

DEA-C02復習範囲、DEA-C02模擬練習



2025年CertJukenの最新DEA-C02 PDFダンプおよびDEA-C02試験エンジンの無料共有: <https://drive.google.com/open?id=1rfF5gr66lldz2sHa4e2VhYE2fxjJHNbg>

これらの2つの特性により、DEA-C02ガイドトレントを使用するほぼすべての候補者が一度にテストに合格できることがわかります。これは自己決定ではありません。統計によると、当社のDEA-C02ガイドトレントは98%~99%の高い合格率を達成しており、これは他のすべてをかなり上回る程度です。同時に、DEA-C02テストトレントが毎日更新されるかどうかを確認する専門スタッフがいます。メールでお問い合わせいただく場合でも、オンラインでお問い合わせいただく場合でも、できるだけ早く問題を解決できるようサポートいたします。心配する必要はまったくありません。

あなたに相応しいCertJuken問題集を探していますか。DEA-C02試験備考資料の整理を悩んでいますか。専門化のIT認定試験資料提供者CertJukenとして、かねてより全面的の資料を準備します。あなたの資料を探す時間を節約し、Snowflake DEA-C02試験の復習をやっています。

>> DEA-C02復習範囲 <<

有効的なDEA-C02復習範囲 & 合格スムーズDEA-C02模擬練習 |100%合格率のDEA-C02トレーニング費用

CertJukenのSnowflakeのDEA-C02試験トレーニング資料はあなたに時間とエネルギーを節約させます。あなたが何ヶ月でもやる必要があることを我々はやってさしあげましたから。あなたがすべきことは、CertJukenのSnowflakeのDEA-C02試験トレーニング資料に受かるのです。あなた自身のために、証明書をもらいます。CertJukenはあなたに必要とした知識と経験を提供して、SnowflakeのDEA-C02試験の目標を作ってあげました。CertJukenを利用したら、試験に合格しないことは絶対ないです。

Snowflake SnowPro Advanced: Data Engineer (DEA-C02) 認定 DEA-C02 試験問題 (Q133-Q138):

質問 # 133

You have a directory table 'my_directory_table' pointing to a stage containing CSV files with headers. You need to query the

directory table to find all files modified in the last 24 hours and load those CSV files using COPY INTO into a target table. Assume the target table exists and has appropriate schema. Which of the following SQL statements, or set of statements, will accomplish this efficiently? Note: Consider efficient file loading.

- A.
COPY INTO my_target_table FROM @my_stage/path_to_csv FILE_FORMAT = (TYPE = CSV, SKIP_HEADER = 1) PATTERN = '. ' || (SELECT relative_path FROM my_directory_table WHERE last_modified >= CURRENT_TIMESTAMP() - INTERVAL '1 day') || ' ';
- B.
SELECT \$1 FROM @my_stage/path_to_csv WHERE metadata\$filename IN (SELECT relative_path FROM my_directory_table WHERE last_modified >= CURRENT_TIMESTAMP() - INTERVAL '1 day');
- C.
COPY INTO my_target_table FROM @my_stage/path_to_csv FILE_FORMAT = (TYPE = CSV, SKIP_HEADER = 1) FILES = (SELECT relative_path FROM my_directory_table WHERE last_modified >= CURRENT_TIMESTAMP() - INTERVAL '1 day');
- D.
COPY INTO my_target_table FROM @my_stage/path_to_csv FILES = (SELECT relative_path FROM my_directory_table WHERE last_modified >= CURRENT_TIMESTAMP() - INTERVAL '1 day');
- E.
CREATE TEMPORARY TABLE files_to_load AS SELECT relative_path FROM my_directory_table WHERE last_modified >= CURRENT_TIMESTAMP() - INTERVAL '1 day';
COPY INTO my_target_table FROM @my_stage/path_to_csv FILES = (SELECT relative_path FROM files_to_load);

正解: C

解説:

Option E is the most efficient way to load the files using 'COPY INTO' and the 'FILES' parameter. The 'FILES' parameter accepts a list of filenames to load. Option D is incorrect because the 'PATTERN' parameter expects a regular expression, not a concatenation of filenames. Options A, B, and C require more complex processing. Using subquery expression in the copy command is the best approach for filtering files based on directory table data.

質問 # 134

You are implementing row access policies on a 'SALES DATA' table to restrict access based on the 'REGION' column. Different users are allowed to see data only for specific regions. You have a mapping table 'USER REGION MAP' with columns 'USERNAME' and 'REGION'. You want to create a row access policy that dynamically filters the 'SALES DATA' based on the user and their allowed region. Which of the following options represents a correct approach to create and apply this row access policy?

- CREATE ROW ACCESS POLICY region_policy AS (region STRING) RETURNS BOOLEAN -> CURRENT_USER() IN (SELECT USERNAME FROM USER_REGION_MAP WHERE REGION = region); ALTER TABLE SALES_DATA ADD ROW ACCESS POLICY region_policy ON REGION;
- CREATE ROW ACCESS POLICY region_policy AS (region STRING) RETURNS BOOLEAN -> EXISTS (SELECT 1 FROM USER_REGION_MAP WHERE USERNAME = CURRENT_USER() AND REGION = region); ALTER TABLE SALES_DATA ADD ROW ACCESS POLICY region_policy ON REGION;
- CREATE ROW ACCESS POLICY region_policy AS (region STRING) RETURNS BOOLEAN -> EXISTS (SELECT 1 FROM USER_REGION_MAP WHERE USERNAME = CURRENT_USER() AND REGION = region); ALTER TABLE SALES_DATA SET ROW ACCESS POLICY region_policy ON (REGION);
- CREATE ROW ACCESS POLICY region_policy AS (region STRING) RETURNS BOOLEAN -> EXISTS (SELECT 1 FROM USER_REGION_MAP WHERE USERNAME = CURRENT_USER() AND REGION = region); ALTER TABLE SALES_DATA MODIFY ROW ACCESS POLICY region_policy ON (REGION);
- CREATE ROW ACCESS POLICY region_policy AS (region STRING) RETURNS BOOLEAN -> EXISTS (SELECT 1 FROM USER_REGION_MAP WHERE USERNAME = CURRENT_ROLE() AND REGION = region); ALTER TABLE SALES_DATA ADD ROW ACCESS POLICY region_policy ON REGION;

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

正解: C

解説:

Option B is the correct approach. It creates a row access policy that checks if a row exists in the table where the username matches the current user and the region matches the 'REGION' column in the 'SALES_DATA' table. 'ADD ROW ACCESS POLICY' is the correct command to apply the policy. Option A is incorrect as it uses IN clause, which can become inefficient with large datasets. Option C uses 'SET' which is not a valid operation, and Option D uses 'MODIFY' which is used for masking policy and not row access policy. Option E uses 'CURRENT_ROLE' instead of 'CURRENT_USER' which is not the appropriate filter criteria.

質問 # 135

You are designing a data sharing solution for a multi-tenant application where each tenant's data must be isolated. You have a 'sales' table with a 'tenant_id' column. You need to implement row-level security to ensure that each tenant can only access their own data when querying the shared table. Which of the following approaches, considering performance and security, is the MOST suitable for implementing this row-level filtering in Snowflake?

- A. Use Snowflake's data masking policies to mask all data for tenants other than the one currently querying the table.
- B. Create a separate VIEW for each tenant, filtering by 'tenant_id'. Grant each tenant access only to their respective view.
- **C. Implement a row access policy on the 'sales' table that filters data based on the 'tenant_id' column and the current role or user context.**
- D. Create a scheduled task that duplicates the sales table into a new table for each tenant, filtering by the tenant_id.
- E. Implement a user-defined function (UDF) that checks the current user's tenant ID and returns a boolean value indicating whether the row should be visible. Use this UDF in a WHERE clause in every query.

正解: C

解説:

Row access policies are the most efficient and scalable solution for row-level security in Snowflake. They are applied at the table level and automatically enforced for all queries, ensuring consistent data isolation. Creating separate views for each tenant is administratively burdensome and doesn't scale well. UDFs can impact performance. Data masking policies do not filter rows, they only redact data. Creating scheduled task create data duplication, its not ideal for data movements.

質問 # 136

A Snowflake data warehouse contains a table 'WEB EVENTS' with columns like 'EVENT ID', 'EVENT TIMESTAMP', 'USER', 'PAGE URL', and 'SESSION ID'. The data engineering team has enabled search optimization on 'PAGE URL' because analysts frequently filter on specific URLs. However, they notice that queries filtering on multiple 'PAGE URL' values (e.g., using 'WHERE PAGE URL IN ('url1', 'url2)', are not performing as well as expected. What are the potential reasons for this behavior, and what strategies can be used to improve performance in this scenario? Select all that apply:

- **A. Search optimization is not designed to efficiently handle IN list lookups with a large number of values. Consider using a temporary table or common table expression (CTE) to pre-filter the data.**
- **B. The number of distinct values in the 'PAGE URL' column is very high, leading to a large search access path, making IN list lookups inefficient. Consider clustering by PAGE_URL**
- C. The warehouse size is too small to handle the complexity of the IN list lookup. Increase the warehouse size.
- D. Search optimization is automatically disabled when using IN clause, therefore it is important to rewrite the query without using IN operator.
- E. Statistics on the 'PAGE URL' column are outdated. Run 'ANALYZE TABLE WEB EVENTS' to refresh the statistics.

正解: A、 B

解説:

Options A and B are correct. A high cardinality in (Option A) means the search access path is large, making IN list lookups expensive. Search optimization is generally optimized for point lookups or small range scans, not large IN lists. Option B suggests using a temporary table or CTE to pre-filter the data, which can significantly improve performance by reducing the amount of data the optimizer needs to consider during the final filtering step. Option C (increasing warehouse size) might improve performance, but it doesn't directly address the issue of IN list lookups. Option D (analyzing the table) is always a good practice, but it's unlikely to be the primary cause of the slow IN list performance. Option E is wrong. Search optimization is not automatically disabled when using IN clause

質問 # 137

A data engineering team is loading a large fact table 'SALES DATA' daily, partitioned by 'SALE DATE'. After several months, query performance degrades significantly. An analyst reports that queries filtering on 'CUSTOMER ID' are slow, despite 'CUSTOMER ID' having high cardinality. The table definition is as follows: CREATE TABLE SALES_DATA (SALE DATE DATE NOT NULL, CUSTOMER_ID NUMBER NOT NULL, PRODUCT ID NUMBER NOT NULL, SALE_AMOUNT ... Which of the following actions would BEST improve query performance for queries filtering on 'CUSTOMER ID', considering the existing partitioning by 'SALE DATE'?

- A. Increase the virtual warehouse size.
- B. Create a secondary index on 'CUSTOMER ID'
- **C. Cluster the 'SALES DATA' table on 'CUSTOMER ID'.**

- D. Partition the table by 'CUSTOMER_ID' instead of 'SALE_DATE'.
- E. Create a materialized view that aggregates data by 'CUSTOMER_ID' and relevant dimensions.

正解: C

解説:

Clustering the table on 'CUSTOMER_ID' will physically organize the data based on this column, improving the performance of queries filtering on 'CUSTOMER ID'. While increasing warehouse size (E) might provide some performance boost, clustering addresses the underlying data organization issue. Secondary indexes (A) are not supported in Snowflake. Partitioning by 'CUSTOMER_ID' (D) isn't possible in Snowflake. Materialized views (B) are a valid option for pre-aggregation, but clustering will directly improve base table performance for filtering. Therefore, clustering is the best option.

質問 # 138

.....

いろいろな人はSnowflakeのDEA-C02を長い時間で復習して試験のモードへの不適応で失敗することを心配していますから、我々CertJukenはあなたに試験の前に試験の真実なモードを体験させます。SnowflakeのDEA-C02試験のソフトは問題数が豊富であなたに大量の練習で能力を高めさせます。そのほかに、専門家たちの解答への詳しい分析があります。あなたにSnowflakeのDEA-C02試験に自信を持たせます。

DEA-C02模擬練習: <https://www.certjuken.com/DEA-C02-exam.html>

Snowflake DEA-C02復習範囲 もちろん、資格試験を審査するとき、非公開にすることはできません、Snowflake DEA-C02復習範囲 一年後、あなたは最新の勉強資料を使って続けたいなら、いくつかの費用を支払うのは必要です、CertJuken DEA-C02模擬練習は同業の中でそんなに良い地位を取るの理由は弊社のかかなり正確な試験の練習問題と解答そえに迅速の更新で、このようにとても良い成績がとられています、当社のDEA-C02模擬試験は、DEA-C02試験に合格して認定を取得するのに非常に役立つためです、Snowflake DEA-C02復習範囲 払い戻しプロセス中に問題が発生した場合は、いつでもカスタマーサービススタッフに連絡することもできます。

その引渡しの時、目録に犬の数の記載があったという、論文によると、これは主要な推進要因DEA-C02復習範囲によるものです、もちろん、資格試験を審査するとき、非公開にすることはできません、一年後、あなたは最新の勉強資料を使って続けたいなら、いくつかの費用を支払うのは必要です。

DEA-C02試験の準備方法 | 便利なDEA-C02復習範囲試験 | 一番優秀なSnowPro Advanced: Data Engineer (DEA-C02)模擬練習

CertJukenは同業の中でそんなに良い地位を取るの理由は弊社のかかなり正確な試験の練習問題と解答そえに迅速の更新で、このようにとても良い成績がとられています、当社のDEA-C02模擬試験は、DEA-C02試験に合格して認定を取得するのに非常に役立つためです。

払い戻しプロセス中に問題が発生したDEA-C02場合は、いつでもカスタマーサービススタッフに連絡することもできます。

- 最新-効率的なDEA-C02復習範囲試験-試験の準備方法DEA-C02模擬練習 □ ➡ DEA-C02 □ の試験問題は □ www.xhs1991.com □ で無料配信中DEA-C02試験解説問題
- DEA-C02試験の準備方法 | 有難いDEA-C02復習範囲試験 | 検証するSnowPro Advanced: Data Engineer (DEA-C02)模擬練習 □ ✓ www.goshiken.com □ ✓ □ の無料ダウンロード ➡ DEA-C02 □ ページが開きますDEA-C02無料試験
- 素晴らしいSnowflake DEA-C02: SnowPro Advanced: Data Engineer (DEA-C02)復習範囲 - 有用的な www.japancert.com DEA-C02模擬練習 □ 《 www.japancert.com 》は、【 DEA-C02 】を無料でダウンロードするのに最適なサイトですDEA-C02全真問題集
- DEA-C02日本語版テキスト内容 □ DEA-C02試験解答 □ DEA-C02全真問題集 □ ▷ DEA-C02 ◁ を無料でダウンロード 【 www.goshiken.com 】 ウェブサイトを入力するだけDEA-C02テスト難易度
- DEA-C02試験の準備方法 | 有難いDEA-C02復習範囲試験 | 検証するSnowPro Advanced: Data Engineer (DEA-C02)模擬練習 □ ➡ jp.fast2test.com □ には無料の ➡ DEA-C02 □ 問題集がありますDEA-C02試験問題
- 最新-効率的なDEA-C02復習範囲試験-試験の準備方法DEA-C02模擬練習 □ ✓ www.goshiken.com □ ✓ □ に移動し、【 DEA-C02 】を検索して、無料でダウンロード可能な試験資料を探しますDEA-C02日本語版テキスト内容
- 素晴らしいSnowflake DEA-C02: SnowPro Advanced: Data Engineer (DEA-C02)復習範囲 - 有用的な www.mogixexam.com DEA-C02模擬練習 □ 【 www.mogixexam.com 】 には無料の ▷ DEA-C02 ◁ 問題集がありま

すDEA-C02テスト難易度

- [illegible]

ちなみに、CertJuken DEA-C02の一部をクラウドストレージからダウンロードできます：<https://drive.google.com/open?id=1rff5gr66lldz2sHa4e2VhYE2fXjJHNbg>