Providing You Pass-Sure New SOL-C01 Test Notes with 100% Passing Guarantee

Our SOL-C01 learning guide materials have won the favor of many customers by virtue of their high quality. Started when the user needs to pass the qualification test, choose the SOL-C01 real questions, they will not have any second or even third backup options, because they will be the first choice of our practice exam materials. Our SOL-C01 Practice Guide is devoted to research on which methods are used to enable users to pass the test faster. Therefore, through our unremitting efforts, our SOL-C01 real questions have a pass rate of 98% to 100%.

If you prefer to practice SOL-C01 exam dumps on paper, then our exam dumps is your best choice. SOL-C01 PDF version is printable, and you can print them into hard one if you like, and you can also take some notes on them and practice them anytime and anyplace. Moreover, SOL-C01 training materials cover most of knowledge points for the exam, and you can have a good command of the major knowledge points as well as improve your professional ability in the process of practicing. We offer you free update for 365 days for SOL-C01 Exam Materials after purchasing. Our system will send the update version to you automatically.

>> New SOL-C01 Test Notes <<

SOL-C01 Vce Test Simulator | Valid Exam SOL-C01 Blueprint

The product is made in three different formats to help customers with different preparation styles meet their needs. One of these formats is Snowflake SOL-C01 Dumps PDF file which is printable and portable. Users can take Snowflake Certified SnowPro Associate - Platform Certification (SOL-C01) PDF questions anywhere and use them anytime.

Snowflake SOL-C01 Exam Syllabus Topics:

Topic	Details
Topic 1	• Identity and Data Access Management: This domain focuses on Role-Based Access Control (RBAC) including role hierarchies and privileges, along with basic database administration tasks like creating objects, transferring ownership, and executing fundamental SQL commands.
Topic 2	• Interacting with Snowflake and the Architecture: This domain covers Snowflake's elastic architecture, key user interfaces like Snowsight and Notebooks, and the object hierarchy including databases, schemas, tables, and views with practical navigation and code execution skills.
Topic 3	• Data Loading and Virtual Warehouses: This domain covers loading structured, semi-structured, and unstructured data using stages and various methods, virtual warehouse configurations and scaling strategies, and Snowflake Cortex LLM functions for AI-powered operations.
Topic 4	• Data Protection and Data Sharing: This domain addresses continuous data protection through Time Travel and cloning, plus data collaboration capabilities via Snowflake Marketplace and private Data Exchange sharing.

Snowflake Certified SnowPro Associate - Platform Certification Sample Questions (Q160-Q165):

NEW QUESTION # 160
You are tasked with securing sensitive data in a table called `CUSTOMER DATA'. You want to grant the 'SELECT privilege on this table to a role called 'ANALYST ROLE, but prevent them from seeing the 'CREDIT CARD NUMBER column, which contains personally identifiable information (PII). Which of the following approaches are valid in Snowflake to achieve this?

- A. Use dynamic data masking on the column and grant the 'UNMASK' privilege to only specific roles other than 'ANALYST_ROLE'.
- B. Create a view on 'CUSTOMER_DATA' that excludes the column and grant 'SELECT' on the view to 'ANALYST_ROLE.
- C. Apply a row access policy to filter out rows containing sensitive information when accessed by 'ANALYST_ROLE.
- D. Grant `SELECT on 'CUSTOMER_DATA' to 'ANALYST_ROLE and then 'REVOKE SELECT specifically on the column.
- E. Create a secure view on 'CUSTOMER DATA that excludes the 'CREDIT CARD NUMBER column and grant 'SELECT on the secure view to 'ANALYST ROLE.

Answer: A,B,E

Explanation:
Option A is valid: Creating a view and granting 'SELECT on the view allows you to control which columns are visible to the 'ANALYST ROLE. Option B is valid: Dynamic data masking allows you to mask the sensitive column for unauthorized users (in this case, 'ANALYST_ROLE') while allowing authorized users to see the unmasked data using the 'UNMASK' privilege. Option C is invalid: You cannot revoke privileges on a specific column; privileges are granted and revoked at the table level. Option D is incorrect: Row Access Policies filter rows, not columns. Option E is valid: Secure views are the most secure way to filter the specific data.

NEW QUESTION # 161
You are using a Snowflake Notebook to perform data analysis on a large dataset. As part of your analysis, you need to create a custom Python function that calculates a complex metric based on multiple columns in a Snowflake table.
You want to apply this function to each row of the table and store the results in a new column.
Which of the following approaches is the MOST efficient and scalable way to achieve this using Snowflake and Python?

- A. Use the '%%osql' magic command to execute a series of SQL UPDATE' statements that call the Python function using a IJDF.

- B. Iterate over the rows of the Snowflake table using the Snowflake Connector for Python, call the Python function for each row, and then use "INSERT statements to insert the calculated values into a new table.
- C. Create a Snowflake Python User-Defined Function (UDF) that encapsulates the calculation logic and then use it in a 'SELECT statement to create a new column with the calculated values. Store the result in a new table using 'CREATE TABLE AS SELECT
- D. Load the entire Snowflake table into a Pandas DataFrame, apply the Python function to each row using 'DataFrame.apply()', and then write the modified DataFrame back to Snowflake.
- E. Create a stored procedure in Snowflake that runs the logic in a separate environment.

Answer: C

Explanation:
Option C, creating a Snowflake Python IJDF and using it in a `SELECT statement within a `CREATE TABLE AS SELECT statement, is the most efficient and scalable approach. Snowflake IJDFs allow you to execute Python code directly within the Snowflake engine, leveraging Snowflake's distributed processing capabilities. This avoids the overhead of transferring large amounts of data between Snowflake and the Python environment in the Notebook. Loading the entire table into a Pandas DataFrame (A) is not scalable for large datasets and can lead to memory issues. Using `%%osql' with `UPDATE statements (B) would be very slow due to the row-by-row updates. Iterating over rows using the Snowflake Connector (D) is also inefficient and not scalable. Option E is incorrect because it doesn't directly use Python code from the Notebook.

NEW QUESTION # 162
What is the maximum duration of data retention using Time Travel in Snowflake for standard accounts?

- A. 0
- B. 1
- C. 2
- D. 3

Answer: C

Explanation:
For Snowflake Standard Edition accounts, the maximum Time Travel retention period is1 day(24 hours).
Time Travel allows users to query historical data, restore dropped objects, and clone objects at a point in time.
Although the retention period can be adjusted at the account, database, schema, or table level, Standard Edition restricts the upper limit to 1 day. A value of 0 disables Time Travel entirely. A 7-day retention period applies to Fail-safe, not Time Travel, and Fail-safe is a disaster-recovery mechanism managed exclusively by Snowflake. The 90-day Time Travel retention is available only for Enterprise Edition and above. Therefore, the correct maximum retention period for Standard Edition is 1 day.

NEW QUESTION # 163
You are exploring a large dataset in Snowflake with millions of rows. You want to get a quick overview of the data types and basic statistics for each column in a table named 'ORDERS in the
'SALES SCHEMA' schema of the 'RETAIL DB' database. Which of the following methods provides the most efficient and comprehensive overview of the data without requiring you to write complex SQL queries?

- A. Use the DESCRIBE TABLE command to display the column definitions.
- B. Use the Snowflake web interface to navigate to the table and view the 'Columns' tab.
- C. Use Snowflake's data profiling features within Snowsight, selecting the 'Orders' table and viewing the automatically generated statistics.
- D. Use the `SELECT FROM LIMIT 10;' command followed by manually inspecting the data types.
- E. Use the `SELECT column_name, data_type FROM INFORMATION SCHEMA.COLUMNS WHERE table_name = 'ORDERS' AND table_schema = 'SALES_SCHEMA' AND table_catalog = 'RETAIL_DB';' command.

Answer: C

Explanation:
Snowsight's data profiling feature automatically provides data types, statistics (min, max, average, etc.), and histograms without requiring any SQL. 'DESCRIBE TABLE only gives column definitions, not statistics. Viewing the 'Columns' tab in the web interface gives column definitions.

'SELECT LIMIT Iff only shows sample data, not data types of all columns. Using the
'INFORMATION_SCHEMR provides only the data types but not any statistics. Snowflake's data profiling within Snowsight gives
both. So, 'E' is the correct answer.

NEW QUESTION # 164
A data engineer needs to grant SELECT privileges on all tables within a newly created schema named 'SALES DATA' to a role
called 'ANALYST ROLE'. However, new tables may be added to this schema in the future. What is the most efficient and secure
way to achieve this, ensuring that future tables automatically inherit the necessary permissions?

- A. Create a custom role that inherits from 'ANALYST ROLE and grant 'SELECT privilege on all tables to this new role.
- B. Grant the USAGE privilege on the database containing the schema and SELECT privilege on
  `FUTURE GRANTS for tables in 'SALES DATA' to the 'ANALYST ROLE using 'GRANT USAGE ON DATABASE TO
  ROLE ANALYST ROLE; GRANT SELECT ON FUTURE TABLES IN SCHEMA SALES DATA TO ROLE
  ANALYST ROLE;'
- C. Use a stored procedure to automatically grant the SELECT privilege to 'ANALYST ROLE whenever a new table is
  created in the 'SALES DATA' schema.
- D. Grant 'SELECT privilege to the PUBLIC role on all tables within the 'SALES_DATX schema.
- E. Grant SELECT privilege to `ANALYST ROLE directly on each table using `GRANT SELECT ON TABLE ... TO
  ROLE ANALYST ROLE;' after table creation.

Answer: B

Explanation:
Using GRANT SELECT ON FUTURE TABLES' ensures that any new tables created in the schema automatically inherit the
SELECT privilege for the specified role. Option A requires manual intervention for each new table. Options C and E are
unnecessarily complex. Option D poses a security risk by granting privileges to the PUBLIC role.

NEW QUESTION # 165
......

Laziness will ruin your life one day. It is time to have a change now. Although we all love cozy life, we must work hard to create our
own value. Then our SOL-C01 training materials will help you overcome your laziness. Study is the best way to enrich your life. On
one hand, you may learn the newest technologies in the field with our SOL-C01 Study Guide to help you better adapt to your work,
and on the other hand, you will pass the SOL-C01 exam and achieve the certification which is the symbol of competence.

- Free PDF Snowflake - SOL-C01 –Professional New Test Notes ☐ Download （SOL-C01） for free by simply entering ➟ www.pass4test.com ☐ website ☐SOL-C01 Exam Brain Dumps
- www.slideshare.net, www.stes.tyc.edu.tw, cocoasr18.blogspot.com, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, Disposable vapes