PySpark Q&A

Youtube : #pysparkinterviewquestions

1. What is spark?

Apache Spark is an open-source, distributed processing system used for big data workloads. It utilizes in-memory caching and optimized query execution for fast queries against data of any size. Simply put, Spark is a fast and general engine for large-scale data processing.

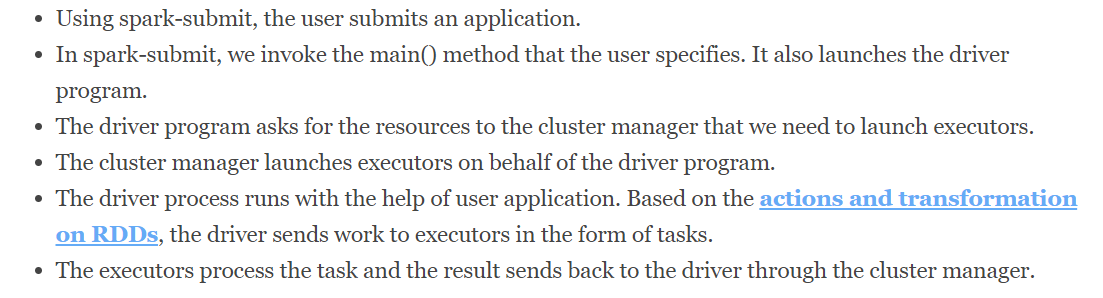
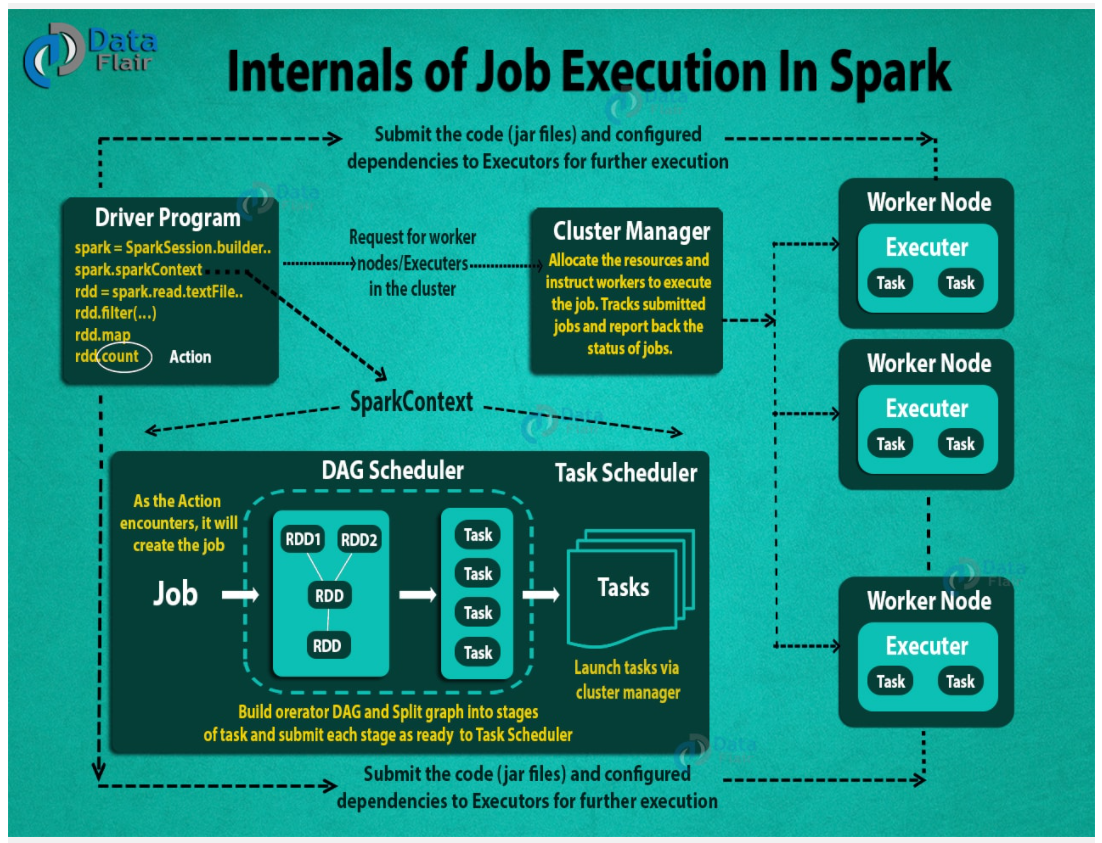
1. Which is the spark version and python version you have worked with in your project?
2. What is pyspark?

It is an apache spark API built in python.

1. What are the components of pyspark ecosystem?

Spark SQL, Spark Streaming, MLlib, GraphX, SparkR and the Core API component.

1. What are the internals of job execution in spark? Or what is the job flow in spark?



1. What is a driver?

The spark driver is the program that declares the transformations and actions on RDDs of data and submits such requests to the master. In practical terms, the driver is the program that creates the SparkContext, connecting to a given Spark Master

1. What is a transformation?

Spark Transformation is a function that produces new RDD from the existing RDDs. It takes RDD as input and produces one or more RDD as output. Each time it creates new RDD when we apply any transformation.

1. What is an action?

RDD actions are operations that return the raw values, In other words, any RDD function that returns other than RDD[T] is considered as an action in spark programming.

1. Difference between action and transformation.

<https://blog.knoldus.com/deep-dive-into-apache-spark-transformations-and-action/>

<https://intellipaat.com/blog/interview-question/apache-spark-interview-questions/#10>

<https://www.besanttechnologies.com/pyspark-interview-questions-and-answers>

<https://www.edureka.co/blog/interview-questions/top-apache-spark-interview-questions-2016/>

1. What are broadcast variables?

Broadcast variables allow the programmer to keep a read-only variable cached on each machine rather than shipping a copy of it with tasks. They can be used, for example, to give every node a copy of a large input dataset in an efficient manner.

1. What are accumulator variables?

PySpark Accumulators are shared variables that can be updated by executors and propagates back to driver program. These variables are used to add sum or counts and final results can be accessed only by driver program.

1. What is fold?

It takes function as an input which has two parameters of the same type and outputs a single value of the input type.

sc.parallelize([1,25,8,4,2]).fold(0,add) 🡪 40

sc.parallelize([1,25,8,4,2], 2).fold(10,add) 🡪(40+10) +2\*10(for each partition)

1. What are closures?

closure is those variables and methods which must be visible for the executor to perform its computations on the RDD.

It is basically encapsulation of all variables and functions defining the scope of the variable

1. Practical use of coalesce

When you want to reduce the number of partitions and the data is not too huge such that it can reside in the memory, you can use coalesce instead of repartition to avoid high latency due to shuffle operations performed in repartition.

Repartition is used for increasing or decreasing the number of partitions but internally it calls coalesce with shuffle = True

1. Why is RDD resilient?

Because it is immutable

1. What is lineage and how it works in RDD and Dataframe?
2. Difference between persist and cache
3. Narrow vs wide transformation
4. What are shared variables and its uses
5. How to define custom accumulator
6. Create a UDF in spark
7. Create a hive UDF in spark
8. What are accumulators and broadcast variables
9. How to decide various parameter values in spark-submit
10. Difference between coalesce and repartition
11. Difference between rdd dataframe and dataset. When to use which one
12. What is skew join and how it works
13. Why should we use group by transformations in spark
14. How to do mapside join in spark
15. If we have 50 GB memory and 100 GB data how will spark process it
16. Challenges you faced in spark project
17. Explain map, flatmap, mappartition, foreach, foreachpartition
18. What is pair rdd and when to use them
19. Performance optimization techniques in spark
20. Difference between cluster and client mode
21. What happens if a worker node is dead
22. How to handle bad records in spark and those types?

While reading a filesystem, spark has a mode called datareadmode that has 3 options: permissive(default), dropmalformed and failfast.

Permissive allows the corrupt data

Dropmalformed removes the corrupt data

Failfast never used, fails execution

You can use a column to capture corrupt data with permissive mode with option columnNameOfCorruptRecord.

Df=spark.read.schema(df\_schema).option(“columnNameOfCorruptRecord”, “\_corrupt\_record”).option(“mode”,”PERMISSIVE”).csv(‘filename’, header=True, inferSchema=True)

OR

Use badRecordsPath option of spark.read to store rejected records in external location

1. How to get all dataframes in spark

From pyspark.sql import DataFrame

Print([k for (k,v) in globals().items() if isinstance(v, DataFrame)])

1. How to track add source file name in one of the columns in dataframe

From pyspark.sql.functions import input\_file\_name

Df=spark.read.csv(“filename”, header=True)

Df.withColumn(“filename”, input\_file\_name())

1. Get number of rows on each file in a dataframe

From pyspark.sql.functions import input\_file\_name

Df=spark.read.csv(“filename”, header=True)

Df.withColumn(“filename”, input\_file\_name()).groupBy(“filename”).count()

1. How to add partitionId in dataframe

From pyspark.sql.functions import spark\_partition\_id

Df=spark.read.csv(“filename”, header=True)

Df.withColumn(“partition\_id”, spark\_partition\_id().select(“partition\_id”).distinct())

1. How to get row count by partitioned in df

From pyspark.sql.functions import spark\_partition\_id

Df=spark.read.csv(“filename”, header=True)

Df.withColumn(“partition\_id”, spark\_partition\_id().select(“partition\_id”).count())

1. How to add sequence generated surrogate key as column in df

From pyspark.sql.functions import monotonically\_increasing\_id

Df.withColumn(“id”, monotonically\_increasing\_id())

OR

Using MD5

Where you can get duplicates

From pyspark.sql.functions import md5

Df.withColumn(“id”, md5(EMPNO))

OR

Using SHA2

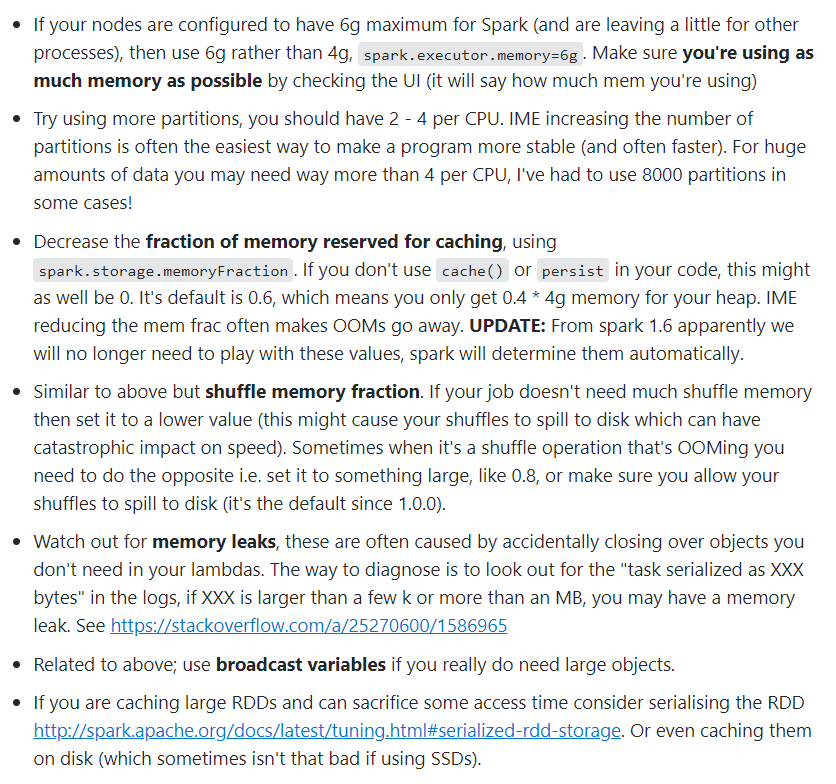
From pyspark.sql.functions import sha2

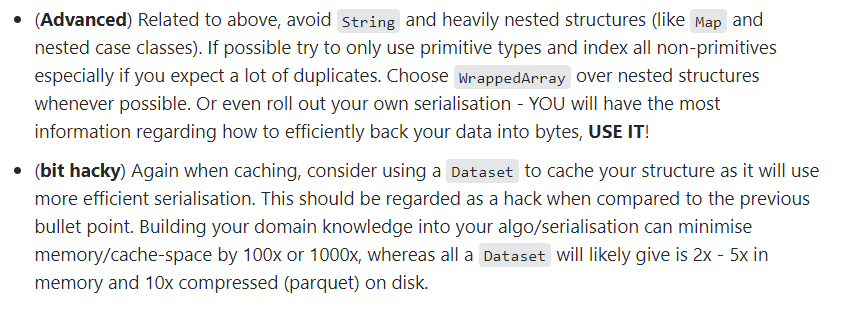
Df.withColumn(“id”, sha2(EMPNO, 256))

1. What is a global temporary view and temporary view.

Global temp view can be shared at server level and temp view at user level

1. How to handle out of memory exception?





1. How do you join spark dataframes?

<https://techvidvan.com/tutorials/apache-spark-terminologies/>

1. What is PySpark SparkContext?

PySpark SparkContext can be seen as the initial point for entering and using any Spark functionality. The SparkContext uses py4j library to launch the JVM, and then create the JavaSparkContext. By default, the SparkContext is available as ‘sc’.

1. Difference between SparkContext and SparkSession

In Spark 1.x, three entry points were introduced: SparkContext, SQLContext and HiveContext. Since Spark 2.x, a new entry point called SparkSession has been introduced that essentially combined all functionalities available in the three aforementioned contexts.

1. How do you create a spark session?

From pyspark.sql import SparkSession

spark = SparkSession.builder.appName(‘App’).getOrCreate()

1. What is the difference between inferSchema and header options in spark?

inferSchema is used to infer the schema of each and every column in the csv file. This will require an extra pass over the csv file and likely reduce the reading speed but will ensure the schema is in line with the csv file being read.

How to do it?

multiline\_df = spark.read.option("multiline","true").option("inferSchema","true").json("test.json")

Header option is used to declare that the csv being read has the first row as header and the headers used in csv become columns of the dataframe, which will be \_c0, \_c1 etc. otherwise.

1. How do you add an id column to an existing dataframe?

from pyspark.sql.functions import monotonically\_increasing\_id

df = df.select("\*").withColumn("id", monotonically\_increasing\_id())

If you want to start from 1

from pyspark.sql.functions import desc, row\_number, monotonically\_increasing\_id

from pyspark.sql.window import Window

df\_with\_seq\_id = df.withColumn('index\_column\_name', row\_number().over(Window.orderBy(monotonically\_increasing\_id())))

1. How do you create sqlContext from sparkSession?

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

sc = spark.sparkContext

sqlContext = SQLContext(sparkContext=sc, sparkSession=spark)

SQLContext needs to be explicitly imported unlike sparkContext which is like an attribute of spark session.

1. How to read a dictionary into a dataframe?

from pyspark.sql import Row

spark.createDataFrame(Row(\*\*x) for x in mylist).show(truncate=False)

1. How do you select specific columns from a dataframe?

From pyspark.sql.functions import \*

Df.select(col(“Emp\_id”)).show()

1. How do you rename a column in a df?

df=df.withColumnRenamed("\_2","Age")

1. How do you cast a column in df to IntegerType

From pyspark.sql.types import IntegerType

df.withColumn("Age",col("Age").cast(IntegerType()))

1. How can you get the count of distinct of a column?

From pyspark.sql.functions import countDistinct

df.select(countDistinct("b")).show()

1. Write code to get the maximum and minimum salary of an employee in a department.

From pyspark.sql.window import Window

From pyspark.sql.functions import \*

windowfunc1=Window.partitionBy(“Dept\_Id”).orderBy(desc(“Sal”))

windowfunc2=Window.partitionBy(“Dept\_Id”).orderBy(asc(“Sal”))

df \

.withColumn(“MaxSalaryInDept”,max(“Sal”).over(windowfunc1)) \

. withColumn(“MinSalaryInDept”,min(“Sal”).over(windowfunc2)) \

orderBy is optional

If partition by is not required,

maxDF = df.select(max(col("Salary").cast(DoubleType())))

df.crossJoin(maxDF).show()

1. Give some examples of data transformations you did in spark
2. Validation of email

from validate\_email import validate\_email

from pyspark.sql.types import BooleanType

from pyspark.sql.functions import udf

valid\_email = udf(lambda x: validate\_email(x), BooleanType())

df.withColumn('is\_valid', valid\_email('EmailAddress')).show()

1. Replacing null values with -1 in a few foreign key columns

df=df.fillna("-1")

This is needed when we have tags in xml which are having minOccurs=1 and they still have null values.

1. What are the different datatypes in pyspark.sql?

Null, String, Binary, Boolean, Date, Timestamp, Decimal, Double, Float, Byte, Integer, Long, Short, Array, Map, Struct, StructField.

1. How can you read values from a database and save them in your env variables?

Consider reading values from database with pandas read\_sql module into a pandas df.

Df=read\_sql(‘select key,value from config\_table’,connection)

Now you can read from the dataframe in following 2 ways

1. Using pandas

For ind in Df.index:

Os.environ[Df[‘key’][ind]]=Df[‘value’][ind]

1. Using spark dataframes

Sdf=spark.createDataFrame(Df)

For row in Sdf.rdd.toLocalIterator():

Os.environ[row.key]=row.value

1. How to read avro files

df = spark.read.format("avro").load("examples/src/main/resources/users.avro")

df.select("name", "favorite\_color").write.format("avro").save("namesAndFavColors.avro")

1. How to use parquet file in pyspark?

peopleDF = spark.read.json("examples/src/main/resources/people.json")

# DataFrames can be saved as Parquet files, maintaining the schema information.

peopleDF.write.parquet("people.parquet")

# Read in the Parquet file created above.

# Parquet files are self-describing so the schema is preserved.

# The result of loading a parquet file is also a DataFrame.

parquetFile = spark.read.parquet("people.parquet")

# Parquet files can also be used to create a temporary view and then used in SQL statements.

parquetFile.createOrReplaceTempView("parquetFile")

teenagers = spark.sql("SELECT name FROM parquetFile WHERE age >= 13 AND age <= 19")

teenagers.show()

1. How do you read json files?

Spark.read.json() however, each line must be a valid json object

1. When to use parquet files over avro?

Parquet is a columnar format which means columns are stored in sequential blocks rather than a record, hence retrieving columns at once becomes faster than any row based file format in which all records will be loaded, parsed for columns and then columns will be returned.

1. How do you rename columns of a spark df?

df.withColumnRenamed("dob","DateOfBirth").printSchema()

Or directly in spark.sql use as alias for columns

from pyspark.sql.functions import \*

df.select(col("name.firstname").alias("fname"), \

col("name.middlename").alias("mname"), \

col("name.lastname").alias("lname"), \

col("dob"),col("gender"),col("salary")) \

.printSchema()

Or

from pyspark.sql.functions import \*

df4 = df.withColumn("fname",col("name.firstname")) \

.withColumn("mname",col("name.middlename")) \

.withColumn("lname",col("name.lastname")) \

.drop("name")

df4.printSchema()

or

newColumns = ["newCol1","newCol2","newCol3","newCol4"]

df.toDF(\*newColumns).printSchema()

1. Difference between flatten and explode

Explode will return 1 row for each exploded element from the array

Flatten will combine all elements and return single row.

1. Difference between explode and explode\_outer?

When column to explode contains null/None data, explode will just omit the record, while explode\_outer will keep the record while exploding rest of the data.

1. Difference between pandas and spark explode function?

Pandas explode function is like sparks’ explode\_outer function.

1. What is explode alternative for rdd?

flatMap

1. What are globalTempViews?

These are spark views created by createGlobalTempView function and can be accessed in different session unlike temporary local views

1. How do you read xml into a spark dataframe?

Spark.read.format(“com.databricks.spark.xml”).option().load()

1. How do you write spark dataframe to redshift database ?

Df.write.

format(“com.databricks.spark.redshift”)

.option(“url”,”redshift\_url”)

.option(“dbtable”,”table\_name”)

.option(“driver”, “com.amazon.redshift.jdbc42.Driver”)

.option(“tempdir”,”temp\_out\_dir”)

.option(“aws\_iam\_role”,”role”)

.option(“tempformat”, “CSV GZIP”)

.option(“csv\_separator”, “,”)

.option(“preactions”,”actions to be performed before writin to the table”)

.option(“postactions”, ”actions to be performed after writin to the table”)

.mode(“append”)

.save()

1. What are different modes while writing to redshift db?

Append

Overwrite

Error : Throw an error if data already exists

CreateIfNotExists

1. What option can you use if your S3 bucket and Redshift cluster are in different AWS regions

Use .option(“extracopyoptions”,”region eu-west-1”)

Or

.option(“awsregion”, “eu-west-1”)

1. What is the difference between collect and glom.collect?

When you create an rdd with partitions, collect will coalesce all elements in all partitions into a single list, but glom will collect all elements from all partitions and group them within partitions and return list of list.

Example:

Rdd=sc.parallelize([1,2,3,4],2)

Rdd.collect()

[1,2,3,4]

Rdd.glom().collect()

[[1,2],[3,4]]

1. What to do when you have Long-running Spark query that hangs indefinitely even though the corresponding Redshift operation is done

Use tcpKeepAlive JDBC flag and TCPKeepAliveMinutes to a low value.

1. What is the default format in which data is written in temp directory while writing to redhisft?

Default is AVRO, but CSV is faster.

1. How have you used spark dataframes in your project?

We had stored sql queries and config details in the database as key and value columns, to read that we used pandas read\_sql function and stored results in a var lets call it sqldetails.

Post that we used sd=spark.createDataFrame(sqldetails) and then iterated(row) over it to set those values in the environment variables using sd.rdd.toLocalIterator() and os.environ[row.key]=row.value

1. Which property would you use to avoid using show on each dataframe in jupyter?

Set the param spark.sql.repl.eagerEval.enabled to True and set the value of spark.sql.repl.eagerEval.maxNumRows to control the number of rows.

1. How do you fetch rows based on some conditions?

Df.filter(df.a==’abc’)

1. Have you used pandas UDFs?

No, but to use those we need to use the decorator @pandas\_udf or to apply pandas function defined in your program, you can use mapInPandas or applyInPandas on your spark dataframe.

1. Difference between mapInPandas and applyInPandas.

Apply works on grouped data, map yields generators.

1. Difference between merge and merge\_asof

Merge is used to perform inner join merge\_asof for left join and match is done based on nearest keys rather than exact keys

1. How do you read file from S3 bucket?

Using spark.sparkContext.wholeTextFiles()

Or spark.read.text(s3path, wholetext=True).select(input\_file\_name(), “value”).rdd

1. Can you use python or pandas function as udf in spark sql

Yes, for that you need to register the udf with

Spark.udf.register

Or

Using sqlContext.registerFunction

sqlContext.registerFunction("stringLengthString", lambda x: len(x))

sqlContext.sql("SELECT stringLengthString('test')").collect()

1. DataFrame and Spark SQL share the same execution engine so they can be interchangeably used seamlessly. Give an example of this

Df.createOrReplaceTempView(“tablename”)

Now you can run spark.sql on this tablename.

1. Difference between coalesce and repartition

Repartition can be used to increase or decrease the number of partitions, whereas coalesce can only be used to decrease.

1. How can you sort on a dataframe?

Use df.sort or df.orderby, asc for ascending and desc otherwise. Default is asc.

1. How can you use pyspark sql functions?

By importing **pyspark.sql.functions**

1. What are discretized streams?

Discretized Stream or DStream is the basic abstraction provided by Spark Streaming. It represents a continuous stream of data, either the input data stream received from source, or the processed data stream generated by transforming the input stream.

1. What are all the options available with diff formats while reading in spark?

<https://spark.apache.org/docs/2.0.2/api/java/org/apache/spark/sql/DataFrameReader.html>

1. How do you connect to redshift or aurora?

Using python’s mysql.connector for aurora

And

Psycopg2 for redshift

1. How to handle encryption of password in python?

From base64 import standar4\_b64decode

Spark Q&A

1. What all is required to get spark up and running?

Python – used anaconda distribution for that.

JDK 8.0 – spark not available for 9+

Apache spark – from Apache.com

1. Sparkcontext object is automatically created in the pyspark or scala shell.
2. What is Maven?

Maven is a popular package management tool for Java-based languages that lets you link to libraries in public repositories.

1. How do you run python standalone applications for spark?

By using spark-submit script which includes spark dependencies in Python.

1. How do you create a spark session?

**from** **pyspark.sql** **import** SparkSession

spark = SparkSession.builder.getOrCreate()

1. How can you control the number of rows that you can show on a jupyter notebook with pyspark dataframe?

By using variables like :

spark.conf.set('spark.sql.repl.eagerEval.enabled', True)

This will set the eager evaluation to true and when the following value is set to a specific number only those number of lines will be shown.

spark.sql.repl.eagerEval.maxNumRows

1. How to collect the dataframe data from all executors at the driver?

Use df.collect()

1. Convert a spark df to pandas df

Df.toPandas()

1. How can you create a new column in existing df?

Use the withColumn function as below:

df.withColumn('upper\_c', upper(df.c)).show()

1. How to apply conditions to filter data from a dataframe?

df.fliter(df.a == 1).show()

1. How can you run pandas functions directly on your spark dataframes?

Use mapInPandas() function

def pandas\_filter\_func(iterator):

for pandas\_df in iterator:

yield pandas\_df[pandas\_df.a == 1]

sdf.mapInPandas(pandas\_filter\_func, schema=sdf.schema).show()