp_replace\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#regexp_replace-replace-string-columns\)](https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#regexp_replace-replace-string-columns)

[PySpark – overlay\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#overlay-function\)](https://sparkbyexamples.com/pyspark/pyspark-replace-column-values/#overlay-function)

[PySpark – to_timestamp\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-to-timestamp-convert-string-to-timestamp-type/\)](https://sparkbyexamples.com/pyspark/pyspark-to-timestamp-convert-string-to-timestamp-type/)

[PySpark – to_date\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-to-date-convert-timestamp-to-date/\)](https://sparkbyexamples.com/pyspark/pyspark-to-date-convert-timestamp-to-date/)

[PySpark – date_format\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-date-format-convert-date-to-string-format/\)](https://sparkbyexamples.com/pyspark/pyspark-date-format-convert-date-to-string-format/)

[PySpark – datediff\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#datediff\)](https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#datediff)

[PySpark – months_between\(\).
\(https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#months_between\)](https://sparkbyexamples.com/pyspark/pyspark-difference-between-two-dates-days-months-years/#months_between)

```
from pyspark.sql.window import Window
from pyspark.sql.functions import row_number

windowSpec = Window.partitionBy('department')

df.withColumn("row_number", row_number().over(windowSpec))
.show(truncate=False)
```

Yields below output.

employee_name	department	salary
James	Sales	3000
James	Sales	3000
Robert	Sales	4100
Saif	Sales	4100
Michael	Sales	4600
Maria	Finance	3000
Scott	Finance	3300
Jen	Finance	3900
Kumar	Marketing	2000
Jeff	Marketing	3000

2.2 rank Window Function

rank() window function is used to provide a rank to the result within a window partition. This function leaves gaps in rank when there are ties.

```
"""rank"""
from pyspark.sql.functions import rank

df.withColumn("rank", rank().over(Window.partitionBy('department').orderBy('salary')))
.show()
```

Yields below output.

LinkedIn ads made easy

Launch your next LinkedIn campaign that works using these proven tips and tricks

LinkedIn Marketing

[Learn More](#)

[PySpark – explode\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-explode-nested-array-into-rows/>).

[PySpark – array_contains\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array_contains).

[PySpark – array\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-arraytype-column-with-examples/#array>).

[PySpark – collect_list\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-list>).

[PySpark – collect_set\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#collect-set>).

[PySpark – create_map\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-convert-dataframe-columns-to-maptype-dict/>).

[PySpark – map_keys\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_keys).

[PySpark – map_values\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-maptype-dict-examples/#map_values).

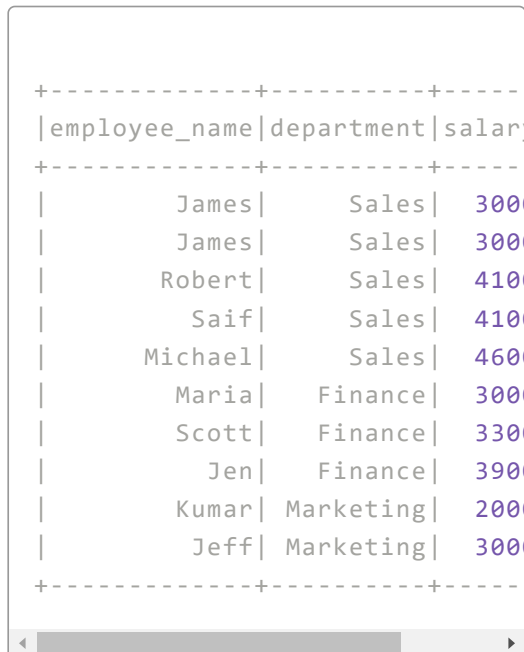
[PySpark – struct\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-structtype-and-structfield/#update-struct-function>).

[PySpark – countDistinct\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-count-distinct-from-dataframe/>).

[PySpark – sum\(\), avg\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-dataframe-groupby-and-sort-by-descending-order/>).

[PySpark – row_number\(\)](#)
(https://sparkbyexamples.com/pyspark/pyspark-window-functions/#row_number).

[PySpark – rank\(\)](#)
(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/#rank>).



employee_name	department	salary
James	Sales	3000
James	Sales	3000
Robert	Sales	4100
Saif	Sales	4100
Michael	Sales	4600
Maria	Finance	3000
Scott	Finance	3300
Jen	Finance	3900
Kumar	Marketing	2000
Jeff	Marketing	3000

This is the same as the RANK function in SQL.

2.3 dense_rank Window Function

`dense_rank()` window function is used to get the result with rank of rows within a window partition without any gaps. This is similar to `rank()` function difference being rank function leaves gaps in rank when there are ties.



```
"""dens_rank"""
from pyspark.sql.functions import dense_rank
df.withColumn("dense_rank", dense_rank().over(partitionBy=...))
.show()
```

Yields below output.

[PySpark – dense_rank\(\).](https://sparkbyexamples.com/pyspark/pyspark-window-functions/#dense_rank)
(https://sparkbyexamples.com/pyspark/pyspark-window-functions/#dense_rank).

[PySpark – percent_rank\(\).](https://sparkbyexamples.com/pyspark/pyspark-window-functions/#percent_rank)
(https://sparkbyexamples.com/pyspark/pyspark-window-functions/#percent_rank).

[PySpark – typedLit\(\).](https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/#typedlit)
(<https://sparkbyexamples.com/pyspark/pyspark-lit-add-literal-constant/#typedlit>).

[PySpark – from_json\(\).](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#from_json)
(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#from_json).

[PySpark – to_json\(\).](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#to_json)
(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#to_json).

[PySpark – json_tuple\(\).](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#json_tuple)
(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#json_tuple).

[PySpark – get_json_object\(\).](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#get_json_object)
(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#get_json_object).

[PySpark – schema_of_json\(\).](https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#schema_of_json)
(https://sparkbyexamples.com/pyspark/pyspark-json-functions-with-examples/#schema_of_json).

```
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|      James |     Sales|   3000|
|      James |     Sales|   3000|
|    Robert |     Sales|   4100|
|      Saif |     Sales|   4100|
|   Michael |     Sales|   4600|
|     Maria |  Finance|   3000|
|     Scott |  Finance|   3300|
|        Jen |  Finance|   3900|
|     Kumar |Marketing|   2000|
|        Jeff |Marketing|   3000|
+-----+-----+-----+
```

This is the same as the DENSE_RANK function in SQL.

2.4 percent_rank Window Function

```
""" percent_rank """
from pyspark.sql.functions import percent_rank
df.withColumn("percent_rank", percent_rank())
df.show()
```

Yields below output.

```
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|      James |     Sales|   3000|
|      James |     Sales|   3000|
|    Robert |     Sales|   4100|
|      Saif |     Sales|   4100|
|   Michael |     Sales|   4600|
|     Maria |  Finance|   3000|
|     Scott |  Finance|   3300|
|        Jen |  Finance|   3900|
|     Kumar |Marketing|   2000|
|        Jeff |Marketing|   3000|
+-----+-----+-----+
```

This is the same as the PERCENT_RANK function in SQL.

2.5 ntile Window Function

`ntile()` window function returns the relative rank of result rows within a window partition. In below example we have used 2 as an argument to `ntile` hence it returns ranking between 2 values (1 and 2)

```
"""ntile"""
from pyspark.sql.functions import ntile
df.withColumn("ntile", ntile(2)).show()
```

Yields below output.

```
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|James|Sales|3000|
|James|Sales|3000|
|Robert|Sales|4100|
|Saif|Sales|4100|
|Michael|Sales|4600|
|Maria|Finance|3000|
|Scott|Finance|3300|
|Jen|Finance|3900|
|Kumar|Marketing|2000|
|Jeff|Marketing|3000|
+-----+-----+-----+
```

This is the same as the `NTILE` function in SQL.

3. PySpark Window

Analytic functions

3.1 `cume_dist` Window

Function

`cume_dist()` window function is used to get the cumulative distribution of values within a window partition.

This is the same as the `DENSE_RANK` function in SQL.

```

""" cume_dist """
from pyspark.sql.functions import cume_dist
df.withColumn("cume_dist", cume_dist("salary"))
df.show()

```

```

+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|      James|      Sales|   3000|
|      James|      Sales|   3000|
|    Robert|      Sales|   4100|
|      Saif|      Sales|   4100|
| Michael|      Sales|   4600|
|      Maria|   Finance|   3000|
|      Scott|   Finance|   3300|
|       Jen|   Finance|   3900|
|      Kumar|Marketing|   2000|
|       Jeff|Marketing|   3000|
+-----+-----+-----+

```

3.2 lag Window Function

This is the same as the LAG function in SQL.

```

"""lag"""
from pyspark.sql.functions import lag
df.withColumn("lag", lag("salary", 1))
df.show()

```

```

+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|      James|      Sales|   3000|
|      James|      Sales|   3000|
|    Robert|      Sales|   4100|
|      Saif|      Sales|   4100|
| Michael|      Sales|   4600|
|      Maria|   Finance|   3000|
|      Scott|   Finance|   3300|
|       Jen|   Finance|   3900|
|      Kumar|Marketing|   2000|
|       Jeff|Marketing|   3000|
+-----+-----+-----+

```

3.3 lead Window Function

This is the same as the LEAD function in SQL.

```
"""lead"""
from pyspark.sql.functions import lead
df.withColumn("lead", lead("salary", 1))
df.show()
```

```
+-----+-----+-----+
|employee_name|department|salary|
+-----+-----+-----+
|      James|      Sales|   3000|
|      James|      Sales|   3000|
|    Robert|      Sales|   4100|
|      Saif|      Sales|   4100|
|   Michael|      Sales|   4600|
|      Maria|   Finance|   3000|
|      Scott|   Finance|   3300|
|        Jen|   Finance|   3900|
|      Kumar|Marketing|   2000|
|        Jeff|Marketing|   3000|
+-----+-----+-----+
```

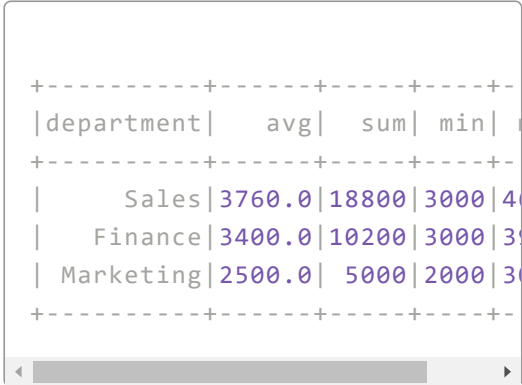
4. PySpark Window

Aggregate Functions

In this section, I will explain how to calculate sum, min, max for each department using PySpark SQL Aggregate window functions and WindowSpec. When working with Aggregate functions, we don't need to use order by clause.

```
windowSpecAgg = Window.partitionBy("department")
from pyspark.sql.functions import avg, sum, min, max, row_number
df.withColumn("row", row_number().over(windowSpecAgg))
df.withColumn("avg", avg("salary").over(windowSpecAgg))
df.withColumn("sum", sum("salary").over(windowSpecAgg))
df.withColumn("min", min("salary").over(windowSpecAgg))
df.withColumn("max", max("salary").over(windowSpecAgg))
df.where(col("row")==1).select("department", "avg", "sum", "min", "max")
df.show()
```

This yields below output



```
+-----+-----+-----+-----+
|department|  avg|  sum| min| max|
+-----+-----+-----+-----+
|      Sales|3760.0|18800|3000|4000|
|   Finance|3400.0|10200|3000|3500|
| Marketing|2500.0| 5000|2000|3000|
+-----+-----+-----+-----+
```

Please refer for more [Aggregate Functions](https://sparkbyexamples.com/spark/sql-aggregate-functions/)
(<https://sparkbyexamples.com/spark/sql-aggregate-functions/>)

Source Code of Window Functions Example

```

import pyspark
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("EmployeeRanking").getOrCreate()

simpleData = (("James", "Sales", 4600),
              ("Michael", "Sales", 4600),
              ("Robert", "Sales", 4100),
              ("Maria", "Finance", 3000),
              ("James", "Sales", 3000),
              ("Scott", "Finance", 3300),
              ("Jen", "Finance", 3900),
              ("Jeff", "Marketing", 3000),
              ("Kumar", "Marketing", 2000),
              ("Saif", "Sales", 4100) \
              )

columns= ["employee_name", "department", "salary"]

df = spark.createDataFrame(data=simpleData, schema=columns)

df.printSchema()
df.show(truncate=False)

from pyspark.sql.window import Window
from pyspark.sql.functions import row_number
windowSpec = Window.partitionBy("department").orderBy("salary")

df.withColumn("row_number", row_number().over(windowSpec)) \
    .show(truncate=False)

from pyspark.sql.functions import rank
df.withColumn("rank", rank().over(windowSpec)) \
    .show()

from pyspark.sql.functions import dense_rank
df.withColumn("dense_rank", dense_rank().over(windowSpec)) \
    .show()

from pyspark.sql.functions import percent_rank
df.withColumn("percent_rank", percent_rank().over(windowSpec)) \
    .show()

from pyspark.sql.functions import ntile
df.withColumn("ntile", ntile(2).over(windowSpec)) \
    .show()

from pyspark.sql.functions import cume_dist
df.withColumn("cume_dist", cume_dist().over(windowSpec)) \
    .show()

from pyspark.sql.functions import lag
df.withColumn("lag", lag("salary", 1).over(windowSpec)) \
    .show()

```

```

from pyspark.sql.functions import lead
df.withColumn("lead", lead("salary", 1))
    .show()

windowSpecAgg = Window.partitionBy()
from pyspark.sql.functions import row_number
df.withColumn("row", row_number())
    .withColumn("avg", avg(col("salary")))
    .withColumn("sum", sum(col("salary")))
    .withColumn("min", min(col("salary")))
    .withColumn("max", max(col("salary")))
    .where(col("row") == 1).select()
    .show()

```

The complete source code is available at [PySpark Examples GitHub](https://github.com/spark-examples/pyspark-examples/blob/master/pyspark-window-functions.py) (<https://github.com/spark-examples/pyspark-examples/blob/master/pyspark-window-functions.py>) for reference.

Conclusion

In this tutorial, you have learned what are PySpark SQL Window functions their syntax and how to use them with aggregate function along with several examples in Scala.

References

I would recommend reading [Window Functions Introduction](https://databricks.com/blog/2015/07/15/introducing-window-functions-in-spark-sql.html) (<https://databricks.com/blog/2015/07/15/introducing-window-functions-in-spark-sql.html>) and [SQL Window Functions API](https://github.com/apache/spark/blob/master/sql/core/src/main/scala/org/apache/spark/sql/functions.scala) (<https://github.com/apache/spark/blob/master/sql/core/src/main/scala/org/apache/spark/sql/functions.scala>) blogs for a further understanding of Windows functions. Also, refer to [SQL Window functions](http://www.sqlservertutorial.net/sql-server-window-functions/) (<http://www.sqlservertutorial.net/sql-server-window-functions/>) to know window functions from native SQL.

Happy Learning !!

Share this:



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=facebook&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=reddit&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=pinterest&nb=1>)

4



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=tumblr&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=pocket&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=linkedin&nb=1>)



(<https://sparkbyexamples.com/pyspark/pyspark-window-functions/?share=twitter&nb=1>)

TAGS: [AGGREGATE FUNCTIONS](#)

(<https://sparkbyexamples.com/tag/aggregate-functions/>), **[ANALYTIC FUNCTIONS](#)**

(<https://sparkbyexamples.com/tag/analytic-functions/>), **[OVER](#)**

(<https://sparkbyexamples.com/tag/over/>), **[RANK](#)**

(<https://sparkbyexamples.com/tag/rank/>), **[RANKING FUNCTIONS](#)**

(<https://sparkbyexamples.com/tag/ranking-functions/>), **[ROW](#)**

(<https://sparkbyexamples.com/tag/row/>)



[NNK](#)

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

(<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 9 COMMENTS**



Raymond

25 APR 2021 [REPLY](#)

Great post, keep it up



Anonymous

22 FEB 2021 [REPLY](#)

This is great, would appreciate, we add more examples for order by (rowsBetween and rangeBetween)



NNK 23 FEB 2021 [REPLY](#)

Sure. will do.



BobCG 10 DEC 2020 [REPLY](#)

Great job!!!

The same result for Window Aggregate Functions:

```
df.groupBy('dep').agg(
  avg('salary').alias('avg'),
  sum('salary').alias('sum'),
  min('salary').alias('min'),
  max('salary').alias('max')
).select('dep', 'avg',
'sum', 'min', 'max').show()
```



David 4 FEB 2021 [REPLY](#)

The difference would be that with the Window Functions you can append these new columns to the existing DataFrame. If you just group by department you would have the department plus the aggregate values but not the employee name or salary for each one.



Anonymous

8 DEC 2020 [REPLY](#)

Awesome explanations.





NNK 10 DEC 2020 [REPLY](#)

Thanks.



Ayan 19 OCT 2020 [REPLY](#)

Great job.Super easy to comprehend



NNK 19 OCT 2020 [REPLY](#)

Thanks for your comment and liking Pyspark window functions.



Anonymous

16 OCT 2020 [REPLY](#)

thank you very much

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/>)

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/>)

Recent Posts

Spark regex_replace() – Replace String Value
(https://sparkbyexamples.com/spark/spark-regex_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/>)

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/>)

About SparkByExamples.Com

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/>)

Follow Us



<https://www.> <https://www.>



<https://twitter.> ok.co n.com/ <https://github.>

r.com/ m/spar in/n- b.com/

sparkb kbyex nk- spark-

yexam ample b860a examp

[ples\)](https://ples) [s/\)](https://s/) [8193/\)](https://8193/) [les/\)](https://les/)

