10 Pyspark Dataframe interview questions with solution

Here are 10 critical interview questions on Spark DataFrame operations along with their solutions:

Question 1: How do you create a DataFrame in Spark from a collection of data?

Solution:

```
from pyspark.sql import SparkSession

# Initialize Spark session
spark = SparkSession.builder.appName("CreateDataFrame").getOrCreate()

# Sample data
data = [("John", 25), ("Doe", 30), ("Jane", 28)]

# Create DataFrame
columns = ["name", "age"]
df = spark.createDataFrame(data, columns)

# Show DataFrame
df.show()

# Stop Spark session
spark.stop()
```

Question 2: How do you select specific columns from a DataFrame?

Solution:

```
# Select specific columns
selected_df = df.select("name", "age")

# Show DataFrame
selected_df.show()
```

Question 3: How do you filter rows in a DataFrame based on a condition?

Solution:

```
# Filter rows where age is greater than 25
filtered_df = df.filter(df["age"] > 25)

# Show DataFrame
filtered_df.show()
```

Question 4: How do you group by a column and perform an aggregation in Spark DataFrame?

Solution:

```
# Sample data
data = [("John", "HR", 3000), ("Doe", "HR", 4000),("Jane",
"IT", 5000), ("Mary", "IT", 6000)]

# Create DataFrame
columns = ["name", "department", "salary"]
df = spark.createDataFrame(data, columns)

# Group by department and calculate average salary
avg_salary_df = df.groupBy("department").avg("salary")
```

Ref: codeinspark.com

```
# Show the result
avg_salary_df.show()
```

Question 5: How do you join two DataFrames in Spark?

Solution:

```
# Sample data
data1 = [("John", 1), ("Doe", 2), ("Jane", 3)]
data2 = [(1, "HR"), (2, "IT"), (3, "Finance")]

# Create DataFrames
columns1 = ["name", "dept_id"]
columns2 = ["dept_id", "department"]

df1 = spark.createDataFrame(data1, columns1)
df2 = spark.createDataFrame(data2, columns2)

# Join DataFrames on dept_id
joined_df = df1.join(df2, "dept_id")

# Show the result
joined_df.show()
```

Question 6: How do you handle missing data in Spark DataFrame?

Solution:

```
# Sample data
data = [("John", None), ("Doe", 25), ("Jane", None), ("Mary", 30)]

# Create DataFrame
columns = ["name", "age"]
df = spark.createDataFrame(data, columns)

# Fill missing values with a default value
df_filled = df.fillna({'age': 0})

# Show the result
df_filled.show()
```

Question 7: How do you apply a custom function to a DataFrame column using UDF?

Solution:

```
from pyspark.sql.functions
import udf from pyspark.sql.types
import StringType


# Define UDF to convert department to uppercase
def convert_uppercase(department):
       return department.upper()


# Register UDF
convert_uppercase_udf = udf(convert_uppercase, StringType())

# Apply UDF to DataFrame
df_transformed = df.withColumn("department_upper", convert_uppercase_udf(df["department"]))

# Show the result
df_transformed.show()
```

Ref: codeinspark.com

Question 8: How do you sort a DataFrame by a specific column?

Solution:

```
# Sort DataFrame by age
sorted_df = df.orderBy("age")


# Show the result
sorted_df.show()
```

Question 9: How do you add a new column to a DataFrame?

Solution:

```
# Add a new column with a constant value
df_with_new_column = df.withColumn("new_column", df["age"] * 2)

# Show the result
df_with_new_column.show()
```

Question 10: How do you remove duplicate rows from a DataFrame?

Solution:

```
# Sample data with duplicates
data = [("John", 25), ("Doe", 30), ("Jane", 28), ("John", 25)]

# Create DataFrame
columns = ["name", "age"]
df = spark.createDataFrame(data, columns)

# Remove duplicate rows
df_deduplicated = df.dropDuplicates()

# Show the result
df_deduplicated.show()
```

These questions and solutions cover fundamental and advanced operations with Spark DataFrames, which are essential for data processing and analysis using Spark.