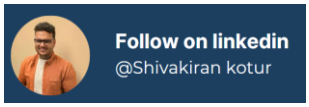
Master PySpark: From Zero to Big Data Hero!!

Windows Function in PySpark Part 3

```
from pyspark.sql import SparkSession
from pyspark.sql.window import Window
import pyspark.sql.functions as F

# Updated sample data with students, different subjects, marks, and
semesters
data = [
    ("Alice", "Math", 90, 1),
    ("Alice", "Science", 85, 1),
    ("Alice", "History", 78, 1),
    ("Bob", "Math", 80, 1),
    ("Bob", "Science", 81, 1),
    ("Bob", "History", 77, 1),
    ("Charlie", "Math", 75, 1),
    ("Charlie", "Science", 82, 1),
    ("Charlie", "History", 79, 1),
    ("Alice", "Physics", 86, 2),
    ("Alice", "Chemistry", 92, 2),
    ("Alice", "Biology", 80, 2),
    ("Bob", "Physics", 94, 2),
    ("Bob", "Chemistry", 91, 2),
    ("Bob", "Biology", 96, 2),
    ("Charlie", "Physics", 89, 2),
    ("Charlie", "Chemistry", 88, 2),
    ("Charlie", "Biology", 85, 2),
    ("Alice", "Computer Science", 95, 3),
    ("Alice", "Electronics", 91, 3),
    ("Alice", "Geography", 97, 3),
    ("Bob", "Computer Science", 88, 3),
    ("Bob", "Electronics", 66, 3),
    ("Bob", "Geography", 92, 3),
    ("Charlie", "Computer Science", 92, 3),
    ("Charlie", "Electronics", 97, 3),
    ("Charlie", "Geography", 99, 3)
]
```



Follow me on LinkedIn – Shivakiran kotur

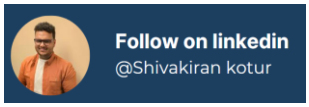
```
# Create a DataFrame
columns = ["First Name", "Subject", "Marks", "Semester"]
df = spark.createDataFrame(data, columns)

# 1. Which student scored max marks in each semester considering
all subjects
window_spec_max_marks =
Window.partitionBy("Semester").orderBy(F.desc("Marks"))
max_marks_df = df.withColumn("Rank",
F.rank().over(window_spec_max_marks))
top_scorer = max_marks_df.filter(max_marks_df["Rank"] == 1)
print("top_scorer:")
top_scorer.show()
```

```
top_scorer:
+----------+---------+-----+--------+----+
|First Name|  Subject|Marks|Semester|Rank|
+----------+---------+-----+--------+----+
|     Alice|     Math|   90|       1|   1|
|       Bob|  Biology|   96|       2|   1|
|   Charlie|Geography|   99|       3|   1|
+----------+---------+-----+--------+----+
```

```
# 2. Percentage of each student considering all subjects
window_spec_total_marks = Window.partitionBy("First Name",
"Semester")
df = df.withColumn("TotalMarks",
F.sum("Marks").over(window_spec_total_marks))
df = df.withColumn("Percentage", (F.col("TotalMarks") / (3 *
100)).cast("decimal(5, 2)")*100)
df2 = df.groupBy("First Name",
"Semester").agg(F.max("TotalMarks").alias("TotalMarks"),
F.max("Percentage").alias("Percentage"))
print("percentage:")
df2.show()
```

```
percentage:
+----------+--------+----------+----------+
|First Name|Semester|TotalMarks|Percentage|
+----------+--------+----------+----------+
|     Alice|       1|       253|     84.00|
|     Alice|       2|       258|     86.00|
|     Alice|       3|       283|     94.00|
|       Bob|       1|       238|     79.00|
|       Bob|       2|       281|     94.00|
|       Bob|       3|       246|     82.00|
|   Charlie|       1|       236|     79.00|
|   Charlie|       2|       262|     87.00|
|   Charlie|       3|       288|     96.00|
+----------+--------+----------+----------+
```



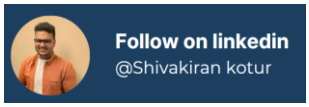
Follow me on LinkedIn – Shivakiran kotur

```
# 3. Who is the top rank holder in each semester considering all subjects
window_spec_rank = Window.partitionBy("Semester").orderBy(F.desc("Percentage"))
rank_df = df.withColumn("Rank", F.rank().over(window_spec_rank))
top_rank_holder = rank_df.filter(rank_df["Rank"] == 1).select("First Name","Semester", "Rank", "Percentage").distinct()
print("top_rank_holder:")
top_rank_holder.show()
```

```
top_rank_holder:
+----------+--------+----+----------+
|First Name|Semester|Rank|Percentage|
+----------+--------+----+----------+
|     Alice|       1|   1|     84.00|
|       Bob|       2|   1|     94.00|
|   Charlie|       3|   1|     96.00|
+----------+--------+----+----------+
```

```
# 4. Who scored max marks in each subject in each semester
window_spec_max_subject_marks = Window.partitionBy("Semester", "Subject").orderBy(F.desc("Marks"))
max_subject_marks_df = df.withColumn("Rank", F.rank().over(window_spec_max_subject_marks))
max_subject_scorer = max_subject_marks_df.filter(max_subject_marks_df["Rank"] == 1)
print("max_subject_scorer")
max_subject_scorer.show()
```

```
max_subject_scorer
+----------+----------------+-----+--------+----------+----------+----+
|First Name|         Subject|Marks|Semester|TotalMarks|Percentage|Rank|
+----------+----------------+-----+--------+----------+----------+----+
|   Charlie|         History|   79|       1|       236|     79.00|   1|
|     Alice|            Math|   90|       1|       253|     84.00|   1|
|     Alice|         Science|   85|       1|       253|     84.00|   1|
|       Bob|         Biology|   96|       2|       281|     94.00|   1|
|     Alice|       Chemistry|   92|       2|       258|     86.00|   1|
|       Bob|         Physics|   94|       2|       281|     94.00|   1|
|     Alice|Computer Science|   95|       3|       283|     94.00|   1|
|   Charlie|     Electronics|   97|       3|       288|     96.00|   1|
|   Charlie|       Geography|   99|       3|       288|     96.00|   1|
+----------+----------------+-----+--------+----------+----------+----+
```



Follow me on LinkedIn – Shivakiran kotur