BDA Assignment 10

Name Devansh Thakkar
Srn no: 202201187
Roll no: 25
TY-A

Problem Statement: Join two Spark Data frames on a single column

CODE & OUTPUT:

1) **Create SparkSession**: First, we initialize a SparkSession to interact with Spark.

```
>>> from pyspark.sql import SparkSession
>>> # Initialize Spark session
>>> spark = SparkSession.builder.master("local").appName("Join Example").getOrCr
eate()
24/11/14 10:36:24 WARN SparkSession: Using an existing Spark session; only runti
me SQL configurations will take effect.
```

```
// Import Spark session
import org.apache.spark.sql.SparkSession

val spark =
SparkSession.builder().appName("JoinExample").master("local[*]").getOrCrea
te()
import spark.implicits._

// Create the DataFrames
df1 = Seq(
  (1, "Alice", 25),
  (2, "Bob", 30),
  (3, "Charlie", 35)
).toDF("id", "name", "age")

val df2 = Seq(
  (1, "New York", "USA"),
  (2, "London", "UK"),
  (4, "Paris", "France")
```

).toDF("id", "city", "country")

// Perform an inner join on the 'id' column

val joinedDf = df1.join(df2, Seq("id"), "inner")

// Show the joined DataFrame

joinedDf.show()

2) **Create DataFrames**: We create two DataFrames:

- df1: Contains employee names, departments, and salaries.
- df2: Contains department names and their corresponding full names.

```
>>> # Create the first DataFrame
>>> data1 = [
...     ("Alice", "HR", 3000),
...     ("Bob", "Finance", 4000),
...     ("Charlie", "IT", 5000)
... ]
>>> columns1 = ["Name", "Department", "Salary"]
>>> df1 = spark.createDataFrame(data1, columns1)
>>> # Create the second DataFrame
>>> data2 = [
...     ("HR", "Human Resources"),
...     ("Finance", "Finance Department"),
...     ("IT", "Information Technology")
... ]
>>> columns2 = ["Department", "Department_Name"]
>>> df2 = spark.createDataFrame(data2, columns2)
```

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

df1 = [

... ("Alice","HR",3000),

... ("Bob","Finance",4000),

... ("Charlie","Tech",5000)]

>>> columns1=["Name","Department","Salary"]

>>> data1=spark.createDataFrame(df1,columns1)

>>> data1.show()

df2=[

... ("HR","Human Resources"),

... ("IT","Information Technology"),

... ("Finance","Finance Department")]

>>> columns2=["Dept","Description"]

>>> data2=spark.createDataFrame(df2,columns2)

>>> data2.show()

```
>>> df1.show()
+-------+----------+------+
|   Name|Department|Salary|
+-------+----------+------+
|  Alice|        HR|  3000|
|    Bob|   Finance|  4000|
|Charlie|        IT|  5000|
+-------+----------+------+
```

3) **Join Operation**:

- **df1.join(df2, on="Department", how="inner")**:
    - on="Department": Specifies that the join should be performed on the "Department" column.
    - how="inner": Specifies the type of join. Here we are using an **inner join**, which returns rows when there is a match in both DataFrames. Other join types include left, right, and outer.

```
>>> result = df1.join(df2, on="Department", how="inner")
>>> result.show()
```

4) **Show the Result**: The show() method is used to display the result of the join.

```
>>> result.show()
+----------+-------+------+--------------------+
|Department|   Name|Salary|     Department_Name|
+----------+-------+------+--------------------+
|   Finance|    Bob|  4000|  Finance Department|
|        HR|  Alice|  3000|     Human Resources|
|        IT|Charlie|  5000|Information Techn...|
+----------+-------+------+--------------------+
```

5) **Join Types:**

- **inner**: Returns rows that have matching values in both DataFrames.
- **left**: Returns all rows from the left DataFrame and matching rows from the right DataFrame. If no match, the result will have null values for columns from the right DataFrame.

- **right**: Returns all rows from the right DataFrame and matching rows from the left DataFrame. If no match, the result will have null values for columns from the left DataFrame.
- **outer**: Returns all rows from both DataFrames. If no match, the result will have null values for the missing side.

```
>>> result_left = df1.join(df2, on="Department", how="left")
>>> result_left.show()result_left.show()
  File "<stdin>", line 1
    result_left.show()result_left.show()
                      ^^^^^^^^^^
SyntaxError: invalid syntax
>>> result_left.show()
+----------+-------+------+--------------------+
|Department|   Name|Salary|     Department_Name|
+----------+-------+------+--------------------+
|        HR|  Alice|  3000|     Human Resources|
|   Finance|    Bob|  4000|  Finance Department|
|        IT|Charlie|  5000|Information Techn...|
+----------+-------+------+--------------------+
```