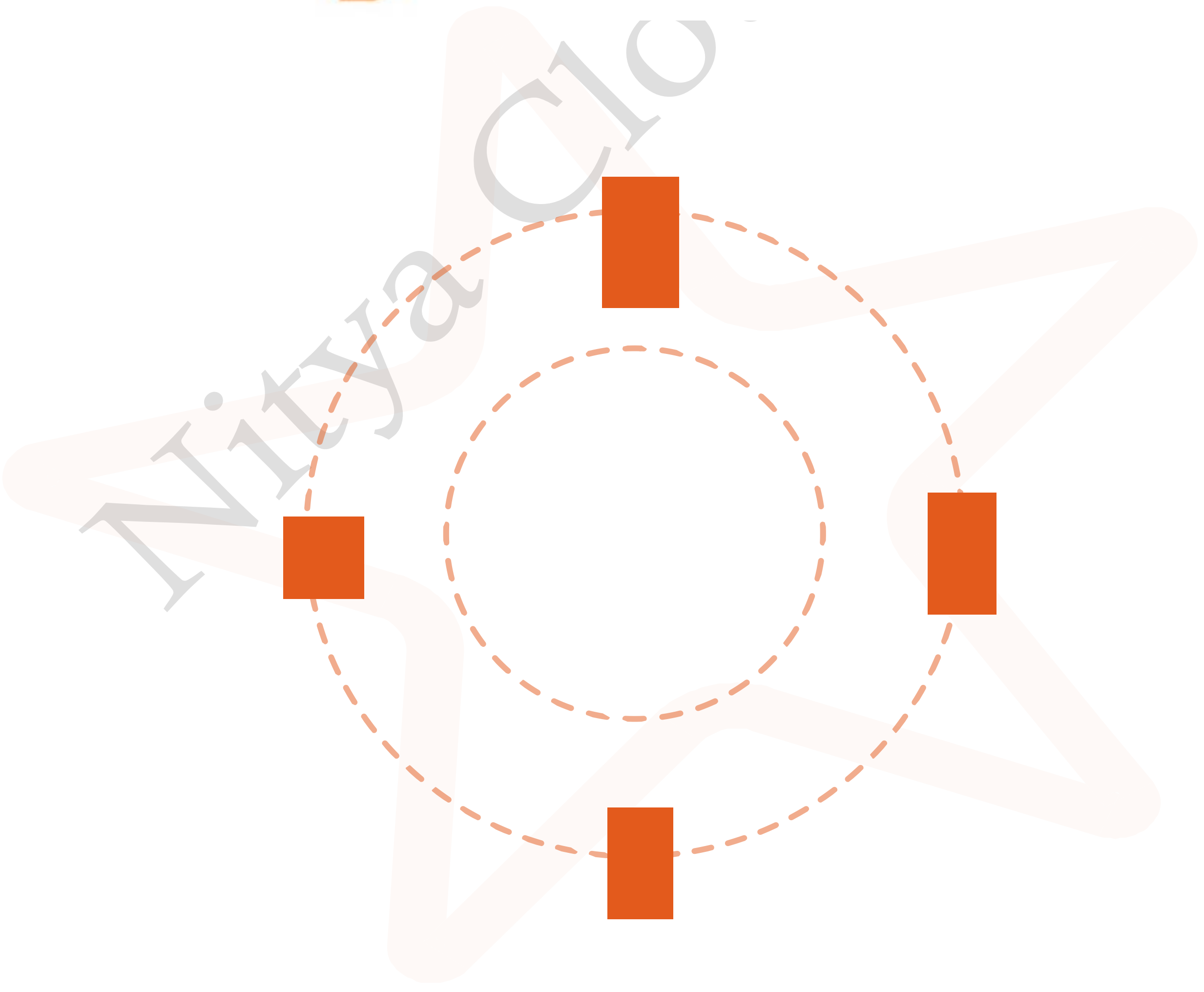
Nitya CloudTech Pvt Ltd.

PySpark Scenario-Based Interview Questions & Answers



1. Question: How would you convert a PySpark DataFrame column with a JSON string to multiple columns?

- **Answer:** Use `from_json()` along with a defined schema to parse the JSON string:

```
from pyspark.sql.functions import from_json
from pyspark.sql.types import StructType, StructField,
StringType

schema = StructType([StructField("name", StringType()),
StructField("age", StringType())])
df = df.withColumn("parsed_json",
from_json(df["json_column"],
schema)).select("parsed_json.*")
```

2. Question: How can you remove duplicate rows in a DataFrame based on specific columns?

- **Answer:** Use `dropDuplicates()`:

```
df.dropDuplicates(["col1", "col2"])
```

  This removes rows where values in the specified columns are identical, keeping only one instance.

3. Question: Describe how you would filter a DataFrame to include only rows where a column contains a specific substring.

- **Answer:** Use the `contains()` function:

```
df.filter(df["column_name"].contains("substring"))
```

  This filters rows where "column_name" includes "substring".

4. Question: How do you add a constant column to a PySpark DataFrame?

- **Answer:** Use `withColumn()` and `lit()`:

```
from pyspark.sql.functions import lit
df = df.withColumn("new_column", lit("constant_value"))
```

5. Question: How would you convert a DataFrame column from one data type to another?

- **Answer:** Use `cast()` to change the data type:

```
df = df.withColumn("column_name",
df["column_name"].cast("new_data_type"))
```

6. Question: Explain how to use a PySpark SQL query on a DataFrame.

- **Answer:** Register the DataFrame as a temporary view and run SQL:

```
df.createOrReplaceTempView("temp_view")
spark.sql("SELECT * FROM temp_view WHERE column > 10")
```

7. Question: How can you rename multiple columns in a PySpark DataFrame?

- **Answer:** Chain `withColumnRenamed()` or use list comprehension with `toDF()`:

```
new_column_names = ["new_name1", "new_name2"]
df = df.toDF(*new_column_names)
```

8. Question: Describe how you would pivot a DataFrame in PySpark.

- **Answer:** Use `groupBy()` with `pivot()`:

```
df.groupBy("pivot_column").pivot("category_column").agg({
"value_column": "sum"})
```

9. Question: How can you calculate the distinct count of values in a column?

- **Answer:** Use `distinct().count()` or `approx_count_distinct()`:

```
df.select("column").distinct().count()
```

10. Question: What is the difference between `select()` and `selectExpr()` in PySpark?

- **Answer:** `select()` requires column names or expressions directly, while `selectExpr()` allows SQL-like expressions as strings:

  ```
  df.selectExpr("col1 + col2 AS col_sum")
  ```

11. Question: Explain the purpose of the `groupBy()` and `agg()` functions.

- **Answer:** `groupBy()` groups rows based on column(s), and `agg()` performs aggregations:

  ```
  df.groupBy("group_column").agg({"value_column": "sum"})
  ```

12. Question: How do you handle null values in a DataFrame column?

- **Answer:** Use `fillna()` to replace nulls, or `dropna()` to remove rows with nulls:

  ```
  df.fillna({"column_name": "default_value"})
  ```

13. Question: How would you read a CSV file into a PySpark DataFrame with a header and custom delimiter?

- **Answer:**

  ```
  df = spark.read.option("header",
  "true").option("delimiter", ",").csv("path/to/file.csv")
  ```

14. Question: How can you sort a DataFrame by multiple columns in ascending and descending order?

- **Answer:** Use `orderBy()` and specify order for each column:

  ```
  df.orderBy(df["col1"].asc(), df["col2"].desc())
  ```

15. Question: Describe how you would join two DataFrames on multiple keys.

- **Answer:** Use `join()` and specify multiple conditions:

```
df1.join(df2, (df1["col1"] == df2["col1"]) & (df1["col2"]
== df2["col2"]), "inner")
```

16. Question: How do you calculate the rank of rows within each partition of a DataFrame?

- **Answer:** Use `Window` functions along with `rank()`:

```
from pyspark.sql.window import Window
from pyspark.sql.functions import rank

window_spec =
Window.partitionBy("partition_column").orderBy("order_col
umn")
df = df.withColumn("rank", rank().over(window_spec))
```

17. Question: How do you perform an inner join and filter out rows with nulls in one of the columns after the join?

- **Answer:** Use an inner join and `filter()` to remove nulls:

```
df = df1.join(df2, "join_column",
"inner").filter(df2["column"].isNotNull())
```

18. Question: How can you change the number of partitions in a DataFrame to improve performance?

- **Answer:** Use `repartition()` to increase partitions or `coalesce()` to reduce them:

```
df = df.repartition(10)   # Increases partitions
df = df.coalesce(5)       # Reduces partitions
```

19. Question: How would you aggregate data using both sum and average functions in a single query?

- **Answer:** Use `groupBy()` with `agg()`:

```
from pyspark.sql.functions import sum, avg

df.groupBy("group_column").agg(sum("value_column"),
avg("value_column"))
```

20. Question: Explain how to concatenate two string columns with a separator.

- **Answer:** Use `concat_ws()` to join with a separator:

```
from pyspark.sql.functions import concat_ws

df = df.withColumn("full_name", concat_ws(" ",
df["first_name"], df["last_name"]))
```