PySpark

QUESTION-6

Given a DataFrame with columns Name and Age, remove duplicate rows based on the Name column, keeping only the latest entry based on a given timestamp column Timestamp.

DIKSHA CHOURASIYA

SOLUTION

```
+-----+---+-------------------+
| Name|Age| Timestamp|
+-----+---+-------------------+
| John| 28|2023-01-01 10:00:00|
| Alice| 35|2023-01-02 11:00:00|
| David| 32|2023-01-03 12:00:00|
| John| 29|2023-01-04 13:00:00|
| Alice| 36|2023-01-05 14:00:00|
+-----+---+-------------------+
Expected Output:
+-----+---+-------------------+
| Name|Age| Timestamp|
+-----+---+-------------------+
| David| 32|2023-01-03 12:00:00|
| John| 29|2023-01-04 13:00:00|
| Alice| 36|2023-01-05 14:00:00|
```

DIKSHA CHOURASIYA

SOLUTION

```
from pyspark.sql import SparkSession
from pyspark.sql import Window
from pyspark.sql.functions import *
spark=SparkSession.builder.appName("Question-6").getOrCreate()

#Method-1
data = [
    ("John", 28, "2023-01-01 10:00:00"),
    ("Alice", 35, "2023-01-02 11:00:00"),
    ("David", 32, "2023-01-03 12:00:00"),
    ("John", 29, "2023-01-04 13:00:00"),
    ("Alice", 36, "2023-01-05 14:00:00"),
]
schema=['Name','Age','Timestamp']
df=spark.createDataFrame(data,schema)
df.display()
print("after aggregation")
df1=df.groupBy(col('Name')).agg(max('Timestamp').alias('Timestamp'))
df1.display()
df_selected_field=df.select('Age','Timestamp')
df_joined=df_selected_field.join(df1,on='Timestamp',how='inner')
df_joined.display()
```

DIKSHA CHOURASIYA

SOLUTION

```
#Method-2
window=Window.partitionBy('Name').orderBy(desc('Timestamp'))
df2=df.withColumn('records_with_rank_1',row_number().over(window))
df_first_record=df2.filter(df2['records_with_rank_1']==1).orderBy(asc('Timestamp'))
df_first_record.display()
```

▸ (11) Spark Jobs

Table

	Name	Timestamp
1	Alice	2023-01-05 14:00:00
2	David	2023-01-03 12:00:00
3	John	2023-01-04 13:00:00

DIKSHA CHOURASIYA

THANKYOU

DIKSHA CHOURASIYA