Assignment – Day 14

-Sarthak Niranjan Kulkarni (Maverick)

- [sarthakkul2311@gmail.com](mailto:sarthakkul2311@gmail.com) - (+91) 93256 02791

**21/11/2024 (Thursday)**

**Joins in Spark Practice:-**

1. **Creating and displaying PySpark DataFrames with employee and department data:-**

from pyspark.sql import SparkSession

# Initialize SparkSession

spark = SparkSession.builder \

.appName("example") \

.getOrCreate()

# Data

emp = [(1,"Smith",-1,"2018","10","M",3000),(2, "Rose",1 , "2010", "20","M", 4000),(3,"Williams",1,"2010","10","M",1000),(4, "Jones",2 ,"2005","10","F",2000),(5,"Brown",2,"2010","40","",-1),(6, "Sarthak", 2, "2010","23","",-1)]

empColumns = ["emp\_id","name","superior\_emp\_id","year\_joined", "emp\_dept\_id","gender","salary"]

empDF = spark.createDataFrame(data=emp, schema = empColumns)

empDF.printSchema()

empDF.show()

dept = [("Finance",10),("Marketing",20),("Sales",30),("IT",40)]

deptColumns = ["dept\_name","dept\_id"]

deptDF = spark.createDataFrame(data=dept, schema = deptColumns)

deptDF.printSchema()

deptDF.show()

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

1. **Performing inner, outer, and full joins between employee and department DataFrames in PySpark.**

#Inner join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "inner").show()

#outer join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "outer").show()

#full join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "full").show()

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

1. **Performing left and left outer joins between employee and department DataFrames in PySpark.**

#Left join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "left").show()

#Left Outer join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "leftouter").show()

**A screenshot of a computer code

Description automatically generated**

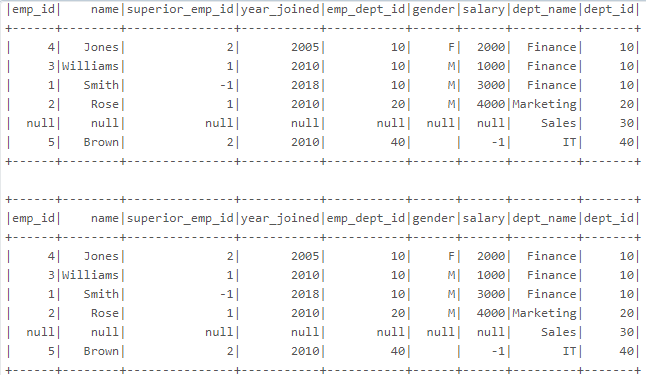
1. **Performing right and right outer joins between employee and department DataFrames in PySpark.**

#right join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "right").show()

#right outer join

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "rightouter").show()

****

1. **Performing left semi and left anti joins between employee and department DataFrames in PySpark.**

#left semijoin

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "leftsemi").show()

#left anti

empDF.join(deptDF,empDF.emp\_dept\_id == deptDF.dept\_id, "leftanti").show()

**A screenshot of a computer program

Description automatically generated**

**Joins in Spark Summary:-**

The above codes demonstrates the creation of two PySpark DataFrames: empDF containing employee data and deptDF containing department data. It showcases various types of joins to combine the two DataFrames based on the common key emp\_dept\_id in empDF and dept\_id in deptDF.

1. **Inner Join** returns rows where there is a match in both DataFrames.
2. **Outer Join** (or Full Join) includes all rows from both DataFrames, with null values for non-matching rows.
3. **Left and Right Joins** (and their outer variants) return all rows from one DataFrame and matching rows (if any) from the other.

Additionally, **Left Semi Join** filters rows in empDF that have a match in deptDF, while **Left Anti Join** returns rows in empDF that do not match with deptDF.