Associate-Developer-Apache-Spark practice tests

Free demo are available for Associate-Developer-Apache-Spark study materials for you to have a try before purchasing, which will help you have a deeper understanding of what you are going to buy. You can find the free demo for Associate-Developer-Apache-Spark exam braindumps in our website. If you are quite satisfied with the free demo, and want the complete version, just add it to the cart and pay for it. You will get the downloading link and password for the Associate-Developer-Apache-Spark Study Materials within ten minutes, if you don't receive, you can ask for help from our service stuff.

The Databricks Certified Associate Developer for Apache Spark 3.0 Exam certification exam covers various topics, including Spark's core concepts, dataframes, transformations, actions, and Spark SQL. Associate-Developer-Apache-Spark exam also tests the candidate's knowledge and skills in Spark's machine learning and streaming capabilities. Associate-Developer-Apache-Spark exam is designed to test the candidate's ability to write Spark applications using Scala or Python, and they are expected to have experience working with Spark's APIs.

Databricks Certified Associate Developer for Apache Spark 3.0 exam is the latest version of the certification. It covers the latest features and updates in Apache Spark 3.0, such as adaptive query execution, dynamic partition pruning, and improved SQL performance. Associate-Developer-Apache-Spark Exam consists of 60 multiple-choice questions that need to be answered within 90 minutes. The passing score for the exam is 70%.

>> Relevant Associate-Developer-Apache-Spark Answers <<

Associate-Developer-Apache-Spark Exam Torrent & Associate-Developer-Apache-Spark Exam Bootcamp & Associate-Developer-Apache-Spark Exam Cram

Don't mind what others say, trust you and make a right choice. We hope that you understand our honesty and cares, so we provide free demo of Associate-Developer-Apache-Spark exam software for you to download before you purchase our dump so that you are rest assured of our dumps. After your payment of our dumps, we will provide more considerate after-sales service to you. Once the update of Associate-Developer-Apache-Spark Exam Dump releases, we will inform you the first time. You will share the free update service of Associate-Developer-Apache-Spark exam software for one year after you purchased it.

Databricks Associate-Developer-Apache-Spark Certification Exam is a globally recognized certification program that validates the skills and knowledge required to develop and deploy Apache Spark applications using the Databricks platform. Databricks Certified Associate Developer for Apache Spark 3.0 Exam certification exam is designed for developers and data engineers who are familiar with Apache Spark and are looking to validate their skills in the field. Databricks Certified Associate Developer for Apache Spark 3.0 Exam certification exam is offered by Databricks, a leading cloud-based platform for data engineering, data science, and analytics.

Databricks Certified Associate Developer for Apache Spark 3.0 Exam Sample Questions (Q153-Q158):

NEW QUESTION # 153
Which of the following code blocks returns a copy of DataFrame transactionsDf where the column storeId has been converted to

string type?

- A. transactionsDf.withColumn("storeId", convert("storeId").as("string"))
- B. transactionsDf.withColumn("storeId", col("storeId").convert("string"))
- C. transactionsDf.withColumn("storeId", convert("storeId", "string"))
- D. transactionsDf.withColumn("storeId", col("storeId").cast("string"))
- E. transactionsDf.withColumn("storeId", col("storeId", "string"))

Answer: D

Explanation:
Explanation
This question asks for your knowledge about the cast syntax. cast is a method of the Column class. It is worth noting that one could also convert a column type using the Column.astype() method, which is just an alias for cast.
Find more info in the documentation linked below.
More info: pyspark.sql.Column.cast - PySpark 3.1.2 documentation
Static notebook | Dynamic notebook: See test 2

NEW QUESTION # 154
Which of the following describes characteristics of the Spark UI?

- A. The Scheduler tab shows how jobs that are run in parallel by multiple users are distributed across the cluster.
- B. Via the Spark UI, stage execution speed can be modified.
- C. Some of the tabs in the Spark UI are named Jobs, Stages, Storage, DAGs, Executors, and SQL.
- D. There is a place in the Spark UI that shows the property spark.executor.memory.
- E. Via the Spark UI, workloads can be manually distributed across executors.

Answer: D

Explanation:
Explanation
There is a place in the Spark UI that shows the property spark.executor.memory.
Correct, you can see Spark properties such as spark.executor.memory in the Environment tab.
Some of the tabs in the Spark UI are named Jobs, Stages, Storage, DAGs, Executors, and SQL.
Wrong - Jobs, Stages, Storage, Executors, and SQL are all tabs in the Spark UI. DAGs can be inspected in the "Jobs" tab in the job details or in the Stages or SQL tab, but are not a separate tab.
Via the Spark UI, workloads can be manually distributed across distributors.
No, the Spark UI is meant for inspecting the inner workings of Spark which ultimately helps understand, debug, and optimize Spark transactions.
Via the Spark UI, stage execution speed can be modified.
No, see above.
The Scheduler tab shows how jobs that are run in parallel by multiple users are distributed across the cluster.
No, there is no Scheduler tab.

NEW QUESTION # 155
Which of the following code blocks returns the number of unique values in column storeId of DataFrame transactionsDf?

- A. transactionsDf.select(distinct("storeId")).count()
- B. transactionsDf.select(count("storeId")).dropDuplicates()
- C. transactionsDf.distinct().select("storeId").count()
- D. transactionsDf.select("storeId").dropDuplicates().count()
- E. transactionsDf.dropDuplicates().agg(count("storeId"))

Answer: D

Explanation:
Explanation
transactionsDf.select("storeId").dropDuplicates().count()
Correct! After dropping all duplicates from column storeId, the remaining rows get counted, representing the number of unique

values in the column.
transactionsDf.select(count("storeId")).dropDuplicates()
No. transactionsDf.select(count("storeId")) just returns a single-row DataFrame showing the number of non-null rows. dropDuplicates() does not have any effect in this context.
transactionsDf.dropDuplicates().agg(count("storeId"))
Incorrect. While transactionsDf.dropDuplicates() removes duplicate rows from transactionsDf, it does not do so taking only column storeId into consideration, but eliminates full row duplicates instead.
transactionsDf.distinct().select("storeId").count()
Wrong. transactionsDf.distinct() identifies unique rows across all columns, but not only unique rows with respect to column storeId. This may leave duplicate values in the column, making the count not represent the number of unique values in that column.
transactionsDf.select(distinct("storeId")).count()
False. There is no distinct method in pyspark.sql.functions.

NEW QUESTION # 156
Which of the following describes the role of tasks in the Spark execution hierarchy?

- A. Tasks with wide dependencies can be grouped into one stage.
- B. Tasks are the smallest element in the execution hierarchy.
- C. Stages with narrow dependencies can be grouped into one task.
- D. Within one task, the slots are the unit of work done for each partition of the data.
- E. Tasks are the second-smallest element in the execution hierarchy.

Answer: B

Explanation:
Explanation
Stages with narrow dependencies can be grouped into one task.
Wrong, tasks with narrow dependencies can be grouped into one stage.
Tasks with wide dependencies can be grouped into one stage.
Wrong, since a wide transformation causes a shuffle which always marks the boundary of a stage. So, you cannot bundle multiple tasks that have wide dependencies into a stage.
Tasks are the second-smallest element in the execution hierarchy.
No, they are the smallest element in the execution hierarchy.
Within one task, the slots are the unit of work done for each partition of the data.
No, tasks are the unit of work done per partition. Slots help Spark parallelize work. An executor can have multiple slots which enable it to process multiple tasks in parallel.

NEW QUESTION # 157
Which of the following describes characteristics of the Dataset API?

- A. In Python, the Dataset API mainly resembles Pandas' DataFrame API.
- B. The Dataset API does not provide compile-time type safety.
- C. The Dataset API is available in Scala, but it is not available in Python.
- D. The Dataset API does not support unstructured data.
- E. In Python, the Dataset API's schema is constructed via type hints.

Answer: C

Explanation:
Explanation
The Dataset API is available in Scala, but it is not available in Python.
Correct. The Dataset API uses fixed typing and is typically used for object-oriented programming. It is available when Spark is used with the Scala programming language, but not for Python. In Python, you use the DataFrame API, which is based on the Dataset API.
The Dataset API does not provide compile-time type safety.
No - in fact, depending on the use case, the type safety that the Dataset API provides is an advantage.
The Dataset API does not support unstructured data.
Wrong, the Dataset API supports structured and unstructured data.
In Python, the Dataset API's schema is constructed via type hints.

No, this is not applicable since the Dataset API is not available in Python.
In Python, the Dataset API mainly resembles Pandas' DataFrame API.
The Dataset API does not exist in Python, only in Scala and Java.

NEW QUESTION # 158
.....