DSA-C03인기자격증시험대비덤프문제 - DSA-C03최고품질덤프문제보기

DSA-C03는Snowflake의 인증시험입니다.DSA-C03인증시험을 패스하면Snowflake인증과 한 발작 더 내디딘 것입니다. 때문에DSA-C03시험의 인기는 날마다 더해갑니다.DSA-C03시험에 응시하는 분들도 날마다 더 많아지고 있습니다. 하지만DSA-C03시험의 통과 율은 아주 낮습니다.DSA-C03인증시험준비중인 여러분은 어떤 자료를 준비하였나요?

Snowflake DSA-C03 덤프는 pdf버전,테스트엔진버전, 온라인버전 세가지 버전의 파일로 되어있습니다. pdf버전은 반드시 구매하셔야 하고 테스트엔진버전과 온라인버전은 pdf버전 구매시 추가구매만 가능합니다. pdf버전은 인쇄 가능하기에 출퇴근길에서도 공부가능하고 테스트엔진버전은 pc에서 작동가능한 프로그램이고 온라인버전은 pc 외에 휴태폰에서도 작동가능합니다.

>> DSA-C03인기자격증 시험대비 덤프문제 <<

최신 DSA-C03인기자격증 시험대비 덤프문제 덤프는 SnowPro Advanced: Data Scientist Certification Exam시험문제의 모든 유형과 범위를 커버

인테넷에 검색하면 Snowflake DSA-C03시험덤프공부자료가 헤아릴수 없을 정도로 많이 검색됩니다. 그중에서

최신 SnowPro Advanced DSA-C03 무료샘플문제 (Q31-Q36):

질문 #31

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. 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).
- C. 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.
- D. 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.
- E. 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.

정답：A,B,D,E

설명：

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.

질문 #32

You are building a machine learning model to predict loan defaults. You have a dataset in Snowflake with the following features: 'income' (annual income in USD), 'loan_amount' (loan amount in USD), and 'credit_score' (FICO score). You need to normalize these features before training your model. The data has outliers in both 'income' and 'loan_amount', and 'credit_score' has a roughly normal distribution but you still want to standardize it to have a mean of 0 and standard deviation of 1. You want to perform these normalizations using only SQL in Snowflake (no UDFs). Which of the following SQL transformations are most suitable?

○ Apply Min-Max scaling to all three features:
```
(income - MIN(income) OVER ()) / (MAX(income) OVER () - MIN(income) OVER ())
(loan_amount - MIN(loan_amount) OVER ()) / (MAX(loan_amount) OVER () - MIN(loan_amount) OVER ())
(credit_score - MIN(credit_score) OVER ()) / (MAX(credit_score) OVER () - MIN(credit_score) OVER ())
```

○ Apply Z-score standardization to all three features:
```
(income - AVG(income) OVER ()) / STDDEV(income) OVER ()
(loan_amount - AVG(loan_amount) OVER ()) / STDDEV(loan_amount) OVER ()
(credit_score - AVG(credit_score) OVER ()) / STDDEV(credit_score) OVER ()
```

○ Apply Robust Scaling to 'income' and 'loan_amount' and Z-score standardization to 'credit_score':
```
WITH Percentiles AS (
 SELECT
 APPROX_PERCENTILE(income, 0.25) AS income_q1,
 APPROX_PERCENTILE(income, 0.75) AS income_q3,
 APPROX_PERCENTILE(loan_amount, 0.25) AS loan_amount_q1,
 APPROX_PERCENTILE(loan_amount, 0.75) AS loan_amount_q3
 FROM your_table
)
SELECT
(income - (SELECT income_q1 FROM Percentiles)) / ((SELECT income_q3 FROM Percentiles) - (SELECT income_q1 FROM Percentiles)),
(loan_amount - (SELECT loan_amount_q1 FROM Percentiles)) / ((SELECT loan_amount_q3 FROM Percentiles) - (SELECT loan_amount_q1 FROM Percentiles)),
(credit_score - AVG(credit_score) OVER ()) / STDDEV(credit_score) OVER ()
FROM your_table;
```

○ Apply Log transformation to 'income' and 'loan_amount', and Z-score to credit score:
```
LOG(income), LOG(loan_amount), (credit_score - AVG(credit_score) OVER ()) / STDDEV(credit_score) OVER ()
```

○ Apply the arcsinh transformation for income and loan amount and Z-score for the credit score:
```
ASINH(income), ASINH(loan amount), (credit score - AVG(credit score) OVER ()) / STDDEV(credit score) OVER ()
```

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

정답：C

설명：

Option C is the most suitable. Robust Scaling is appropriate for 'income' and 'loan_amount' due to the presence of outliers. Robust scaling, using IQR is less sensitive to extreme values than Min-Max or Z-score. Z-score standardization is suitable for 'credit_score' as it has a roughly normal distribution, and standardization is desired. Option A is incorrect since Min-Max scaling is highly sensitive to outliers. Option B is incorrect because Z-score is not outlier resilient and it doesn't take into account the data properties given for credit score. Log transformation and arcsinh transform can handle outliers, they're not as resilient as robust scaling. The arcsinh transformation is also useful for features that may have negative values, but we don't have that information here.

질문 # 33
You are tasked with deploying a real-time fraud detection model in Snowflake. The model requires very low latency (under 100ms) to prevent fraudulent transactions. The input data is streamed into a Snowflake table. You are considering using either a Scalar or Vectorized Python UDF for scoring. Which of the following approaches and considerations are MOST critical for achieving the desired performance and reliability? Assume the model itself is computationally inexpensive. Select all that apply.

- A. Pre-load the model into a static variable within the UDF code, ensuring it's only loaded once per worker node.
- B. Use a Scalar UDF because it has lower overhead per invocation compared to a Vectorized UDF when processing individual transactions.
- C. Use a Vectorized UDF with a small 'MAX BATCH_SIZE to minimize latency while still leveraging vectorization benefits.
- D. Utilize Snowflake's Materialized Views to pre-compute frequently used features, reducing the amount of data the UDF needs to process.
- E. Configure Snowflake's Auto-Suspend feature to aggressively suspend the warehouse when idle, to minimize costs.

정답：A,C,D

설명：

For real-time fraud detection with low latency requirements, careful optimization is crucial. Vectorized UDFs (B) can be faster than scalar UDFs even with small batch sizes because of the reduced overhead per record compared to scalar UDFs. Pre-loading the model (C) is essential to avoid repeated model loading overhead. Using Materialized Views (D) to pre-compute features reduces the data the UDF needs to handle, improving performance. While scalar UDFs can have lower overhead per invocation, vectorized UDFs optimized with proper will generally provide better performance. Aggressively auto-suspending the warehouse (E) is

counterproductive as it introduces latency due to warehouse startup time.

질문 # 34

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

정답：A

설명：

Option E is the most efficient and accurate approach. It uses F.try_to_decimal 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.

질문 # 35

You are tasked with predicting the sales price of houses based on their size (square footage) using linear regression in Snowflake. You have a table named 'HOUSE_PRICES' with columns 'SQUARE_FOOTAGE' and 'SALES_PRICE'. You want to calculate the slope and intercept using Snowflake SQL. Which of the following queries, considering potential NULL values in the data, is the MOST robust and statistically sound for calculating the slope and intercept for a simple linear regression model?

```
SELECT COVAR_POP(SALES_PRICE, SQUARE_FOOTAGE) / VAR_POP(SQUARE_FOOTAGE) AS slope, AVG(SALES_PRICE) - (COVAR_POP(SALES_PRICE, SQUARE_FOOTAGE) /
VAR_POP(SQUARE_FOOTAGE)) AVG(SQUARE_FOOTAGE) AS intercept FROM HOUSE_PRICES;
```

```
SELECT STDDEV_POP(SALES_PRICE) / STDDEV_POP(SQUARE_FOOTAGE) AS slope, AVG(SALES_PRICE) - (STDDEV_POP(SALES_PRICE) / STDDEV_POP(SQUARE_FOOTAGE))
AVG(SQUARE_FOOTAGE) AS intercept FROM HOUSE_PRICES;
```

```
SELECT REGR_SLOPE(SALES_PRICE, SQUARE_FOOTAGE) AS slope, REGR_INTERCEPT(SALES_PRICE, SQUARE_FOOTAGE) AS intercept FROM HOUSE_PRICES;
```

```
SELECT CORR(SALES_PRICE, SQUARE_FOOTAGE) STDDEV(SALES_PRICE) / STDDEV(SQUARE_FOOTAGE) AS slope, AVG(SALES_PRICE) - (CORR(SALES_PRICE,
SQUARE_FOOTAGE) STDDEV(SALES_PRICE) / STDDEV(SQUARE_FOOTAGE)) AVG(SQUARE_FOOTAGE) AS intercept FROM HOUSE_PRICES WHERE SALES_PRICE IS NOT NULL AND
SQUARE_FOOTAGE IS NOT NULL;
```

```
SELECT (SUM(SQUARE_FOOTAGE SALES_PRICE) - SUM(SQUARE_FOOTAGE) SUM(SALES_PRICE) / COUNT( )) / (SUM(SQUARE_FOOTAGE SQUARE_FOOTAGE) -
SUM(SQUARE_FOOTAGE) SUM(SQUARE_FOOTAGE) / COUNT( )) AS slope, AVG(SALES_PRICE) - ((SUM(SQUARE_FOOTAGE SALES_PRICE) - SUM(SQUARE_FOOTAGE)
SUM(SALES_PRICE) / COUNT( )) / (SUM(SQUARE_FOOTAGE SQUARE_FOOTAGE) - SUM(SQUARE_FOOTAGE) SUM(SQUARE_FOOTAGE) / COUNT( ))) AVG(SQUARE_FOOTAGE) AS
intercept FROM HOUSE_PRICES WHERE SALES_PRICE IS NOT NULL AND SQUARE_FOOTAGE IS NOT NULL;
```

- A. Option A
- B. Option B
- C. Option C

- D. Option E
- E. Option D

정답：C

설명：
REGR SLOPE and REGR INTERCEPT are built-in Snowflake functions specifically designed for calculating the slope and intercept of a linear regression. These functions handle NULL values appropriately by excluding pairs where either SALES PRICE or SQUARE FOOTAGE is NULL, giving a statistically valid result. While options A, D, and E are theoretically correct in ideal conditions, they don't inherently handle NULL values and would require explicit filtering or potentially lead to incorrect calculations if NULLs are present. Option B uses standard deviation, which doesn't directly compute covariance needed for the slope. Option C is the most concise and reliable solution.

질문 # 36

......

우리KoreaDumps 에서 제공하는 학습가이드에는 IT전문가들이 만들어낸 시험대비 자료들과Snowflake DSA-C03인증시험의 완벽한 문제와 답들입니다. 그리고 우리KoreaDumps에서는 IT업계에서의 높은 신뢰감으로 여러분들한테 100%보장을 드립니다. 우리에 믿음을 드리기 위하여Snowflake DSA-C03관련자료의 일부분 문제와 답 등 샘플을 무료로 다운받아 체험해볼수 있게 제공합니다.

DSA-C03최고품질 덤프문제보기: https://www.koreadumps.com/DSA-C03_exam-braindumps.html

그리고Snowflake DSA-C03인증시험 패스는 진짜 어렵다고 합니다, Snowflake DSA-C03 덤프구매전 한국어 온라인상담서비스부터 구매후 덤프 무료 업데이트버전제공 , Snowflake DSA-C03시험불합격시 덤프비용 전액환불 혹은 다른 과목으로 교환 등 저희는 구매전부터 구매후까지 철저한 서비스를 제공해드립니다, Snowflake DSA-C03인기자격증 시험대비 덤프문제 MB2-706덤프를 주문하시면 결제후 즉시 고객님 메일주소에 시스템 자동으로 메일이 발송됩니다, Snowflake DSA-C03인기자격증 시험대비 덤프문제 덤프에 있는 문제만 열심히 공부하시면 시험통과 가능하기에 시간도 절약해줄수있어 최고의 믿음과 인기를 받아왔습니다, DSA-C03 덤프는 퍼펙트한 시험대비자료가 될 것입니다.

콜리의 모습이 완전히 드러났다, 그렇게 잡아 와 아무 데나 때렸다, 그리고Snowflake DSA-C03인증시험 패스는 진짜 어렵다고 합니다, Snowflake DSA-C03 덤프구매전 한국어 온라인상담서비스부터 구매후 덤프 무료 업데이트버전제공 , Snowflake DSA-C03시험불합격시 덤프비용 전액환불 혹은 다른 과목으로 교환 등 저희는 구매전부터 구매후까지 철저한 서비스를 제공해드립니다.

DSA-C03인기자격증 시험대비 덤프문제 100% 유효한 덤프문제

MB2-706덤프를 주문하시면 결제후 즉시 고객님 메일주소에 시스템 자DSA-C03동으로 메일이 발송됩니다, 덤프에 있는 문제만 열심히 공부하시면 시험통과 가능하기에 시간도 절약해줄수있어 최고의 믿음과 인기를 받아왔습니다.

DSA-C03 덤프는 퍼펙트한 시험대비자료가 될 것입니다.

- DSA-C03 덤프 Snowflake 인증 □ 시험 자료를 무료로 다운로드하려면⇒ www.koreadumps.com⇐을 통해▶ DSA-C03 ◀를 검색하십시오DSA-C03인기자격증 최신시험 덤프자료
- DSA-C03완벽한 시험기출자료 □ DSA-C03시험유효자료 □ DSA-C03퍼펙트 덤프공부자료 □ { www.itdumpskr.com }웹사이트를 열고✔ DSA-C03 □✔□를 검색하여 무료 다운로드DSA-C03완벽한 덤프문제
- 최근 인기시험 DSA-C03인기자격증 시험대비 덤프문제 덤프문제보기 □ "www.dumptop.com"은 □ DSA-C03 □무료 다운로드를 받을 수 있는 최고의 사이트입니다DSA-C03인기자격증 덤프문제
- DSA-C03시험패스 인증덤프문제 □ DSA-C03높은 통과율 덤프공부 □ DSA-C03시험패스 인증덤프문제 □ 무료 다운로드를 위해□ DSA-C03 □를 검색하려면「 www.itdumpskr.com 」을(를) 입력하십시오DSA-C03적중율 높은 시험덤프
- DSA-C03퍼펙트 덤프공부자료 □ DSA-C03최신버전 시험대비자료 □ DSA-C03높은 통과율 시험덤프 □ ✔ www.pass4test.net □✔□을(를) 열고➡ DSA-C03 □□□를 입력하고 무료 다운로드를 받으십시오DSA-C03인증시험 인기 덤프자료
- 시험대비 DSA-C03인기자격증 시험대비 덤프문제 덤프공부문제 □ "www.itdumpskr.com"은 《 DSA-C03 》무료 다운로드를 받을 수 있는 최고의 사이트입니다DSA-C03높은 통과율 인기 시험자료
- 시험패스 가능한 DSA-C03인기자격증 시험대비 덤프문제 공부자료 □ ⇒ DSA-C03 ⇐를 무료로 다운로드하려면➡ www.dumptop.com □웹사이트를 입력하세요DSA-C03퍼펙트 덤프공부자료