Menu | Logout

Java-Success.com

800+ Java & Big Data Interview Q&As with code & diagrams to fast-track & go places with choices.

search here … Go

Home 300+ Java FAQs ▾ 300+ Big Data FAQs ▾ Courses ▾ Membership ▾ Career ▾

Home › bigdata-success.com › 300+ Big Data FAQs 🔥 › FAQs Data - 08: Spark › 17: Spark interview Q&As with coding examples in pyspark (i.e. python)

17: Spark interview Q&As with coding examples in pyspark (i.e. python)

Posted on May 13, 2018

Q01. How will you create a Spark context?
A01.

```
from pyspark.sql import SparkSession

 spark = SparkSession.builder.appName("my spark job")
 spark.master('local[*]')

 spark.config('spark.jars.packages', 'com.amazonaws:aws-java-sdk:1.11
              .config('spark.hadoop.mapreduce.fileoutputcommitter.alg
              .config('spark.speculation', 'false')

```

Q02. How will you create a Dataframe by reading a file from AWS S3 bucket?
A02.

```

      csvFileAsDataframe = spark.read.format("com.databricks.spark.cs
                                  .option("header", "false") \
                                  .option("inferSchema", "true") \
                                  .load(s3://my-bucket/some-path/input-file.cs

```

Q03. How will you create a Dataframe by reading a table in a database?
A03.

```

       jdbcTableAsDataframe = self.spark.read \
                   .format("jdbc") \
                   .option("url", "jdbc:mysql://myhost:3306/mydatabas
                   .option("dbtable", "mytable") \
                   .option("user", "root") \
```

300+ Java Interview FAQs

300+ Java FAQs 🔥
16+ Java Key Areas Q&As
150+ Java Architect FAQs
80+ Java Code Quality Q&As
150+ Java Coding Q&As

300+ Big Data Interview FAQs

300+ Big Data FAQs 🔥
FAQs Data - 01: SQL
FAQs Data - 02: Data Modelling
FAQs Data - 02: Data Warehouse
FAQs Data - 03: Big Data
FAQs Data - 04: Hadoop (HDFS)
FAQs Data - 05: MapReduce
FAQs Data - 06: Hive
FAQs Data - 07: Impala
FAQs Data - 08: Spark
FAQs Data - 09: Spark SQL
FAQs Data - 10: Apache Kafka
FAQs Data - 11: Data Governance
FAQs Data - 11: NoSQL
FAQs Data - 12: Data security
FAQs Data - 13: Analytics & Science
FAQs Data - 14: AWS
FAQs Data - 15: Sqoop & Nifi
FAQs Data - 16: Yarn, Zookeeper
FAQs Data - 40: Scala 140+ FAQs

```
                    .option("password", "password") \
                    .option("driver", "com.mysql.jdbc.Driver") \
                    .option("useSSL", false) \
                    .load()

```

Q04. How will you select columns from a Dataframe?

A04.

```

    from pyspark.sql.functions import col

    outputDataframe = csvFileAsDataframe.select(col("_c0").alias("em
                                                col("_c1").alias("em
                                                col("_c2").alias("de
                                                col("_c3").alias("sa

```

Q05. How do you join two Dataframes and filter?

A05.

```

#join condition
condSendDate = [dfTable1.party_id == dfTable2.send_party_id, dfTable

#joining 3 tables
dfJoined = dfTable1.join(dfTable3, dfTable1.campaign_id== dfTable3.l
               .join(dfAction, dfTable1.action_id == dfAction.action
               .join(dfTable2, condSendDate , 'left_outer' ) \
               .filter(dfTable3['channel'].rlike("[wW][eE][bB]")) \
               .filter(to_timestamp(dfTable1.available_from_date) <=
               .filter(to_timestamp(dfTable1.available_until_date >=

```

Q06. How will you add a new column to a Dataframe?

A06. Use “withColumn” function.

```

from pyspark.sql.functions import dense_rank
from pyspark.sql.window import Window

#rank is a new column evaluated
dfTable1.withColumn("rank", dense_rank().over(windowSpec))

```

Q07. What is a window function in Spark?

A07. A **window function** calculates a return value for every input row of a table based on a group of rows. Spark SQL supports three kinds of window functions: ranking functions, analytic functions, and aggregate functions.

Students are grouped by student_ids and for each student his/her subjects are ranked by “written_test_grade” and “practical_test_grade” in descending order and thetop ranking subject is selected.

```

from pyspark.sql.window import Window
from pyspark.sql.functions import dense_rank

windowSpec = Window.partitionBy(dfTable1['student_id']) \
```

FAQs Data - 41: 100+ Python FAQs

Tutorials - Big Data

800+ Java Interview Q&As

300+ Core Java Q&As

300+ Enterprise Java Q&As

150+ Java Frameworks Q&As

120+ Companion Tech Q&As

Tutorials - Enterprise Java

```
           .orderBy(dfTable1['written_test_grade'].desc(), dfTable1[

dfTable1Ranked = dfTable1.withColumn("rank", dense_rank().over(windo
dfTable1Toprank =  dfTable1Ranked.where(dfTable1Ranked.rank == 1)

```

Q08. When performing Dataframe operations, how do you verify your results?
A08. The "show()" and "printSchema()" methods are very handy for debugging.

```

myDataframe.show()     # prints the top 20 rows in the dataframe
myDataframe.show(100)  # prints the top 100 rows in the dataframe

myDataframe.printSchema()   # prints the schema of the Dataframe

```

Q09. What is a Spark udf?
A09. udf stands for **U**ser **D**efined **F**unctions. Here is an example to create a new column by adding a few days to a given date column.

Spark let's you define custom SQL functions called user defined functions (UDFs). UDFs are great when built-in SQL functions aren't sufficient, but should be used sparingly because they're not performant.

```
from pyspark.sql.types import DateType,TimestampType
from pyspark.sql.functions import udf
from pyspark.sql.functions import to_date,to_timestamp

date_calc_udf = udf(funcDateCalc, DateType())

dfResult = dfTable1.withColumn("p_calc_ts, date_calc_udf(dfTable1['r

def funcDateCalc(inputDate, sendDate,  flag, addDays):

    from datetime import datetime, timedelta

    if(flag == 1):
        if(sendDate is None):
            modified_date = inputDate + timedelta(days=int(addDays))
            return modified_date
        else:
            modified_date = sendDate + timedelta(days=int(addDays))
            return modified_date
    else:
        return None

```

Q10. How do you write the Dataframe results to a relational database table?
A10.

```

dfResult.write \
    .mode('overwrite') \
    .format("jdbc") \
    .option("url", jdbcUrlWrite) \
    .option("dbtable", "my-table") \
    .option("user", "root") \
    .option("password", "password") \
    .option("batchsize", 10000) \
    .option("driver", "com.mysql.jdbc.Driver") \
    .option("truncate", 'false') \
    .option("useSSL", false) \
    .save()

```

Q11. How do you write the Dataframe results to a AWS s3 bucket?
A11.

```

dfResult.coalesce(1)
         .write.csv("s3://my-bucket/some-path/output-file.csv", sep="|

```

Q12. What is a lit function?
A12. “lit” is a Spark SQL built-in function. It creates a Column of literal value.

```

from pyspark.sql.functions import lit
from datetime import date,datetime

send_date = datetime.now().date()
dfTable1.withColumn("run_date", lit(send_date ))

```

Q13. How do you create an SQLContext?
A13.

```

from pyspark.sql import SQLContext

spark = SparkSession.builder.appName("my-saprk")
sc = spark.sparkContext
sqlContext = SQLContext(sc)

```

Q14. How do you use sql like syntax in SparkSQL?
A14.

```

dfResult = spark.sql("SELECT * FROM dfTable1 WHERE NOT (calc_date  IS

```

Note: Refer to the pyspark SQL module API for more detail – http://spark.apache.org/docs/2.1.0/api/python/pyspark.sql.html.

Brush up with more examples? PySpark on Databricks examples.

‹ 21: Q138 – Q142 AWS data lakes overview Interview Q&As

01: Docker tutorial with Java & Maven ›

Arulkumaran

Mechanical Engineer to self-taught Java engineer within 2 years & a **freelancer** within 3 years. Freelancing since 2003. Preparation empowered me to **attend 190+ job interviews** & choose from **150+ job offers** with sought-after contract rates. Author of the book "**Java/J2EE job interview companion**", which sold **35K+ copies** & superseded by this site with 2,050+ registered users. Amazon.com profile | **Reviews** | LinkedIn | LinkedIn Group | YouTube

Contact us: java-interview@hotmail.com

Disclaimer

The contents in this Java-Success are copyrighted and from EmpoweringTech pty ltd. The EmpoweringTech pty ltd has the right to correct or enhance the current content without any prior notice. These are general advice only, and one needs to take his/her own circumstances into consideration. The EmpoweringTech pty ltd will not be held liable for any damages caused or alleged to be caused either directly or indirectly by these materials and resources. Any trademarked names or labels used in this blog remain the property of their respective trademark owners. Links to external sites do not imply endorsement of the linked-to sites. Privacy Policy