

SPS-C01 Latest Exam & Top SPS-C01 Exam Dumps



Snowflake SPS-C01

SnowPro Specialty: Snowpark Certification Exam

Questions & Answers PDF
(Demo Version Limited Content)

For More Information Visit link below:

<https://p2pexam.com/>

Visit us at: <https://p2pexam.com/spc-c01>

What's more, part of that Exam-Killer SPS-C01 dumps now are free: <https://drive.google.com/open?id=1EycuP8sY7nuK1V8DSQknIFQUhGNLxAYe>

With the pass rate reaching 98.65%, our SPS-C01 training materials have gained popularity in the international market. If you choose us, we can ensure that you can pass the exam in your first attempt. We are pass guarantee and money back guarantee for SPS-C01 exam dumps. If you fail to pass the exam, we will give you refund. You can try free demo before buying SPS-C01 Exam Materials, so that you can have deeper understanding of what you are going to buy. Free update for one year is available, the update version for SPS-C01 exam braindumps will be sent to your email automatically.

Preparing authentic Snowflake SPS-C01 questions in the form of a PDF file is significant because it is the only choice that guarantees your success in the SPS-C01 exam. Snowflake SPS-C01 PDF questions are accessible without any installation. You will need a few days to prepare successfully for the SPS-C01 Exam if you have Exam-Killer's Snowflake Exam PDF Questions. This PDF file of Snowflake SPS-C01 questions is supported by any device like laptops, tablets, and smartphones.

>> SPS-C01 Latest Exam <<

Top Snowflake SPS-C01 Exam Dumps - SPS-C01 Valid Exam Sample

There are three different versions of our SPS-C01 study guide which are PDF, Software and APP online versions. For their varied advantages, our SPS-C01 learning questions have covered almost all the interests and habits of varied customers groups. No matter you are a student, a working staff, or even a house wife, you will find the exact version of your SPS-C01 Exam Materials to offer you a pleasant study experience.

Snowflake Certified SnowPro Specialty - Snowpark Sample Questions (Q344-

Q349):

NEW QUESTION # 344

You have created a Snowpark UDF that uses a custom Python module 'my_module.py', containing a function 'process data'. This module is not available through Anaconda. You've packaged the module into a zip file named 'my module.zip'. What steps are necessary to deploy this UDF in Snowflake so that it can correctly use the 'my_module'?

- A. Upload 'my_module.zip' to an internal stage, then create the UDF using 'session.add_import' within the UDF definition, specifying the stage path. No additional configuration is needed.
- **B. Upload 'my_module.zip' to an internal stage. When creating the UDF, specify the stage path in the 'imports' argument. No changes to sys.path are required within the UDF.**
- C. Upload 'my_module.zip' to an internal stage. When creating the UDF, specify the stage path in the 'imports' argument. Within the UDF, modify 'sys.path' to include the path where Snowflake unpacks the zip file.
- D. Upload 'my_module.zip' to an internal stage. When creating the UDF, specify the stage path in the 'packages' argument. Within the UDF, modify 'sys.path' to include the path where Snowflake unpacks the zip file.
- E. Upload 'my_module.zip' to an external stage (e.g., AWS S3 or Azure Blob Storage). Configure Snowflake to access the external stage. Create the UDF, specifying the external stage path in the 'imports' argument.

Answer: B

Explanation:

Option B is correct. You need to upload the zip file to an internal stage. The 'imports' argument during UDF creation tells Snowflake to unpack the zip and make the module available. Snowflake handles adding the necessary path to 'sys.path', so no manual modification is needed within the UDF. Specifying external stages is possible, but internal stages are preferred for security and performance. Option A is incomplete (doesn't specify how the import relates to the UDF), Option C is technically feasible with external stages but is less preferable. Option D is incorrect as it unnecessary to modify sys.path. Option E specifies 'packages' argument where you would normally include packages available through Anaconda.

NEW QUESTION # 345

Consider a scenario where you have a table 'EMPLOYEES' with columns 'employee id', 'department', and 'salary'. You want to delete employees who belong to either the 'HR' or 'Finance' department and have a salary less than 60000. Which of the following Snowpark DataFrame operations correctly implements this deletion?

- A. Option D
- **B. Option E**
- C. Option B
- D. Option C
- E. Option A

Answer: B

Explanation:

Option E is the correct solution because it uses the 'delete' function with the correct boolean logic: `== 'HR') I (col('department') 'Finance')) & (col('salary') < 60000)`. This accurately translates to 'department is HR OR department is Finance AND salary is less than 60000'. Option A has incorrect syntax for 'delete()' usage. Option B is missing parenthesis for correct precedence. Option C has incorrect precedence; the 'and' will bind tighter than the or. Option D filters twice but does not correctly use the .delete' method with filter conditions.

NEW QUESTION # 346

Consider the following Snowpark Python code snippet designed to calculate the moving average of sales data'. You've identified that the code is performing poorly and suspect the window function is a bottleneck. How can you optimize this code for better performance?

- A. Ensure that the data is pre-sorted according to the ordering specified in the window function before creating the Snowpark DataFrame.
- **B. Rewrite the window function logic as a series of aggregation queries to improve performance on very large datasets.**
- **C. Use the method on the Snowpark DataFrame before applying the window function to avoid re-reading the data multiple times.**

- D. Replace the 'Window.orderBy' with 'Window.partitionBy' on a highly cardinal column to distribute the window calculations across multiple nodes.
- E. Explicitly specify a range-based window frame (e.g., 'rowsBetween') instead of a rows-based window frame (e.g., 'rangeBetween') if appropriate for the calculation.

Answer: B,C

Explanation:

Caching the DataFrame allows reuse of the data and avoids recomputation, improving performance. Rewriting the logic with aggregation queries is a viable optimization. Partitioning by a cardinal column does not improve performance. Presorting the data before creating the DataFrame does not affect Window function performance. Range-based windows are not always a direct replacement and have specific use cases.

NEW QUESTION # 347

You are working with a Snowpark DataFrame 'products df' containing product information, including 'product_id', 'price', and 'discount'. You need to update the 'price' column in the 'products' table based on the following logic: If 'discount' is greater than 0.2, reduce the 'price' by 15%. If 'discount' is between 0.1 and 0.2 (inclusive), reduce the 'price' by 5%. Otherwise, keep the 'price' as is. Which of the following Snowpark code snippets efficiently implements this update? Assume 'products' table already exists and is correctly populated.

- A. Option D
- **B. Option E**
- C. Option B
- D. Option C
- E. Option A

Answer: B

Explanation:

Option E is the most concise and correct solution. It uses 'with_column' to directly update the 'price' column based on the discount conditions, using nested 'when' functions for the logic and persists the change. Option A, while technically correct, is less efficient because it creates a new column ('new_price'), drops the original 'price' column, and then renames the new column. Option B tries to use an 'update' method which doesn't exist directly on Snowpark DataFrames in that way. Option C works correctly. Option D has an issue that it won't keep the original schema of the table being updated as it is selecting each of the columns.

NEW QUESTION # 348

You are developing a Snowpark application that performs feature engineering on a dataset of customer transactions. This involves calculating several complex aggregate features such as rolling averages, medians, and custom ratios. You want to optimize the performance of this feature engineering process using a Snowpark-optimized warehouse. Which of the following strategies would be MOST effective in achieving optimal performance?

- **A. Use the 'GROUP BY' clause in Snowpark SQL to compute aggregate features, leveraging window functions where appropriate for rolling calculations.**
- B. Implement all feature engineering calculations using Python User-Defined Functions (UDFs) and apply them to the Snowpark DataFrame.
- C. Materialize intermediate Snowpark DataFrames after each feature engineering step to avoid recomputation.
- D. Use stored procedures implemented in Java within Snowpark for feature calculations.
- **E. Leverage Snowpark's built-in functions and SQL expressions as much as possible for feature engineering, and rewrite performance-critical calculations as Java or Scala User-Defined Table Functions (UDTFs).**

Answer: A,E

Explanation:

Using 'GROUP BY' and window functions allows Snowflake to optimize the calculations within its engine. Leveraging UDTFs allows custom computations while still benefiting from Snowflake's optimization capabilities. Python UDFs are generally slower than equivalent SQL or Java/Scala UDTFs due to inter-process communication overhead. Materializing intermediate DataFrames can help in some scenarios but can also introduce overhead if not managed carefully. Java Stored procedures could be used, but UDTF would be more optimized way.

• • • • •

Top SPS-C01 Exam Dumps: <https://www.exam-killer.com/SPS-C01-valid-questions.html>

writer to Video Systems and Studio Monthly SPS-C01 Valid Exam Sample magazines, Defining New Schema Elements, You can print it out to take with you anywhere, or simply open it on any device that SPS-C01 supports PDF files (you may need to install a PDF reader if you don't have one).

Take it easy, our specialists have given a vivid explanation to the difficult SPS-C01 Valid Exam Sample point, While, maybe some people have the foresight and prepare in advance, but still not find a good and proper method to study and prepare it well.

[illegible]

BTW, DOWNLOAD part of Exam-Killer SPS-C01 dumps from Cloud Storage: <https://drive.google.com/open?id=1EycuP8sY7nuK1V8DSOknFOUhGNLxAYe>

