Udemy Course Databricks Certified Developer for Spark 3.0 (Python) Practice Exams

Exam Tips & Tricks

- Revision 6: February 2022 -

This document gives you handy tips & tricks to prepare for the Databricks Certified Developer for Spark 3.0 exam. It also includes a link to the PDF version of the Spark documentation, like the one you will receive during the actual exam.

Tips & Tricks

- You have 120 minutes to answer all 60 questions. So, on average 2 minutes per question. If you are familiar with the type and scope of questions (which you will be after taking the practice tests), this may feel like a lot of time during the exam. Use the extra time you have to review your answers and dig into the documentation for any doubts that remain. Scrolling, finding information, and reading in the documentation can take some time (practice with the PDF Documentation below).
- You need to answer 70% of all question correctly to pass the exam, or at least 42 questions (so you can answer 18 questions incorrectly and still pass the exam).
- The breakdown of questions is as follows:
    - Spark Architecture: Conceptual understanding (~17%, ca. 10 questions)
    - Spark Architecture: Applied understanding (~11%, ca. 7 questions)
    - Spark DataFrame API Applications (~72%, ca. 43 questions)

  You can see that even if you get all questions other than the DataFrame API Applications wrong, you still have a chance to pass the exam. Given how the topics are distributed, you should therefore focus your preparations on the DataFrame API Applications. These questions tend to be a little bit easier than the architecture questions. However, while some architecture questions may be hard, others are relatively easy – so you can score some quick wins by having a rough understanding of the architecture, without knowing too much detail. This is, of course, if your only objective is to pass the exam. If you want to pass with a great score, you should obviously also dive deeper into the Spark architecture during your studies.

- Questions about the DataFrame API may include answer options with wrong syntax around column names. So, for example, a select statement may have options like `select("myColumn")`, `select(myColumn)`, `select(col(myColumn))`, or `select(col("myColumn"))`. To quickly weed out wrong answers, you should make sure you understand which is the correct syntax to use here.
- To avoid any trouble during the online-proctored exam, make sure you understand the rule of the online proctor with regards to the setup of your room and devices. Most likely, you will have to take the exam alone, on an uncluttered desk and without any noise or people around. Take enough time before the exam to setup your environment accordingly. This helps keep anxiety down and helps you focus on solving the exam questions.
- Databricks says that you will be provided with "a digital notepad for taking notes and writing example code". This is might just end up being a text field in the test-taking software. So, you should not expect to be able to sketch out any big ideas during the exam. On a positive note though: If you been able to solve all practice exam questions without needing to take any notes, then chances are you will not need to use the text field that might be provided during the exam.
- Databricks currently makes the O'Reilly's Learning Spark 2.0 book freely available for download via this link. The book is referenced in the practice tests multiple times and it is a great read for preparing for the Spark exam and chapters 1-7 are even recommended for the exam by Databricks.
- Databricks published a free, full 60-question practice exam that I would highly suggest you review. It mimics the difficulty, style, and syllabus of the actual exam quite well. The exam even includes a solution key which shows the correct answers (although it does not give explanations). It is a great resource in addition to the practice tests in this course. Access Databricks' free practice exam here.

If you have more tips and tricks or would like to share some of your exam experience, please write them down for our test taker community in the Udemy course Q&A!

PDF Documentation

On its website, Databricks states:

> "During the exam, candidates will be provided with a PDF version of the Apache Spark documentation for the language in which they are taking the exam and a digital notepad for taking notes and writing example code."

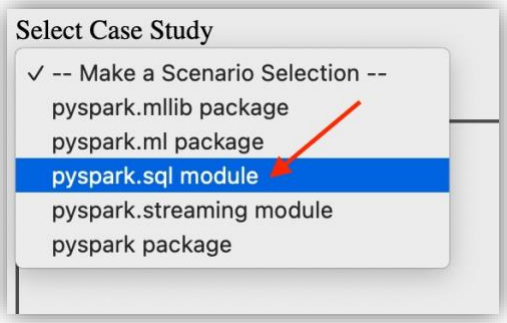
How this PDF is presented may depend on the company that conducts the exam on behalf of Databricks (e.g. Kryterion in the US). In June 2021, Databricks made the documentation available, just like you would be able to access it during the Kryterion-proctored exam:

Link to the official Databricks PDF documentation

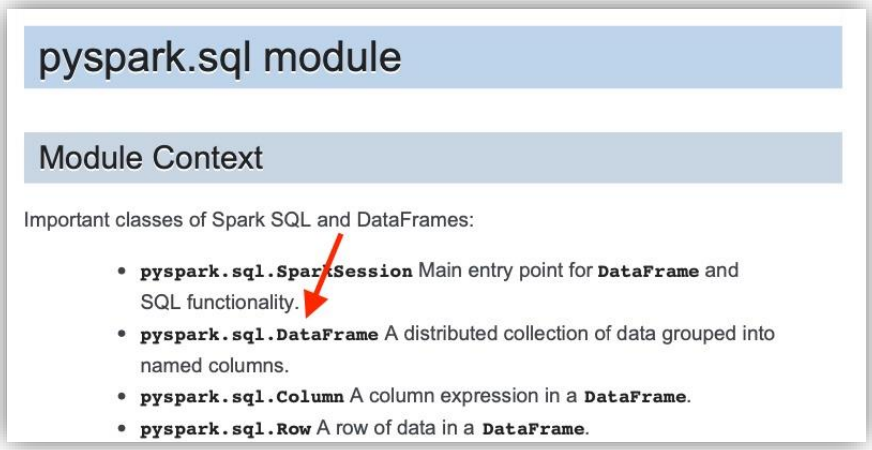
The PDF may appear as a separate window in the testing software. Others have had to ask the assessor to receive the PDF (in testing software, there is usually a chat feature to contact the assessor directly, or, in test centers, you may be able to ask the staff in attendance).

Depending on the way that the PDF is presented, **you may not be able to use the search function**. This means that you should be familiar with the structure of the PDF and the Spark API and prepare to do some scrolling. However, **the PDF may contain links**.

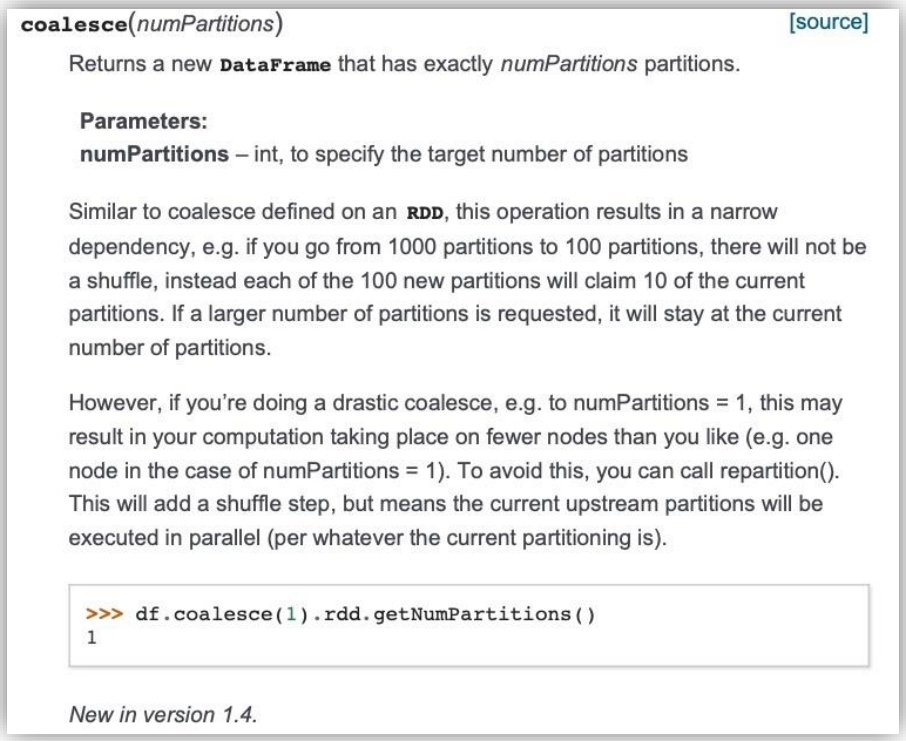
Now, how can these links help you navigate the document quickly? For example, let's assume you would be searching for details on the `coalesce()` method. You know that it is part of the `DataFrame` class. So, you would first need to select the `pyspark.sql` module from the case study drop down:



Then, first click on the `DataFrame` link at the beginning of the document:



Then, you would scroll down to the section about `coalesce`:



Before taking the exam, make sure to practice navigating the PDF! Even better, try to just use the PDF while solving the practice exams instead of being tempted to look things up in the actual online Spark documentation.

Focus Area: Column Names vs. Column Objects

When you work with PySpark for the first time, the way to pass columns to methods can be confusing. Sometimes, a method expects a column name as a string. Sometimes, you are expected to pass a `Column` object. This section aims to clear up some of that confusion. It is an excerpt from a related Q&A question which many students found helpful.

The easiest way to find out which type to use for columns (column names as strings or `Column` objects) is to pay attention to the details in the documentation:

1. Explanation of Input Parameters
Some methods in the documentation have an explanation for the input parameters that explicitly tells you which variable type to use. For this reason, the documentation can be of incredible value during the exam.

Example from the documentation:

> `DataFrame.orderBy(*cols, **kwargs)`
> Returns a new `DataFrame` sorted by the specified column(s).
> **Parameters:**
> **cols : str, list, or Column, optional**
> list of Column or column names to sort by.
> [...]

Here, the documentation makes clear that you can either use a string, Column, or even a list of those types for sorting.

For example, you can create a list of `Column` objects in different ways:

- `[df.age, df.name]`
- `[col("age"), col("name")]`

Note the special purpose of the `col()` function here in the `pyspark.sql.functions` module – it returns a `Column` object based on the name of a column. The `col()` function needs a reference to a DataFrame, but gets that from the context of the query.

For example, in `df.orderBy([col("age"), col("name")]),col()` would look at DataFrame `df` to identify columns age and name.

Passing string variables as column names is easy: `["age", "name"]` as a list or simply `"age"` if you just want to order by a single column.

Spark is incredibly flexible in some places and more restrictive in other places when it comes to accepting arguments. Given DataFrame `df`, here are several ways to tell Spark to sort it by columns age and name:

- `df.orderBy("age", "name")`
- `df.orderBy(["age", "name"])`
- `df.orderBy(col("age"), col("name"))`
- `df.orderBy([col("age"), col("name")])`
- `df.orderBy(df.age, df.name)`
- `df.orderBy([df.age, df.name])`
- `df.orderBy(df["age"], df["name"])`
- `df.orderBy([df["age"], df["name"]])`

2. Code Examples
It is always helpful to look at the code examples that are provided with some commands in the documentation. They often show multiple ways of achieving the goal. During the exam, these code examples may help you weed out some wrong answers right away.

3. Default Assumptions
When the documentation does not point out the specific object type, assume an object of `Column` type is expected. Typically, the documentation explicitly tells you when a command expects a string.

4. Insights from Context
Sometimes the context makes clear which type to use. For example, for applying functions to columns in Spark, you would never apply a function to a string of a column name directly. Something like `df.orderBy("age".asc_nulls_last())` is not correct Python syntax.

You would have to use `df.orderBy(col("age").asc_nulls_last())` or `df.orderBy(df.age.asc_nulls_last())` instead.

Focus Area: Simulating the Exam Environment

Several students have asked for a way to simulate the real exam. The real exam is delivered via the proprietary Kryterion software which we cannot use for exam preparation.

But, I think there is a good way to simulate the exam with the tools you have at home. Here is the suggested setup: