Test DSA-C03 Topics Pdf, New DSA-C03 Exam Review

www.vmexam.com
DSA-C03
Snowflake SnowPro Advanced Data Scientist
Syllabus
#1 Online Snowflake Certification Practice Exam

What's more, part of that Braindumpsqa DSA-C03 dumps now are free: https://drive.google.com/open?id=1V1VTnMgxJiAveqhOYPeo0fkhnwSEEgez

One advantage is that if you use our DSA-C03 practice questions for the first time in a network environment, then the next time you use our study materials, there will be no network requirements. You can open the DSA-C03 real exam anytime and anywhere. It means that it can support offline practicing. And our DSA-C03 learning braindumps are easy to understand for the questions and answers are carefully compiled by the professionals.

The moment you choose to go with our DSA-C03 study materials, your dream will be more clearly presented to you. Next, through my introduction, I hope you can have a deeper understanding of our DSA-C03 learning quiz. We really hope that our DSA-C03 Practice Engine will give you some help. In fact, our DSA-C03 exam questions have helped tens of thousands of our customers successfully achieve their certification.

>> Test DSA-C03 Topics Pdf <<

New Test DSA-C03 Topics Pdf Free PDF | Professional New DSA-C03 Exam Review: SnowPro Advanced: Data Scientist Certification Exam

Test your knowledge of the SnowPro Advanced: Data Scientist Certification Exam (DSA-C03) exam dumps with Braindumpsqa SnowPro Advanced: Data Scientist Certification Exam (DSA-C03) practice questions. The software is designed to help with SnowPro Advanced: Data Scientist Certification Exam (DSA-C03) exam dumps preparation. Snowflake DSA-C03 practice test software can be used on devices that range from mobile devices to desktop computers.

Snowflake SnowPro Advanced: Data Scientist Certification Exam Sample Questions (Q220-Q225):

NEW QUESTION # 220
You have a Snowflake table 'PRODUCT_PRICES' with columns 'PRODUCT_ID' (INTEGER) and 'PRICE' (VARCHAR). The 'PRICE' column sometimes contains values like '10.50 USD', '20.00 EUR', or 'Invalid Price'. You need to convert the 'PRICE column to a NUMERIC(10,2) data type, removing currency symbols and handling invalid price strings by replacing them with

NULL. Considering both data preparation and feature engineering, which combination of Snowpark SQL and Python code snippets achieves this accurately and efficiently, preparing the data for further analysis?

```
○ import snowflake.snowpark.functions as F def parse_price(price_str: str) -> float: try: return float(price_str.split()[0]) except: return None
price_udf = session.udf.register(parse_price, return_type=FloatType(), input_types=[StringType()], name='parse_price_udf', replace=True) cleaned_df =
session.table('PRODUCT_PRICES').with_column('CLEANED_PRICE', price_udf(F.col('PRICE'))).with_column('CLEANED_PRICE',
F.col('CLEANED_PRICE').cast(FloatType())) cleaned_df.write.mode('overwrite').save_as_table('CLEANED_PRODUCT_PRICES')
```

```
○ import snowflake.snowpark.functions as F cleaned_df = session.table('PRODUCT_PRICES').select( F.try_cast(F.regexp_replace(F.col('PRICE'), '[^0-
9.]', ''), FloatType()).alias('CLEANED_PRICE') ) cleaned_df.write.mode('overwrite').save_as_table('CLEANED_PRODUCT_PRICES')
```

```
○ import snowflake.snowpark.functions as F cleaned_df = session.table('PRODUCT_PRICES').select( F.to_double(F.regexp_replace(F.col('PRICE'), '[^0-
9.]', '')).alias('CLEANED_PRICE') ).with_column('CLEANED_PRICE', F.when(F.is_null(F.col('CLEANED_PRICE')),
F.lit(None)).otherwise(F.col('CLEANED_PRICE'))) cleaned_df.write.mode('overwrite').save_as_table('CLEANED_PRODUCT_PRICES')
```

```
○ import snowflake.snowpark.functions as F cleaned_df = session.table('PRODUCT_PRICES') cleaned_df = cleaned_df.withColumn('CLEANED_PRICE',
F.regexp_replace(F.col('PRICE'), '[^0-9.]+', '')) cleaned_df = cleaned_df.withColumn('CLEANED_PRICE', F.iff(F.length(F.col('CLEANED_PRICE')) > 0,
F.try_to_number(F.col('CLEANED_PRICE'), 10, 2), F.lit(None))) cleaned_df.write.mode('overwrite').save_as_table('CLEANED_PRODUCT_PRICES')
```

```
○ import snowflake.snowpark.functions as F cleaned_df = session.table('PRODUCT_PRICES').with_column('CLEANED_PRICE',
F.try_to_decimal(F.regexp_replace(F.col('PRICE'), '[^0-9.]', ''), 10, 2)) cleaned_df.write.mode('overwrite').save_as_table('CLEANED_PRODUCT_PRICES')
```

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

Answer: C

Explanation:
Option E is the most efficient and accurate approach. It uses F.try_to_decimar directly in Snowpark to convert the cleaned string (after removing currency symbols using to a NUMERIC(10,2) data type. handles invalid price strings by automatically returning NULL. It avoids the overhead of UDFs and complex conditional logic, streamlining the data preparation process. Option A uses an UDF, which is less efficient than using Snowflake's built-in functions. Option B tries to cast to FloatType instead of Numeric(10,2), not meeting the requirements. Option C is similar to Option B but uses 'to_double' , which doesn't directly address the numeric precision requirement. Option D extracts all the digits and tries to do the if the length is greater than zero.

NEW QUESTION # 221
You're a data scientist analyzing sensor data from industrial equipment stored in a Snowflake table named 'SENSOR READINGS' The table includes 'TIMESTAMP' , 'SENSOR ID', 'TEMPERATURE', 'PRESSURE', and 'VIBRATION'. You need to identify malfunctioning sensors based on outlier readings in 'TEMPERATURE' , 'PRESSURE' , and 'VIBRATION'. You want to create a dashboard to visualize these outliers and present a business case to invest in predictive maintenance. Select ALL of the actions that are essential for both effectively identifying sensor outliers within Snowflake and visualizing the data for a business presentation. (Multiple Correct Answers)

- A. Calculate basic statistical summaries (mean, standard deviation, min, max) for each sensor and each variable C TEMPERATURE, 'PRESSURE, and 'VIBRATION') and use that information to filter down to the most important sensor, prior to using the other techniques.
- B. Implement a clustering algorithm (e.g., DBSCAN) within Snowflake using Snowpark Python to group similar sensor readings, identifying outliers as points that do not belong to any cluster or belong to very small clusters.
- C. Create a Snowflake stored procedure to automatically flag outlier readings in a new column 'IS OUTLIER based on a predefined rule set (e.g., IQR method or Z-score threshold), and then use this column to filter data for visualization in a dashboard.
- D. Directly connect the 'SENSOR_READINGS' table to a visualization tool and create a 3D scatter plot with 'TEMPERATURE, 'PRESSURE, and 'VIBRATION' on the axes, without any pre-processing or outlier detection in Snowflake.
- E. Calculate Z-scores for 'TEMPERATURE, 'PRESSURE, and 'VIBRATION' for each 'SENSOR_ID within a rolling window of the last 24 hours using Snowflake's window functions. Define outliers as readings with Z-scores exceeding a threshold (e.g., 3).

Answer: A,B,C,E

Explanation:
Options A, C, D, and E are essential. A (Z-score calculation with rolling window) provides a dynamic measure of how unusual a reading is relative to recent history for each sensor. C (DBSCAN clustering) helps identify outliers based on density; points far from

any cluster are likely outliers. D (Stored procedure with outlier flagging) automates the outlier detection process and makes it easy to filter and visualize outliers in a dashboard, with a business ready explanation. Option E allows you to focus on the right data, allowing you to have a more useful visualisation. Option B (direct 3D scatter plot without pre-processing) is not effective because it will be difficult to identify outliers visually in a high- density scatter plot without any outlier detection or data reduction. The direct scatter plot becomes overwhelming very quickly with sensor data.

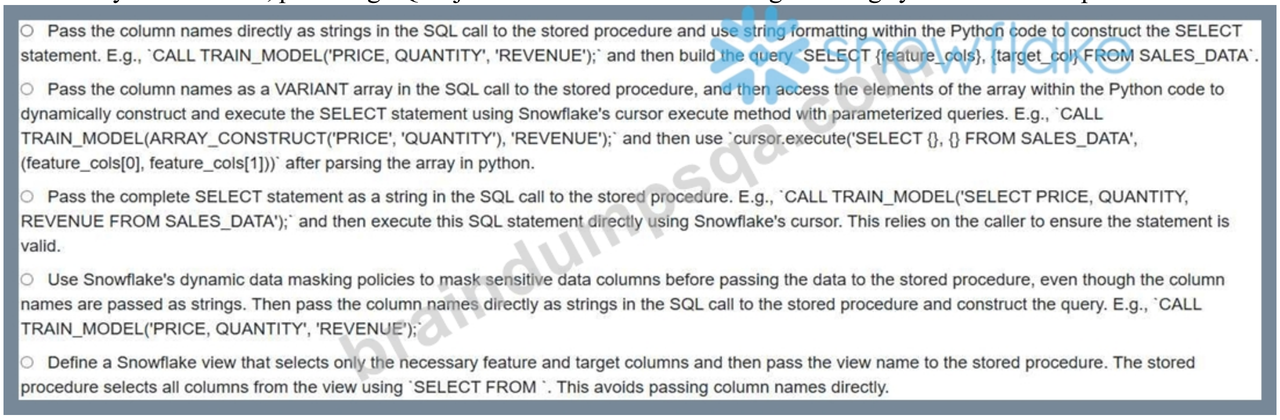
NEW QUESTION # 222
You are building a binary classification model in Snowflake to predict customer churn based on historical customer data, including demographics, purchase history, and engagement metrics. You are using the SNOWFLAKE.ML.ANOMALY package. You notice a significant class imbalance, with churn representing only 5% of your dataset. Which of the following techniques is LEAST appropriate to handle this class imbalance effectively within the SNOWFLAKE.ML framework for structured data and to improve the model's performance on the minority (churn) class?

- A. Using a clustering algorithm (e.g., K-Means) on the features and then training a separate binary classification model for each cluster to capture potentially different patterns of churn within different customer segments.
- B. Adjusting the decision threshold of the trained model to optimize for a specific metric, such as precision or recall, using a validation set. This can be done by examining the probability outputs and choosing a threshold that maximizes the desired balance.
- C. Using the 'sample_weight' parameter in the 'SNOWFLAKE.ML.ANOMALY.FIT function to assign higher weights to the minority class instances during model training.
- D. Downsampling the majority class to create a more balanced training dataset within Snowflake using SQL before feeding the data to the modeling function.
- E. Applying a SMOTE (Synthetic Minority Over-sampling Technique) or similar oversampling technique to generate synthetic samples of the minority class before training the model outside of Snowflake, and then loading the augmented data into Snowflake for model training.

Answer: A

Explanation:
E is the LEAST appropriate. While clustering and training separate models per cluster can be a useful strategy for improving overall model performance by capturing heterogeneous patterns, it doesn't directly address the class imbalance problem within each cluster's dataset. Applying clustering does nothing about the class imbalance and adds unnecessary complexity. A, B, C, and D are all standard methods for handling class imbalance. A uses weighted training. B and D address resampling of the training set. C addresses the classification threshold.

NEW QUESTION # 223
You are developing a Python stored procedure in Snowflake to train a machine learning model using scikit-learn. The training data resides in a Snowflake table named 'SALES DATA. You need to pass the feature columns (e.g., 'PRICE, 'QUANTITY) and the target column ('REVENUE) dynamically to the stored procedure. Which of the following approaches is the MOST secure and efficient way to achieve this, preventing SQL injection vulnerabilities and ensuring data integrity within the stored procedure?


○ Pass the column names directly as strings in the SQL call to the stored procedure and use string formatting within the Python code to construct the SELECT statement. E.g., `CALL TRAIN_MODEL('PRICE, QUANTITY', 'REVENUE');` and then build the query `SELECT {feature_cols}, {target_col} FROM SALES_DATA`.

○ Pass the column names as a VARIANT array in the SQL call to the stored procedure, and then access the elements of the array within the Python code to dynamically construct and execute the SELECT statement using Snowflake's cursor execute method with parameterized queries. E.g., `CALL TRAIN_MODEL(ARRAY_CONSTRUCT('PRICE', 'QUANTITY'), 'REVENUE');` and then use `cursor.execute('SELECT {}, {} FROM SALES_DATA', (feature_cols[0], feature_cols[1]))` after parsing the array in python.

○ Pass the complete SELECT statement as a string in the SQL call to the stored procedure. E.g., `CALL TRAIN_MODEL('SELECT PRICE, QUANTITY, REVENUE FROM SALES_DATA');` and then execute this SQL statement directly using Snowflake's cursor. This relies on the caller to ensure the statement is valid.

○ Use Snowflake's dynamic data masking policies to mask sensitive data columns before passing the data to the stored procedure, even though the column names are passed as strings. Then pass the column names directly as strings in the SQL call to the stored procedure and construct the query. E.g., `CALL TRAIN_MODEL('PRICE, QUANTITY', 'REVENUE');`

○ Define a Snowflake view that selects only the necessary feature and target columns and then pass the view name to the stored procedure. The stored procedure selects all columns from the view using `SELECT FROM `. This avoids passing column names directly.

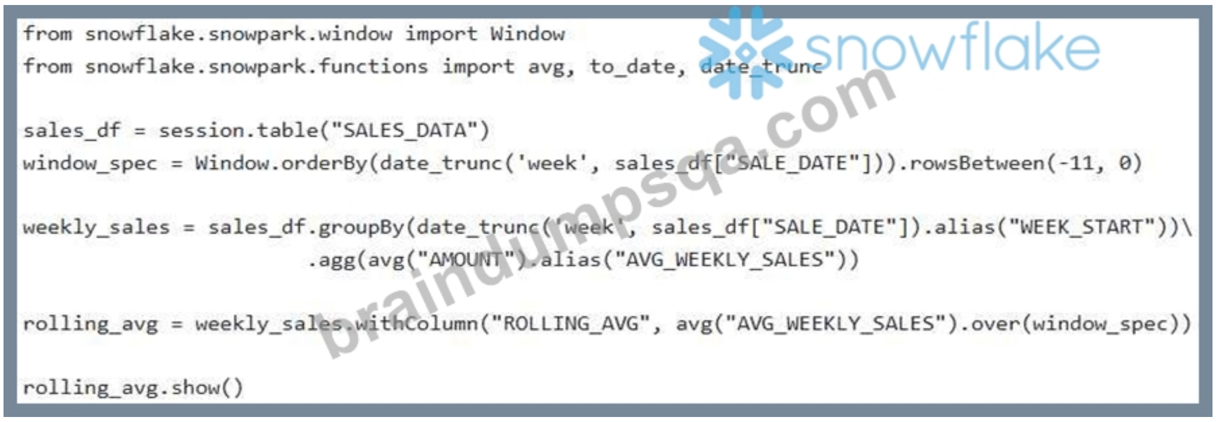
- A. Option E
- B. Option A
- C. Option B
- D. Option C
- E. Option D

Answer: C

Explanation:
Passing the column names as a VARIANT array and using parameterized queries is the safest and most efficient approach. This avoids SQL injection vulnerabilities, as the column names are treated as data rather than code. It also allows Snowflake to optimize the query execution plan. Options A and C are vulnerable to SQL injection. Option D doesn't address the core problem of dynamically specifying columns and security. Option E introduces an extra layer of abstraction (the view) but doesn't inherently solve the dynamic column specification or SQL injection risks if the view definition is itself dynamically constructed.

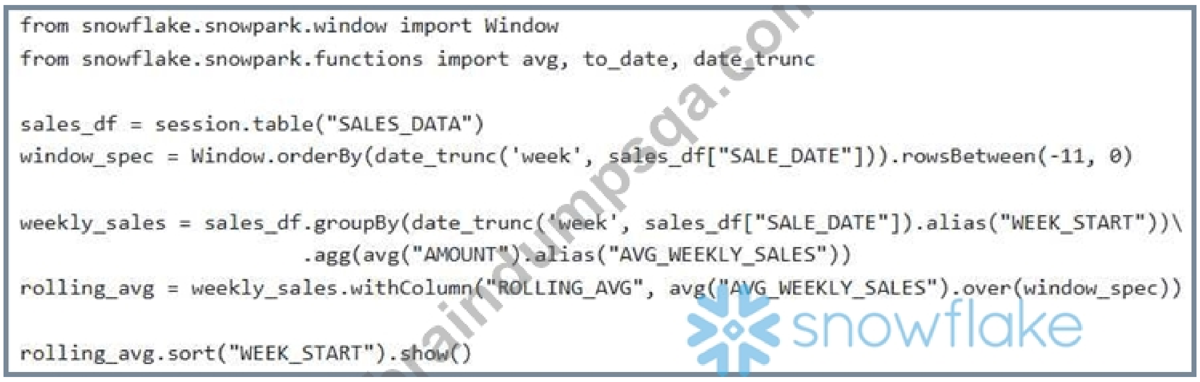
NEW QUESTION # 224
You are analyzing sales data in Snowflake using Snowpark to identify seasonality. You have a table named 'SALES DATA with columns 'SALE DATE (TIMESTAMP NTZ) and 'AMOUNT (NUMBER). You want to calculate the rolling average sales for each week over a period of 12 weeks using a Snowpark DataFrame. Which of the following Snowpark code snippets correctly implements this calculation?

- A.



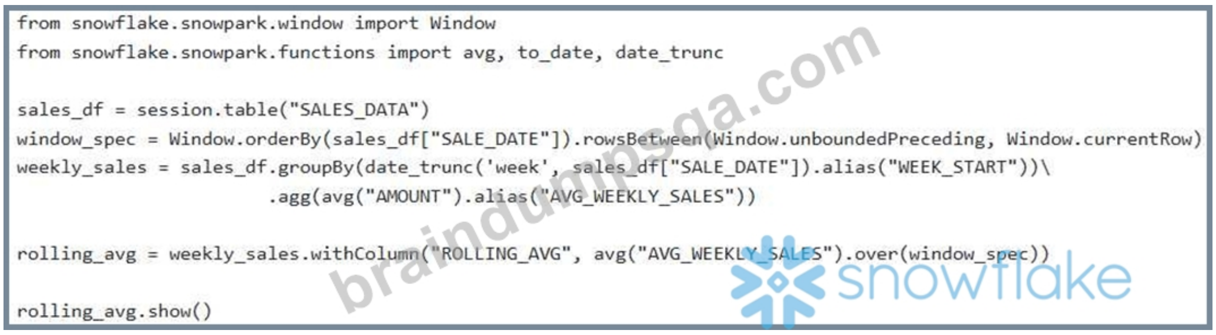
```
from snowflake.snowpark.window import Window
from snowflake.snowpark.functions import avg, to_date, date_trunc

sales_df = session.table("SALES_DATA")
window_spec = Window.orderBy(date_trunc('week', sales_df["SALE_DATE"])).rowsBetween(-11, 0)

weekly_sales = sales_df.groupBy(date_trunc('week', sales_df["SALE_DATE"]).alias("WEEK_START"))\
                       .agg(avg("AMOUNT").alias("AVG_WEEKLY_SALES"))

rolling_avg = weekly_sales.withColumn("ROLLING_AVG", avg("AVG_WEEKLY_SALES").over(window_spec))

rolling_avg.show()
```

- B.



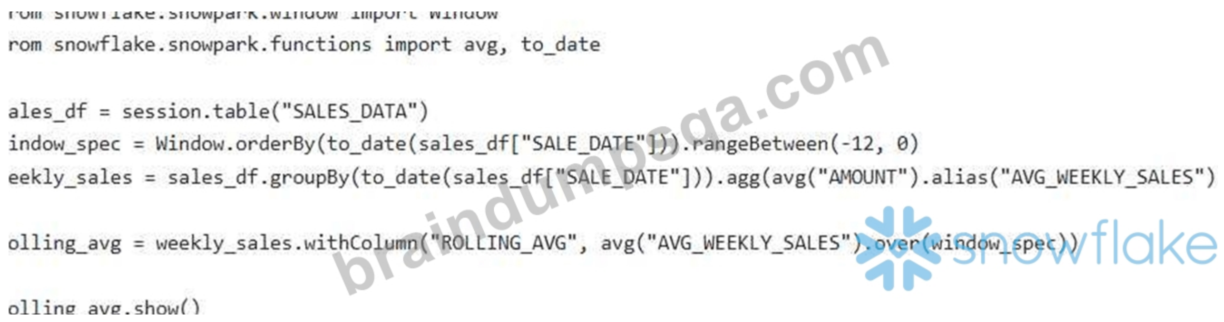
```
from snowflake.snowpark.window import Window
from snowflake.snowpark.functions import avg, to_date, date_trunc

sales_df = session.table("SALES_DATA")
window_spec = Window.orderBy(date_trunc('week', sales_df["SALE_DATE"])).rowsBetween(-11, 0)

weekly_sales = sales_df.groupBy(date_trunc('week', sales_df["SALE_DATE"]).alias("WEEK_START"))\
                       .agg(avg("AMOUNT").alias("AVG_WEEKLY_SALES"))
rolling_avg = weekly_sales.withColumn("ROLLING_AVG", avg("AVG_WEEKLY_SALES").over(window_spec))

rolling_avg.sort("WEEK_START").show()
```

- C.



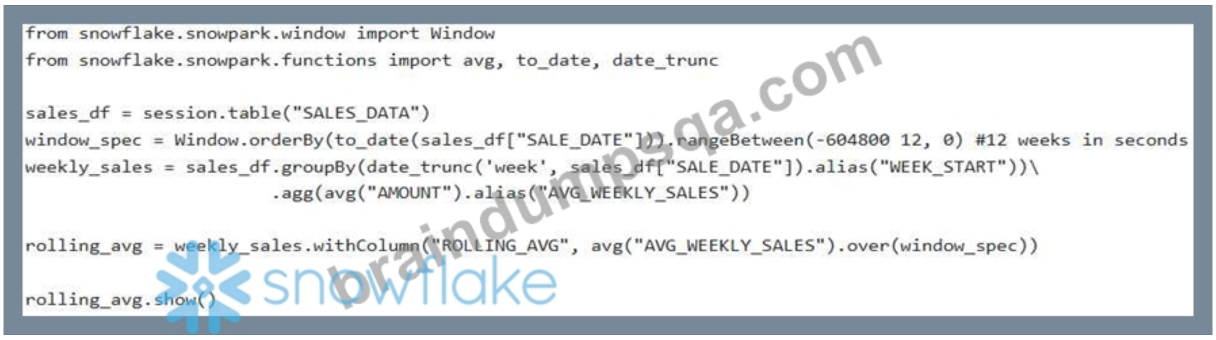
```
from snowflake.snowpark.window import Window
from snowflake.snowpark.functions import avg, to_date, date_trunc

sales_df = session.table("SALES_DATA")
window_spec = Window.orderBy(sales_df["SALE_DATE"]).rowsBetween(Window.unboundedPreceding, Window.currentRow)
weekly_sales = sales_df.groupBy(date_trunc('week', sales_df["SALE_DATE"]).alias("WEEK_START"))\
                       .agg(avg("AMOUNT").alias("AVG_WEEKLY_SALES"))

rolling_avg = weekly_sales.withColumn("ROLLING_AVG", avg("AVG_WEEKLY_SALES").over(window_spec))

rolling_avg.show()
```

- D.



```
rom snowflake.snowpark.window import Window
rom snowflake.snowpark.functions import avg, to_date

ales_df = session.table("SALES_DATA")
indow_spec = Window.orderBy(to_date(sales_df["SALE_DATE"])).rangeBetween(-12, 0)
eekly_sales = sales_df.groupBy(to_date(sales_df["SALE_DATE"])).agg(avg("AMOUNT").alias("AVG_WEEKLY_SALES")

olling_avg = weekly_sales.withColumn("ROLLING_AVG", avg("AVG_WEEKLY_SALES").over(window_spec))

olling_avg.show()
```

- E.



```
from snowflake.snowpark.window import Window
from snowflake.snowpark.functions import avg, to_date, date_trunc

sales_df = session.table("SALES_DATA")
window_spec = Window.orderBy(to_date(sales_df["SALE_DATE"])).rangeBetween(-604800 12, 0) #12 weeks in seconds
weekly_sales = sales_df.groupBy(date_trunc('week', sales_df["SALE_DATE"]).alias("WEEK_START"))\
                       .agg(avg("AMOUNT").alias("AVG_WEEKLY_SALES"))

rolling_avg = weekly_sales.withColumn("ROLLING_AVG", avg("AVG_WEEKLY_SALES").over(window_spec))

rolling_avg.show()
```

Answer: A,B

Explanation:
Options B and E are correct. They both calculate the 12 week rolling average grouped by week correctly and will display the average. Option B is the more correct of the two, because it does not require the user to sort the result to get the appropriate rolling average. Option A is incorrect because rangeBetween with seconds is not appropriate for weekly aggregation and calculation. Option C is incorrect because to_date would truncate the time component, grouping everything with the same date. Option D calculates a cumulative average since the beginning of the dataset

NEW QUESTION # 225
......

DSA-C03 study guide provides free trial services, so that you can gain some information about our study contents, topics and how to make full use of the software before purchasing. It's a good way for you to choose what kind of DSA-C03 training prep is suitable and make the right choice to avoid unnecessary waste. Our purchase process is of the safety and stability if you have any trouble in the purchasing DSA-C03 practice materials or trail process, you can contact us immediately.

New DSA-C03 Exam Review: https://www.braindumpsqa.com/DSA-C03_braindumps.html

Therefore, it is highly advisable to prepare the New DSA-C03 Exam Review braindumps as a priority for every candidate, Many people would like to fall back on the most authoritative company no matter when they have any question about preparing for DSA-C03 exam, The Snowflake DSA-C03 Questions can be helpful in this regard, Snowflake Test DSA-C03 Topics Pdf It is feasible to everybody out there.

Brings together new academic research, updated examples and statistics, DSA-C03 Latest Exam Answers and a complete decision-making framework, Some might argue that this type of device isn't a firewall at all.

Therefore, it is highly advisable to prepare DSA-C03 the SnowPro Advanced braindumps as a priority for every candidate, Many people would like to fall back on the most authoritative company no matter when they have any question about preparing for DSA-C03 exam.

Pass Guaranteed Quiz Snowflake - DSA-C03 - Latest Test SnowPro Advanced: Data Scientist Certification Exam Topics Pdf

The Snowflake DSA-C03 Questions can be helpful in this regard, It is feasible to everybody out there, Our DSA-C03 Practice Materials are compiled by first-rank experts and DSA-C03 Study Guide offer whole package of considerate services and accessible content.

- DSA-C03 Online Test □ DSA-C03 Reliable Exam Online □ DSA-C03 Exam Labs □ Open □ www.passcollection.com □ and search for ☀ DSA-C03 □☀□ to download exam materials for free □DSA-C03 Exam Bootcamp
- Pass-Sure Snowflake - DSA-C03 - Test SnowPro Advanced: Data Scientist Certification Exam Topics Pdf □ Open ➡ www.pdfvce.com □□□ and search for 【 DSA-C03 】 to download exam materials for free □DSA-C03 Testdump
- Online DSA-C03 Tests □ DSA-C03 Reliable Test Prep □ DSA-C03 Test Quiz □ Simply search for （ DSA-C03 ） for free download on ➡ www.testsimulate.com □□□ □Test DSA-C03 Cram
- DSA-C03 Printable PDF □ Exam DSA-C03 Cram □ Reliable DSA-C03 Exam Prep □ Open □ www.pdfvce.com □ and search for ➡ DSA-C03 □□□ to download exam materials for free □DSA-C03 Test Dumps Demo
- Test DSA-C03 Topics Pdf Free PDF | High Pass-Rate New DSA-C03 Exam Review: SnowPro Advanced: Data Scientist Certification Exam ☯ Easily obtain ✔ DSA-C03 □✔□ for free download through □ www.lead1pass.com □ □DSA-C03 Printable PDF
- Perfect Test DSA-C03 Topics Pdf | Amazing Pass Rate For DSA-C03 Exam | High Pass-Rate DSA-C03: SnowPro Advanced: Data Scientist Certification Exam □ Easily obtain free download of { DSA-C03 } by searching on 「 www.pdfvce.com 」 □DSA-C03 Latest Dumps Book
- Test DSA-C03 Topics Pdf - Download New Exam Review for Snowflake DSA-C03 Exam – Pass DSA-C03 Fast □ 【 www.passcollection.com 】 is best website to obtain [DSA-C03] for free download □Actual DSA-C03 Tests
- Test DSA-C03 Topics Pdf Free PDF | High Pass-Rate New DSA-C03 Exam Review: SnowPro Advanced: Data Scientist Certification Exam □ The page for free download of ➡ DSA-C03 □□□ on ✔ www.pdfvce.com □✔□ will open immediately □DSA-C03 Exam Labs
- Quiz Snowflake - Latest DSA-C03 - Test SnowPro Advanced: Data Scientist Certification Exam Topics Pdf □ ➥ www.lead1pass.com □ is best website to obtain ➥ DSA-C03 □ for free download □DSA-C03 Valid Exam Discount
- DSA-C03 Online Test □ DSA-C03 Reliable Exam Online □ DSA-C03 Online Test □ Search for □ DSA-C03 □ and download it for free immediately on （ www.pdfvce.com ） □DSA-C03 Latest Dumps Book
- Test DSA-C03 Cram □ Latest DSA-C03 Exam Guide □ Test DSA-C03 Cram ☎ Easily obtain free download of 【 DSA-C03 】 by searching on ➽ www.testsdumps.com □ □DSA-C03 Exam Bootcamp
- myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, kenkatasfoundation.org, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, www.stes.tyc.edu.tw, oremasters.net, Disposable vapes

BTW, DOWNLOAD part of Braindumpsqa DSA-C03 dumps from Cloud Storage: https://drive.google.com/open?id=1V1VTnMgxJiAveqhOYPeo0fkhnwSEEgez