

## Snowflake.ARA-R01.v2024-03-09.q49

試験コード:	ARA-R01
試験名称:	SnowPro Advanced: Architect Recertification Exam
認定資格:	Snowflake
無料問題数:	49
バージョン:	v2024-03-09
アクセス数:	621
ページビュー数:	490
<a href="https://www.jpnpdf.com/Snowflake.ARA-R01.v2024-03-09.q49-mondaishu.html">https://www.jpnpdf.com/Snowflake.ARA-R01.v2024-03-09.q49-mondaishu.html</a>	

### 最新問題: 1

すべての Snowflake アカウントが Enterprise エディション以上を使用していると仮定すると、データのコピーが必要な開発およびテストのシナリオはどれですか？ゼロコピーのクローン作成は適切ではありません。(2 つ選択してください)。

- A. 開発者は、ライブ データの変換されたバージョンに対して作業するための独自のデータセットを作成します。
- B. 運用環境と開発環境は同じアカウント内の異なるデータベースで実行されます。開発者は、特定の列がマスクされた運用環境のようなデータを表示する必要があります。
- C. データは実稼働 Snowflake アカウントにあり、同じクラウド リージョン内の別の開発/テスト Snowflake アカウントの開発者に提供する必要があります。
- D. 開発者は、初期開発と単体テストのために、開発アカウントで以前に作成された標準テスト データベースの独自のコピーを作成します。
- E. リリース プロセスでは、運用規模と複雑さのデータを使用した変更の運用前テストが必要です。セキュリティ上の理由から、実稼働前も実稼働アカウントで実行されます。

**Answer: A,C (メッセージを残す)**

ゼロコピー クローン作成は、データを物理的にコピーせずにテーブル、スキーマ、またはデータベースのクローンを作成できる機能です。ゼロコピー クローン作成は、クローン オブジェクトが元のオブジェクトと同じデータおよびメタデータを持つ必要があります、クローン オブジェクトを頻繁に変更または更新する必要がないシナリオに適しています。ゼロコピー クローン作成は、クローン作成されたオブジェクトを同じ Snowflake アカウント内、または同じクラウド リージョン内の異なるアカウント間で共有する必要があるシナリオにも適しています<sup>2</sup>。ただし、ゼロコピー クローン作成は、クローン作成されたオブジェクトに異なる属性を持たせる必要があるシナリオには適していません。元のオブジェクトよりもデータやメタデータが大きい場合、または複製されたオブジェクトを頻繁に変更または更新する必要がある場合。

ゼロコピー クローン作成は、クローン化されたオブジェクトを異なるクラウド リージョンの異なるアカウント間で共有する必要があるシナリオにも適していません。これらのシナリオでは、COPY INTO コマンドを使用するか、セキュア ビューでのデータ共有を使用することによって、データのコピーが必要になります<sup>3</sup>。以下は、データのコピーが必要で、ゼロコピー クローン作成が必要ない開発およびテスト シナリオの例です。適切であること：

開発者は、ライブ データの変換されたバージョンに対して作業するための独自のデータセットを作成します。このシナリオでは、開発者が列、行、値の追加、削除、更新などの変換を実行するためにクローン オブジェクトのデータまたはメタデータを変更する必要があるため、データのコピーが必要です。ゼロコピー クローン作成は、元のオブジェクトと同じデータとメタデータを共有する読み取り専用クローンを作成することになり、クローンに加えられた変更は元のオブジェクトにも影響を与えるため、適切ではありません<sup>4</sup> データは運用 Snowflake アカウントにありますこれは、同じクラウド リージョン内の別の開発/テスト Snowflake アカウントの開発者に提供する必要があります。このシナリオでは、同じクラウド リージョン内の異なるアカウント間でデータを共有する必要があるため、データのコピーが必要です。ゼロコピー クローン作成は、元のオブジェクトと同じアカウント内にクローンを作成することになり、そのクローンを別のアカウントと共有できないため、適切ではありません。同じクラウド リージョン内の異なるアカウント間でデータを共有するには、安全なビューまたは COPY INTO コマンドを使用したデータ共有を使用できます<sup>5</sup>。以下は、ゼロ コピー クローン作成が適しており、データのコピーが不要な開発およびテスト シナリオの例です。:

本番環境と開発環境は同じアカウント内の異なるデータベースで実行され、開発者は本番環境と同様のデータを、特定の列がマスクされた状態で表示する必要があります。このシナリオでは、データを同じアカウント内で共有する必要があり、複製されたオブジェクトには元のオブジェクトとは異なるデータやメタデータが必要ないため、ゼロコピー クローン作成を使用できます。ゼロコピー クローン作成では、運用データベースのクローンを開発データベースに作成でき、そのクローンには元のデータベースと同じデータとメタデータを含めることができます。特定の列をマスクするために、クローンの上にセキュア ビューを作成でき、開発者はクローンに直接アクセスする代わりにセキュア ビューにアクセスできます<sup>6</sup> 開発者は、開発アカウントで事前に作成された標準テスト データベースの独自のコピーを作成します。初期開発と単体テスト。このシナリオでは、データを同じアカウント内で共有する必要があり、複製されたオブジェクトには元のオブジェクトとは異なるデータやメタデータが必要ないため、ゼロコピー クローン作成を使用できます。ゼロコピー クローン作成では、開発者ごとに標準テスト データベースのクローンを作成でき、そのクローンには元のデータベースと同じデータとメタデータを含めることができます。開発者は、最初の開発と単体テストにクローンを使用でき、クローンに加えられた変更は元のデータベースや他のクローンに影響を与えません<sup>7</sup>。リリース プロセスでは、運用規模と複雑さのデータを使用した変更の運用前テストが必要です。セキュリティ上の理由から、実稼働前も実稼働アカウントで実行されます。このシナリオでは、データを同じアカウント内で共有する必要があり、複製されたオブジェクトには元のオブジェクトとは異なるデータやメタデータが必要ないため、ゼロコピー クローン作成を使用できます。ゼロコピー クローン作成では、実稼働前データベースに実稼働データベースのクローンを作成でき、そのクローンには元のデータベースと同じデータとメタデータを含めることができます。実稼働前テストでは、クローンを使用して、実稼働の規模と複雑さのデータによる変更をテストできます。クローンに加えられた変更は、元のデータベースや実稼働環境には影響しません<sup>8</sup>。

1: SnowPro Advanced: アーキテクト | 学習ガイド <sup>9</sup>

2: スノーフレイクのドキュメント | クローン作成の概要

3: スノーフレイクのドキュメント | COPY を使用したテーブルへのデータのロード

4: スノーフレイクのドキュメント | ロード中のデータの変換

5: スノーフレイクのドキュメント | データ共有の概要

6: スノーフレイクのドキュメント | 安全なビュー

7: スノーフレイクのドキュメント | データベース、スキーマ、テーブルのクローン作成

8: スノーフレイクのドキュメント | テストと開発のためのクローン作成

- : SnowPro Advanced: アーキテクト | 学習ガイド
- : クローン作成の概要
- : COPY を使用したテーブルへのデータのロード
- : ロード中のデータの変換
- : データ共有の概要
- : 安全なビュー
- : データベース、スキーマ、テーブルのクローン作成
- : テストと開発のためのクローン作成

## 最新問題: 2

A 社は最近 B 社を買収しました。

企業 B の Snowflake デプロイメントは、Azure 西ヨーロッパ リージョンにあります。

統合プロセスの一環として、アーキテクトは、B 社の販売データを、AWS us-east-1 リージョンにある A 社の Snowflake アカウントに統合するように依頼されました。

この要件はどのように満たされるのでしょうか？

- A.** Snowflake 内のクロスリージョン データ レプリケーションを使用して、会社 B の Snowflake アカウントから会社 A の Snowflake アカウントに販売データをレプリケートします。B 社のアカウントから A 社のアカウントへの直接共有を構成します。
- B.** B 社の Snowflake アカウントから売上データを CSV ファイルとしてエクスポートし、そのファイルを A 社の Snowflake アカウントに転送します。Snowflake のデータ読み込み機能を使用してデータをインポートします。
- C.** 企業 B の Snowflake デプロイメントを企業 A の Snowflake デプロイメントと同じリージョンに移行し、データの局所性を確保します。次に、販売データの直接データベース間マージを実行します。
- D.** Azure Data Factory または同様のツールを使用してカスタム データ パイプラインを構築し、B 社の Snowflake アカウントから販売データを抽出します。データを変換して、A 社の Snowflake アカウントにロードします。

**Answer: A (メッセージを残す)**

企業 B の販売データを企業 A の Snowflake アカウントに統合するという要件を満たす最良の方法は、Snowflake 内でリージョン間のデータ レプリケーションを使用することです。この機能により、データ プロバイダーは、さまざまな地域やクラウド プラットフォームにわたってデータ コンシューマーとデータを安全に共有できます。Azure 西ヨーロッパ リージョンの企業 B のアカウントから AWS us-east-1 リージョンの企業 A のアカウントに売上データをレプリケートすると、データが同期され、利用できるようになります。データ レプリケーションを有効にするには、アカウントがリンクされ、ORGADMIN ロールを持つユーザーによってレプリケーションが有効になる必要があります。次に、レプリケーション グループを作成し、販売データベースをそのグループに追加する必要があります。最後に、レプリケートされたデータへのアクセスを許可するには、B 社のアカウントから A 社のアカウントへの直接共有を構成する必要があります。このオプションは、CSV ファイルを使用してデータをエクスポートおよびインポートしたり、Snowflake デプロイメント全体を別のリージョンまたはクラウド プラットフォームに移行したりするよりも効率的かつ安全です。また、外部ツールを使用してカスタム データ パイプラインを構築する必要もありません。

参考文献:

地域およびクラウド プラットフォーム間でデータを安全に共有する  
レプリケーションとフェイルオーバーの概要

## レプリケーションに関する考慮事項 アカウントオブジェクトの複製

### 最新問題: 3

ある企業は、主要なエンタープライズ ID プロバイダーを Snowflake によるフェデレーション認証と統合したいと考えています。

認証統合が構成され、Snowflake でロールが作成されました。ただし、ユーザーは作成時に Snowflake に自動的に表示されず、ユーザーのグループメンバーシップは割り当てられたローテに反映されません。

運用上のオーバーヘッドを最小限に抑えて、不足している機能を有効にするにはどうすればよいでしょうか？

A. OAuth は、アイデンティティ プロバイダーと Snowflake の間で構成する必要があります。次に、ユーザーとロールの適切なマッピングを使用して認可サーバーを構成する必要があります。

B. OAuth は、アイデンティティ プロバイダーと Snowflake の間で構成する必要があります。次に、ユーザーの適切なマッピングを使用して認可サーバーを構成し、ロール割り当ての適切なマッピングを使用してリソースサーバーを構成する必要があります。

C. SCIM はアイデンティティ プロバイダーと Snowflake の間で有効にする必要があります。SCIM を通じて両方が同期されると、それらのグループが Snowflake でグループアカウントとして作成され、適切なロールを付与できるようになります。

D. SCIM は、アイデンティティ プロバイダーと Snowflake の間で有効にする必要があります。SCIM を介して両方が同期されたら、ユーザーは自動的に作成され、そのグループのメンバーシップが Snowflake のロールとして反映されます。

**Answer: D (メッセージを残す)**

エンタープライズ ID プロバイダーをフェデレーション認証と統合し、Snowflake で自動ユーザー作成とロール割り当てを有効にする最良の方法は、SCIM (クロスドメイン ID 管理システム) を使用することです。SCIM を使用すると、Snowflake はアイデンティティ プロバイダーと同期し、アイデンティティ プロバイダーから提供された情報に基づいてユーザーとグループを作成できます。グループは Snowflake のロールにマップされ、ユーザーにはグループメンバーシップに基づいてロールが割り当てられます。こうすることで、アイデンティティ プロバイダーがユーザーとグループ管理の信頼できる情報源となり、Snowflake は手動介入なしで変更を自動的に反映します。他のオプションは、認証やユーザー プロビジョニングではなく承認用のプロトコルである OAuth の使用を必要とし、承認サーバーとリソースサーバーの追加構成が必要なため、不正確または不完全です。

### 最新問題: 4

USER\_01 という名前のユーザーは、スキーマ EDW 上にマテリアライズド ビューを作成するためのアクセス権が必要です。STG\_SCHEMA。このアクセスはどのようにして提供できるのでしょうか？

A. スキーマ EDW.STG\_SCHEMA でのマテリアライズド ビューの作成をユーザーに許可します  
USER\_01;

B. ユーザー USERJD1 にデータベース EDW でのマテリアライズド ビューの作成を許可します。

C. ユーザー USER\_01 にロール NEW\_ROLE を付与します。

スキーマ ECW.STG\_SCHEKA のマテリアライズド ビューの作成を NEW\_ROLE に付与します。

D. USER\_01 にロール NEW\_ROLE を付与します。

EDW.STG\_SCHEMA のマテリアライズド ビューの作成を NEW\_ROLE に付与します。

**Answer: A (メッセージを残す)**

正解は A です。これにより、スキーマ EDW.STG\_SCHEMA 上にマテリアライズド ビューを作成する特定の権限がユーザー USER\_01 に直接付与されます。

オプション B は、データベース EDW 全体に対してマテリアライズド ビューを作成する権限を付与するため、不正解です。これは範囲が広すぎて不必要です。また、ユーザー名にタイプミスがあります (USER\_01 ではなく USERJD1)。

オプション C は、別のスキーマ (EDW.STG\_SCHEMA ではなく ECW.STG\_SCHEKA) でマテリアライズド ビューを作成する権限を付与するため、不正解です。また、この目的のために新しいロールを作成する必要もありません。

オプション D は、無効なオブジェクトに対してマテリアライズド ビューを作成する権限を付与するため、不正解です (EDW.STG\_SCHEMA は有効なスキーマ名ではありません。EDW.STG\_SCHEMA である必要があります)。また、この目的のために新しいロールを作成する必要もありません。参考文献:

Snowflake ドキュメント: マテリアライズド ビューの作成

Snowflake ドキュメント: マテリアライズド ビューの操作

[Snowflake ドキュメント: スキーマに対する GRANT 権限]

最新問題: 5

アーキテクトは次の SQL クエリを実行します。

```
SELECT
  METADATA$FILENAME,
  METADATA$FILE_ROW_NUMBER
FROM @FILEROWS/Food_Reviews.csv
  (file_format=CSV_N)
```

このクエリはどのように解釈できるでしょうか?

- A. FILEROWS はステージです。FILE\_ROW\_NUMBER はファイル内の行番号です。
- B. FILEROWS はテーブルです。FILE\_ROW\_NUMBER はテーブル内の行番号です。
- C. FILEROWS はファイルです。FILE\_ROW\_NUMBER はファイル形式の場所です。
- D. FILERONS はファイル形式の場所です。FILE\_ROW\_NUMBER はステージです。

**Answer: A (メッセージを残す)**

ステージは、データのロードおよびアンロード用のファイルを保存できる Snowflake 内の名前付きの場所です。ステージは、ファイルが保存されている場所に応じて、内部または外部になります。

質問のクエリでは、LIST 関数を使用して、FILEROWS という名前のステージ内のファイルを一覧表示します。この関数は、ステージ内のファイルの行番号である FILE\_ROW\_NUMBER を含む、さまざまな列を含むテーブルを返します。

したがって、クエリは、FILEROWS という名前のステージ内のファイルをリストし、ステージ内の各ファイルの行番号を示すものとして解釈できます。

参考文献:

: ステージ

: リスト関数

**最新問題: 6**

Snowflake がテーブル内のクラスタリング情報を監視するために提供するシステム関数 (2 つ選択してください)。

- A. SYSTEM\$CLUSTERING\_INFORMATION
- B. SYSTEM\$CLUSTERING\_USAGE
- C. SYSTEM\$CLUSTERING\_DEPTH
- D. SYSTEM\$CLUSTERING\_KEYS
- E. SYSTEM\$CLUSTERING\_PERCENT

**Answer: A,C (メッセージを残す)**

Snowflake のドキュメントによると、これら 2 つのシステム関数は、テーブル内のクラスタリング情報を監視するために Snowflake によって提供されています。システム関数は、アクションを実行したり、システムに関する情報を返したりできる関数のタイプです。クラスタリング キーは、テーブル内の 1 つ以上の列に基づいてマイクロパーティション全体でデータを整理できる機能です。クラスタリングにより、スキャンするファイルの数が減り、クエリのパフォーマンスが向上します。

SYSTEM\$CLUSTERING\_INFORMATION は、テーブル内の 1 つ以上の列に基づいてテーブルのクラスタリング情報 (平均クラスタリング深度など) を返すシステム関数です。この関数はテーブル名とオプションの列名または式を引数として受け取り、クラスタリング情報を含む JSON 文字列を返します。クラスタリング情報には、キーによるクラスタリング、総パーティション数、総定数パーティション数、平均オーバーラップ数、および平均深さ 1 が含まれます。

SYSTEM\$CLUSTERING\_DEPTH は、テーブル内の 1 つ以上の列に基づいてテーブルのクラスタリングの深さを返すシステム関数です。この関数はテーブル名とオプションの列名または式を引数として受け取り、クラスタリングの深さを含む整数値を返します。クラスタリングの深さは、テーブル内のマイクロパーティションの重複するマイクロパーティションの最大数です。クラスタリングの深さが低いほど、クラスタリングが良好であることを示します<sup>2</sup>。

参考文献:

SYSTEM\$CLUSTERING\_INFORMATION | スノーフレイクのドキュメント

SYSTEM\$CLUSTERING\_DEPTH | スノーフレイクのドキュメント

**最新問題: 7**

アーキテクトは、ユーザーが受信共有からデータベースを作成できるようにする必要があります。

この要件を満たすには、ユーザーのロールにはどの権限が必要ですか? (2つお選びください。)

- A. インポートシェア;
- B. インポート権限;
- C. データベースを作成します。
- D. 共有を作成します。
- E. データベースをインポートします。

**Answer: C,E (メッセージを残す)**

Snowflake のドキュメントによると、受信共有からデータベースを作成するには、ユーザーのロールに次の権限が必要です。

現在のアカウントの CREATE DATABASE 権限。この権限により、ユーザーは account1 に新しいデータベースを作成できます。

共有に対する IMPORT DATABASE 権限。この権限により、ユーザーはデータベースを共有からアカウントにインポートできます<sup>2</sup>。リストされている他の権限は、この要件には関係ありません。IMPORT SHARE 権限は、データベースではなく、アカウントに共有をインポートするために使用されます<sup>3</sup>。IMPORT PRIVILEGES 権限は、オブジェクト自体ではなく、共有オブジェクトに付与された権限をインポートするために使用されます<sup>2</sup>。CREATE SHARE 権限は、他のアカウントからのデータを消費するためではなく、他のアカウントにデータを提供するための共有を作成するために使用されます<sup>4</sup>。

参考文献:

データベースの作成 | スノーフレイクのドキュメント

共有からのデータのインポート | スノーフレイクのドキュメント

共有のインポート | スノーフレイクのドキュメント

共有を作成 | スノーフレイクのドキュメント

### 最新問題: 8

あなたは組織内のスノーフレイク アーキテクトです。ビジネス チームは、Tableau を通じて視覚化できるデータをロードする必要があるユースケースをデプロイすることにしました。毎日新しいデータが入ってくるので、古いデータは必要なくなります。

この場合、コストを最適化するためにどのタイプのテーブルを使用するか

A. トランジェント

B. 一時的

C. 永続的

**Answer: A (メッセージを残す)**

一時テーブルは、フェイルセーフ期間がなく、0 日または 1 日のタイム トラベル保持期間を持つことができる Snowflake のテーブルのタイプです。一時テーブルは、簡単に再現または複製できる一時データまたは中間データに適しています<sup>1</sup>。

一時テーブルは、セッションが終了するか現在のユーザーがログアウトすると自動的に削除される Snowflake のテーブルのタイプです。一時テーブルにはストレージのコストはかかりませんが、他のユーザーやセッションには表示されません<sup>2</sup>。

永続テーブルは、フェイルセーフ期間と最大 90 日間のタイム トラベル保持期間を持つ Snowflake のテーブルのタイプです。永続テーブルは、偶発的または悪意のある削除から保護する必要がある永続的で耐久性のあるデータに適しています<sup>3</sup>。

この場合、ユースケースでは、Tableau を通じて視覚化できるいくつかのデータをロードする必要があります。データは毎日更新されるため、古いデータは必要なくなります。したがって、この場合にコストを最適化するために使用するテーブルの最適なタイプは一時テーブルです。これは、フェイルセーフ コストが発生せず、タイム トラベルの保持期間が 0 日または 1 日と短いためです。こうすることで、不要なストレージ コストを発生させることなく、Tableau でデータをロードしてクエリした後、削除または上書きすることができます。

参考資料: : 一時テーブル : 一時テーブル : タイムトラベルの理解と使用

### 最新問題: 9

データはインポートされ、VARIANT 列に JSON として保存されます。クエリのパフォーマンスは問題ありませんでしたが、最近ではクエリのパフォーマンスが低下していることが報告されています。

この原因は何でしょうか？

- A. 最近のデータ インポートに JSON null がありました。
- B. JSON 内のキーの順序が変更されました。
- C. 最近のデータ インポートには、通常よりも少ないフィールドが含まれていました。
- D. 最近のデータ インポートの JSON 値の文字列の長さにばらつきがありました。

**Answer: B,D (メッセージを残す)**

データはインポートされ、VARIANT 列に JSON として保存されます。クエリのパフォーマンスは問題ありませんでしたが、最近ではクエリのパフォーマンスが低下していることが報告されています。これは次の要因によって引き起こされる可能性があります。

JSON 内のキーの順序が変更されました。Snowflake は、半構造化データを、最も一般的な要素については列のような構造に内部的に格納し、残りは残り物のような列に格納します。JSON 内のキーの順序は、Snowflake が共通要素を決定する方法と、クエリのパフォーマンスを最適化する方法に影響します。JSON 内のキーの順序が変更された場合、Snowflake はデータを再解析し、内部ストレージを再編成する必要が生じる可能性があります、その結果、クエリのパフォーマンスが低下する可能性があります。

最近のデータ インポートでは、JSON 値の文字列の長さにばらつきがありました。日付やタイムスタンプなどの非ネイティブ値は、VARIANT 列にロードされるときに文字列として保存されます。

これらの値の操作は、対応するデータ型を使用してリレーショナル列に格納する場合よりも遅くなり、より多くの領域を消費する可能性があります。最近のデータ インポートで JSON 値の文字列長にばらつきがあった場合、Snowflake はより多くのスペースを割り当て、より多くの変換を実行する必要があり、その結果、クエリのパフォーマンスが低下する可能性もあります。

他のオプションは、クエリのパフォーマンス低下の正当な原因ではありません。

最近のデータ インポートに JSON null がありました。Snowflake は、半構造化データの 2 種類の null 値、SQL NULL と JSON null をサポートします。SQL NULL は値が欠落しているか不明であることを意味し、JSON null は値が明示的に null に設定されていることを意味します。Snowflake は、これら 2 種類の null 値を区別し、それに応じて処理できます。最近のデータ インポートに JSON null が含まれていても、クエリのパフォーマンスに大きな影響を与えることはありません。

最近のデータ インポートには、通常よりも少ないフィールドが含まれていました。Snowflake は、さまざまなスキーマとフィールドを持つ半構造化データを処理できます。Snowflake は既存のフィールドに基づいてデータの取り込みとクエリの実行を最適化できるため、最近のデータ インポートで通常よりもフィールドが少なくても、クエリのパフォーマンスに大きな影響を与えることはありません。

参考文献:

VARIANT に格納された半構造化データに関する考慮事項

スノーフレイクアーキテククトレーニング

バリエーション列の一意の要素に対するスノーフレイク クエリのパフォーマンス

スノーフレイクのバリエーションのパフォーマンス

最新問題: 10

データのエクスポート先を制限するには、どの統合オブジェクトを使用する必要がありますか？

- A. ステージ統合
- B. セキュリティの統合
- C. ストレージ統合
- D. API 統合

**Answer: B (メッセージを残す)**

SnowPro Advanced: Architect のドキュメントと学習リソースによると、データのエクスポート先を制限するために使用する必要がある統合オブジェクトはセキュリティ統合です。セキュリティ統合は、Snowflake とサードパーティのセキュリティ サービス (Okta、Duo、Google Authenticator など) との間のインターフェイスを提供する Snowflake オブジェクトです。セキュリティ統合を使用すると、多要素認証 (MFA) を要求したり、エクスポート先を特定のネットワークやドメインに制限したりするなど、データ エクスポートに関するポリシーを適用できます。セキュリティ統合を使用して、Snowflake ユーザーのシングル サインオン (SSO) またはフェデレーション認証を有効にすることもできます<sup>1</sup>。

他のオプションは、データのエクスポート先を制限するために使用できる統合オブジェクトではないため、正しくありません。ステージ統合は Snowflake の統合オブジェクトの有効なタイプではないため、オプション A は不正解です。ステージは、内部ステージ、外部ステージ、名前付きステージなど、データ ファイルが保存されている場所を参照する Snowflake オブジェクトです。ステージは、Snowflake とサードパーティ サービス間のインターフェイスを提供する統合オブジェクトではありません<sup>2</sup>。オプション C は不正解です。ストレージ統合は、Snowflake と外部クラウドストレージ (Amazon S3、Azure Blob Storage、Google Cloud Storage など) との間のインターフェイスを提供する Snowflake オブジェクトです。ストレージ統合を使用すると、資格情報を公開せずに外部クラウドストレージからデータ ファイルに安全にアクセスできますが、データのエクスポート先を制限するために使用することはできません<sup>3</sup>。オプション D は不正解です。API 統合は、Snowflake と、REST API を使用するサードパーティ サービス (Salesforce、Slack、Twilio など) との間のインターフェイスを提供する Snowflake オブジェクトです。API 統合を使用すると、CALL\_EXTERNAL\_API 関数を使用して Snowflake から外部 REST API を安全に呼び出すことができますが、データのエクスポート先を制限するために使用することはできません<sup>4</sup>。参考資料: セキュリティ統合の作成 | Snowflake ドキュメント、CREATE STAGE | Snowflake ドキュメント、ストレージ統合の作成 | Snowflake ドキュメント、API 統合の作成 | スノーフレイクのドキュメント

最新問題: 11

クラウド プロバイダーが提供するメッセージング サービスを使用して、ほぼリアルタイムのデータを読み込むために使用できる取り込み方法は次のどれですか？

- A. Kafka 用の Snowflake コネクタ
- B. スノーフレイク ストリーム
- C. スノーパイプ
- D. スパーク

**Answer: A,C (メッセージを残す)**

Snowflake Connector for Kafka と Snowpipe は、クラウド プロバイダーが提供するメッセージング サービスを使用して、ほぼリアルタイムのデータを読み込むために使用できる 2 つの取り込み方法です。Snowflake Connector

for Kafka を使用すると、Apache Kafka トピックから構造化データおよび半構造化データを Snowflake テーブルにストリーミングできます。

Snowpipe を使用すると、Amazon S3 や Azure Blob Storage などのクラウドストレージの場所に継続的に追加されるファイルからデータを読み込むことができます。どちらの方法でも、Snowflake のマイクロパーティショニングと列指向ストレージを利用して、データの取り込みとクエリのパフォーマンスを最適化します。Snowflake ストリームと Spark は取り込みメソッドではなく、Snowflake アーキテクチャのコンポーネントです。Snowflake ストリームは、テーブル内のデータ変更を追跡することにより、変更データ キャプチャ (CDC) 機能を提供します。Spark は、Snowflake Spark コネクタを使用して大規模なデータを処理し、Snowflake に書き込むために使用できる分散コンピューティング フレームワークです。参考文献:

Kafka 用の Snowflake コネクタ

スノーパイプ

スノーフレイクストリーム

スノーフレイク スパーク コネクタ

#### 最新問題: 12

ある会社には、Data という名前の破損したデータを含むテーブルがあります。同社は、クローン作成とタイム トラベルを使用して、5 分前のデータを復元したいと考えています。

どのようなコマンドでこれを実現できるでしょうか？

A. CREATE CLONE TABLE Recover\_Data FROM Data AT(OFFSET => -60\*5);

B. CREATE CLONE Recover\_Data FROM Data AT(OFFSET => -60\*5);

C. CREATE TABLE Recover\_Data CLONE データ AT(OFFSET => -60\*5);

D. CREATE TABLE データのリカバリ CLONE データ AT(TIME => -60\*5);

**Answer: C (メッセージを残す)**

これは、クローン作成とタイム トラベルを使用して、5 分前のテーブル データのクローンを作成するための正しいコマンドです。クローン作成は、データやメタデータを複製せずに、データベース、スキーマ、テーブル、またはビューのコピーを作成できる機能です。タイム トラベルは、定義された期間内の任意の時点で履歴データ (つまり、変更または削除されたデータ) にアクセスできるようにする機能です。過去のある時点でテーブルのクローンを作成する構文は次のとおりです。

CREATE TABLE <クローン名> CLONE <ソーステーブル> AT (OFFSET => <オフセット秒数>); OFFSET パラメータは、現在時刻との時差を秒単位で指定します。負の値は過去の時点を示します。たとえば、-60\*5 は 5 分前を意味します。あるいは、TIMESTAMP パラメータを使用して、過去の正確なタイムスタンプを指定することもできます。クローンには、指定された時点でソース テーブルに存在していたデータが含まれます<sup>12</sup>。

参考文献:

Snowflake ドキュメント: オブジェクトのクローン作成

Snowflake ドキュメント: 過去のある時点でのオブジェクトのクローン作成

#### 最新問題: 13

ある企業では、さまざまな地域に複数のサイトがあり、そこからデータを取り込みたいと考えています。このタイプのデータ取り込みを可能にするのは次のうちどれですか？

- A. 企業は、そのアカウントにデータを取り込むことができるように、各クラウド リージョンに Snowflake アカウントを持っている必要があります。
- B. 会社は、Snowflake アカウント間でデータをレプリケートする必要があります。
- C. 企業は各サイトに読者アカウントをプロビジョニングし、読者アカウントを通じてデータを取り込む必要があります。
- D. 企業は外部ステージにストレージ統合を使用する必要があります。

**Answer: D (メッセージを残す)**

これは正しい答えです。これにより、企業は外部ステージのストレージ統合を使用して、さまざまなリージョンからデータを取り込むことができるようになります。ストレージ統合は、Snowflake から外部クラウドストレージ内のファイルに安全かつ簡単にアクセスできるようにする機能です。ストレージ統合を使用して、外部ストレージ内のファイルを参照する名前付きの場所である外部ステージを作成できます。外部ステージを使用すると、COPY INTO コマンドを使用して Snowflake テーブルにデータをロードしたり、COPY INTO LOCATION コマンドを使用して Snowflake テーブルからデータをアンロードしたりできます。外部ストレージ サービスが Snowflake12 と互換性がある限り、ストレージ統合は複数のリージョンとクラウド プラットフォームをサポートできます。

参考文献:

Snowflake ドキュメント: ストレージ統合

Snowflake ドキュメント: 外部ステージ

最新問題: 14

CREDITCARDINFO テーブルの CREDICARDND 列にマスキング ポリシーが適用される次のシナリオを考えてみましょう。マスキングポリシーの定義は次のとおりです。

```
create or replace masking policy creditcardno_mask as (val string, snowflake string) ->
case
when is_role_in_session('PI_ANALYTICS') then
right(val, 4)
else '***MASKED***'
end;
```

CREDITCARDINFO テーブルのサンプル データは次のとおりです。

名前 有効期限 クレジットカード番号

ジョン・ドウ 2022-07-23 4321 5678 9012 1234

Snowflake システムのローテに追加の役割が付与されていない場合、結果はどうなりますか?

- A. システム管理者は、CREDICARDND 列のデータをクリア テキストで確認できます。
- B. テーブルの所有者には、CREDICARDND 列のデータがクリア テキストで表示されます。
- C. PI\_ANALYTICS ロールを持つユーザーには、CREDICARDND 列データの最後の 4 文字がテキストで表示されます。
- D. PI\_ANALYTICS ロールを持つユーザーには、CREDICARDND 列が\*\*\* 'MASKED\* \*\*' として表示されます。

**Answer: (解答を表示する)**

画像で定義されているマスキング ポリシーは、ユーザーが PI\_ANALYTICS ロールを持っている場合、CREDITCARDNO 列データの最後の 4 文字をクリア テキストで表示できることを示しています。それ以外の

場合は、MASKED」と表示されます。Snowflake システム ロールには追加のロールが付与されていないため、PI\_ANALYTICS ロールが付与されないため、クレジットカード番号の最後の 4 文字を表示できません。Snowflake の列にマスキング ポリシーを適用するには、ALTER TABLE ... ALTER COLUMN コマンドまたは ALTER VIEW コマンドを使用して、ポリシー名を指定する必要があります。たとえば、CREDITCARDINFO テーブルの CREDITCARDNO 列に Creditcardno\_mask ポリシーを適用するには、次のコマンドを使用できます。ALTER TABLE CREDITCARDINFO ALTER COLUMN CREDITCARDNO SET マスキング ポリシー Creditcardno\_mask; Snowflake でマスキング ポリシーを作成および使用方法の詳細については、次のリソースを参照してください。

CREATE MASKING POLICY: このドキュメントでは、新しいマスキング ポリシーを作成したり、既存のマスキング ポリシーを置き換えたりできる CREATE MASKING POLICY コマンドの構文と使用方法について説明します。動的データ マスキングの使用: このガイドでは、Snowflake で動的データ マスキングを構成および使用方法について説明します。これは、ユーザーの実行コンテキストに基づいて機密データをマスクできる機能です。ALTER MASKING POLICY: このドキュメントでは、既存のマスキング ポリシーのプロパティを変更できる ALTER MASKING POLICY コマンドの構文と使用方法について説明します。

参考文献: 1: <https://docs.snowflake.com/en/sql-reference/sql/create-masking-policy> 2: <https://docs.snowflake.com/en/user-guide/security-column-ddm-use> 3: <https://docs.snowflake.com/en/sql-reference/sql/alter-masking-policy>

#### 最新問題: 15

グローバル企業は、Snowflake アカウントを使用して、販売データと在庫データをベンダーと安全に共有する必要があります。

同社は、AWS eu-west 2 Europe (London) リージョンに Snowflake アカウントを持っています。ベンダーの Snowflake アカウントは、西ヨーロッパ リージョンの Azure プラットフォーム上にあります。会社のアーキテクトはデータ共有をどのように構成する必要がありますか？

- A. 1. 共有を作成します。
- 2. オブジェクトを共有に追加します。
- 3. ベンダーがアクセスできるように、コンシューマ アカウントを共有に追加します。
- B. 1. 共有を作成します。
- 2. ベンダーが使用する読者アカウントを作成します。
- 3. リーダーアカウントを共有に追加します。
- C. 1. db\_share という新しいロールを作成します。
- 2. db\_share ロールに、企業データベースおよびスキーマからデータを読み取る権限を付与します。
- 3. ベンダーのユーザーを作成します。
- 4. ベンダーのユーザーに ds\_share ロールを付与します。
- D. 1. 会社のローカル アカウント内の既存のデータベースをプライマリに昇格させます。
- 2. 西ヨーロッパ リージョンの Azure 上の Snowflake にデータベースをレプリケートします。
- 3. 共有を作成し、その共有にオブジェクトを追加します。
- 4. ベンダーがアクセスできるように、コンシューマ アカウントを共有に追加します。

**Answer: A** ([メッセージを残す](#))

別のクラウド プラットフォームおよびリージョン上の Snowflake アカウントを使用してベンダーとデータを安全に共有する正しい方法は、共有を作成し、その共有にオブジェクトを追加し、ベンダーがアクセスできるようにコンシューマー アカウントを共有に追加することです。このようにして、企業は、どのデータを共有するか、誰がデータにアクセスできるか、共有の有効期間を制御できます。ベンダーは、共有データを自分のアカウントにコピーしたり移動したりせずに、そのデータをクエリできるようになります。他のオプションは、不必要な読者アカウント、ユーザー、ロール、データベース レプリケーションの作成を伴うため、不正確であるか非効率的です。  
<https://learn.snowflake.com/en/certifications/snowpro-advanced-architect/>

#### 最新問題: 16

アーキテクトは次のコマンドを順番に入力しました。

```
CREATE DATABASE SANDBOX;  
CREATE ROLE INTERN;  
CREATE TABLE SANDBOX.PUBLIC.AGENDA (ID INT, ITEMS STRING);  
GRANT SELECT ON ALL TABLES IN SCHEMA SANDBOX.PUBLIC TO ROLE INTERN;  
GRANT ROLE INTERN TO USER USER1;
```

USER1 がテーブルを見つけることができません。

最小特権の原則を使用してテーブルを検索するには、アーキテクトが USER1 に対して実行する必要があるコマンドは次のどれですか? (2つお選びください。)

- A. ロールパブリックをロールインターンに付与します。
- B. ロール インターンにデータベース サンドボックスの使用を許可します。
- C. SCHEMA Sandbox.PUBLIC の使用をロールインターンに許可します。
- D. データベース サンドボックスの所有権をユーザー インターンに付与します。
- E. データベース サンドボックスに関するすべての権限をインターンの役割に付与します。

**Answer:** ([解答を表示する](#))

最小権限の原則に従って、アーキテクトは USER1 が SANDBOX データベース内のテーブルを検索するために必要な最小限の権限を付与する必要があります。

USER1 が PUBLIC スキーマ内のテーブルにアクセスできるようにするには、SANDBOX データベースおよび SANDBOX.PUBLIC スキーマに対する USAGE 権限が必要です。したがって、コマンド B と C は実行するのに正しいコマンドです。

コマンド A は不正解です。PUBLIC ロールはアカウント内のすべてのユーザーとロールに自動的に付与され、デフォルトでは SANDBOX データベースに対する権限がありません。

コマンド D は不正解です。これは、SANDBOX データベースの所有権をアーキテクトから USER1 に譲渡することになりますが、これは必要なく、最小特権の原則に違反します。

コマンド E は不正解です。このコマンドは、SANDBOX データベース上で可能なすべての権限を USER1 に付与することになりますが、これも必要ではなく、最小権限の原則に違反します。

参考資料: : Snowflake - 最小特権の原則 : Snowflake - アクセス制御特権 : Snowflake - パブリックロール : Snowflake - 所有権と付与

有効な **ARA-R01** 問題集は GoShiken.com が提供された合格しやすい ARA-R01 試験問題集！ GoShiken.com が最新の **ARA-R01** 試験問題集を提供しています。GoShiken.com ARA-R01 試験問題は最新で、解答が正確でございます。最新の GoShiken.com ARA-R01 問題集をゲットする人はこちら：

<https://www.goshiken.com/Snowflake/ARA-R01-mondaishu.html> (16330%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

#### 最新問題: 17

ある会社では、8 つのテーブルと 5 つの安全なビューで設定された受信共有を持っています。同社は、このシェアを本番データパイプラインの一部にすることを計画している。

インバウンドシェアに対して企業はどのようなアクションを取れるでしょうか? (2つお選びください。)

- A. 共有からテーブルのクローンを作成します。
- B. 共有に対する変更権限を付与します。
- C. 共有データベースからテーブルを作成します。
- D. 共有データベース内に追加のビューを作成します。
- E. 共有テーブルにテーブル ストリームを作成します。

**Answer: A,D** ([メッセージを残す](#))

Snowflake のドキュメントと Web 検索結果によると、これら 2 つのアクションは受信共有で可能です。インバウンド共有は、別の Snowflake アカウント (プロバイダー) によって作成され、自分のアカウント (コンシューマー) にインポートされる共有です。受信共有を使用すると、プロバイダーによって共有されたデータにアクセスできませんが、変更または削除することはできません。ただし、受信共有を使用して次のような一部のアクションを実行できます。

共有からテーブルのクローンを作成します。CREATE TABLE ... CLONE ステートメントを使用して、受信共有からテーブルのコピーを作成できます。クローンには元のテーブルと同じデータとメタデータが含まれますが、共有からは独立しています。クローンは必要に応じて変更または削除できますが、provider1 によって元のテーブルに加えられた変更は反映されません。

共有データベース内に追加のビューを作成します。CREATE VIEW ステートメントを使用して、テーブル上にビューを作成したり、受信共有からビューを作成したりできます。ビューは共有データベースに保存されますが、それらはあなたのアカウントによって所有されます。アカウント内の他のビューをクエリするのと同じようにビューをクエリできますが、share2 の基になるオブジェクトを変更したり削除したりすることはできません。リストされている他のアクションは、コンシューマーにとって読み取り専用である共有または共有オブジェクトを変更する必要があるため、受信共有では実行できません。共有に対する変更権限を付与したり、共有データベースからテーブルを作成したり、共有テーブルにテーブル ストリームを作成したりすることはできません<sup>34</sup>。

参考文献:

共有からのオブジェクトのクローン作成 | スノーフレイクのドキュメント

共有データのビューの作成 | スノーフレイクのドキュメント

共有からのデータのインポート | スノーフレイクのドキュメント

共有テーブル上のストリーム | スノーフレイクのドキュメント

#### 最新問題: 18

データ変換クエリを通じて内部ステージから Snowflake テーブルに CSV 形式のデータをロードする次の COPY コマンドについて考えてみましょう。

```
copy into home_sales(city, zip, sale_date, price)
from (select t.$1, t.$2, t.$6, t.$7 from @mystage/sales.csv.qz t)
file_format =
(
format_name = mycsvformat
empty_field_as_null = true
field_optionally_enclosed_by = ''
)
validation_mode - return_all_errors
;
```

このコマンドを実行すると、次のエラーが発生します。

SQL コンパイル エラー: 無効なパラメーター 'validation\_mode'

構文が正しいと仮定すると、このエラーの原因は何でしょうか？

- A. VALIDATION\_MODE パラメーターは、外部ステージからのみデータをロードする COPY ステートメントをサポートします。
- B. VALIDATION\_MODE パラメータは、CSV ファイル形式の COPY ステートメントをサポートしません。
- C. VALIDATION\_MODE パラメータは、ロード中にデータを変換する COPY ステートメントをサポートしません。
- D. オプション VALIDATION\_MODE の値 return\_all\_errors によりコンパイル エラーが発生しています。

**Answer:** ([解答を表示する](#))

VALIDATION\_MODE パラメータは、データをテーブルにロードするときの COPY ステートメントの動作を指定するために使用されます。これは、ファイル内の行が無効な場合に COPY ステートメントがエラーを返すか、有効な行のロードを続行するかを指定するために使用されます。VALIDATION\_MODE パラメータは、外部ステージからデータをロードする COPY ステートメントでのみサポートされます<sup>1</sup>。

質問内のクエリでは、データ変換クエリを使用して内部ステージからデータをロードします。データ変換クエリは、JSON または XML データの解析、関数の適用、他のテーブルとの結合など、ロード プロセス中にデータを変換するクエリです<sup>2</sup>。

ドキュメントによると、VALIDATION\_MODE はロード中にデータを変換する COPY ステートメントをサポートしていません。パラメータが指定されている場合、COPY ステートメントはエラーを返します<sup>1</sup>。

したがって、選択肢 C が正解です。

参照: : COPY INTO <table> : ロード中のデータの変換

**最新問題: 19**

外部テーブルに対して実行されるクエリのパフォーマンスを向上させる手順は何ですか？

- A. 外部テーブルをパーティション化します。
- B. ソース ファイルの名前を短縮します。
- C. ソース ファイルの文字エンコーディングを UTF-8 に変換します。
- D. 外部ステージの代わりに内部ステージを使用してソース ファイルを保存します。

**Answer: A** ([メッセージを残す](#))

外部テーブルのパーティション化は、スキャンされるデータ量を削減することで、テーブルに対して実行されるクエリのパフォーマンスを向上させる手法です。外部テーブルのパーティション化には、列の値に基づいてテーブルをデータのサブセットに論理的に分割する方法を定義する 1 つ以上のパーティション列の作成が含まれます。パーティション列は、ファイルのメタデータ (ファイル名、パス、サイズ、変更時刻など) またはファイルのコンテンツ (列値や JSON 属性など) から派生できます。外部テーブルをパーティション化すると、クエリ オプティマイザがクエリ述語に一致しないファイルを削除できるため、不必要なデータのスキャンと処理が回避されます<sup>2</sup>。他のオプションは、外部テーブルに対して実行されるクエリのパフォーマンスを向上させる効果的な手順ではありません。ソースファイルの名前を短くします。ファイル名はクエリ処理に使用されないため、このオプションはクエリのパフォーマンスには影響しません。ファイル名は、外部テーブルの作成とクエリ結果の表示にのみ使用されます<sup>3</sup>。ソース ファイルの文字エンコーディングを UTF-8 に変換します。Snowflake は外部テーブル ファイルのさまざまな文字エンコーディング (UTF-8、UTF-16、UTF-32、ISO-8859-1、Windows-1252 など) をサポートしているため、このオプションはクエリのパフォーマンスには影響しません。Snowflake は、ファイルの文字エンコーディングを自動的に検出し、クエリ処理のために内部で UTF-8 に変換します<sup>4</sup>。ソース ファイルを保存するには、外部ステージの代わりに内部ステージを使用します。外部テーブルは、Amazon S3、Google Cloud Storage、Azure Blob Storage などの外部ステージに保存されているファイルのみを参照できるため、このオプションは適用されません。内部ステージは、外部テーブルではなく内部テーブルにデータをロードするために使用されます<sup>5</sup>。

1: SnowPro Advanced: アーキテクト | 学習ガイド

2: スノーフレイクのドキュメント | 外部テーブルのパーティション化

3: スノーフレイクのドキュメント | 外部テーブルの作成

4: スノーフレイクのドキュメント | ステージングされたデータ ファイルでサポートされるファイル形式と圧縮

5: スノーフレイクのドキュメント | ステージ概要

: SnowPro Advanced: アーキテクト | 学習ガイド

: 外部テーブルのパーティション化

: 外部テーブルの作成

: ステージングされたデータ ファイルでサポートされているファイル形式と圧縮

: ステージ概要

## 最新問題: 20

Snowflake Architect は新しいデータ共有を作成し、安全なビュー内の特定のレコードのみがデータ共有内でコンシューマに表示されることを確認したいと考えています。

消費者がデータにアクセスできることを検証するための推奨される方法は何ですか？

**A.** 以下に示すようにリーダー アカウントを作成し、資格情報を使用してログインすることでコンシューマになりすまします。

管理対象アカウントを作成します Reader\_acctl admin\_name = user1、admin\_password 'Sdfed43da!44T'、type = Reader;

**B.** 以下に示すように行アクセス ポリシーを作成し、データ共有に割り当てます。

行アクセス ポリシーの作成または置換 rap\_acct as (acct\_id varchar) はブール値を返します -> case when 'acctl\_role' = current\_role() then true else false end;

**C.** コンシューマ アカウントを偽装するには、SIMULATED\_DATA\_SHARING\_CONSUMER というセッションパラメータを次のように設定します。

セッションセットの変更 Simulated\_data\_sharing\_consumer - 'Consumer Acctl'

D. 特定の消費者アカウントになりすますには、以下に示すように共有設定を変更します。

シェアの変更販売シェアセットアカウント = 'Consumer1' シェア制限 = true

**Answer: C (メッセージを残す)**

SIMULATED\_DATA\_SHARING\_CONSUMER セッション パラメーターを使用すると、データ プロバイダーは、リーダー アカウントを作成したり、消費者の資格情報でログインしたりせずに、消費者アカウントのデータ アクセスをシミュレートできます。このパラメータは、特に現在のアカウントまたはロールに基づいてデータをフィルタリングするセキュア ビューまたはセキュア UDF を使用する場合に、データ共有内のコンシューマによるデータアクセスを検証するために使用できます。このパラメーターをコンシューマー アカウントの名前に設定すると、データ プロバイダーは、コンシューマーが共有データベースにクエリを実行するときに表示されるデータと同じデータを表示できます。これは、データ共有機能をテストし、意図したデータのみが消費者に表示されるようにするための便利で効率的な方法です。

参考文献:

SIMULATED\_DATA\_SHARING\_CONSUMER セッション パラメータの使用

SnowPro Advanced: アーキテクト試験学習ガイド

最新問題: 21

以下の SQL ステートメントではどのような変換がサポートされていますか? (3 つを選択)。

パイプを作成 ... コピーとして ... から (...)

- A. データはオプションの where 句でフィルタリングできます。
- B. 列の順序を変更できます。
- C. 列は省略できます。
- D. 型キャストがサポートされています。
- E. 受信データを他のテーブルと結合できます。
- F. ON ERROR - ABORT ステートメント コマンドを使用できます。

**Answer: A,B,C (メッセージを残す)**

SQL ステートメントは、Snowflake でパイプを作成するためのコマンドです。これは、Snowpipe が取り込みキューから tables1 にデータをロードするために使用する COPY INTO <table> ステートメントを定義するオブジェクトです。このステートメントは、FROM 句でサブクエリを使用して、ステージングされたファイルのデータを table2 にロードする前に変換します。

サブクエリでサポートされている変換は次のとおりです2:

データはオプションの WHERE 句によってフィルタリングできます。これは、サブクエリによって返された行が満たす必要がある条件を指定します。例えば:

SQLAI によって生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パイプの作成

コピーイントミーテーブル

から (

```
select*from@mystage
```

```
wherecol1='A'andcol2>10
```

```
);
```

列は並べ替えることができます。これは、サブクエリ内の列の順序を変更して、ターゲット テーブル内の列の順序と一致させることを意味します。例えば：

SQLAI によって生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パイプの作成

```
copyintomytable (col1, col2, col3)
```

から (

```
selectcol3, col1, col2from@mystage
```

);

列は省略できます。これは、ターゲットテーブルで必要のない一部の列をサブクエリから除外することを意味します。例えば：

SQLAI によって生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パイプの作成

```
copyintomytable (col1, col2)
```

から (

```
selectcol1, col2from@mystage
```

);

他のオプションはサブクエリではサポートされていません。理由 2:

型キャストはサポートされていません。これは、サブクエリ内の列のデータ型を変更することを意味します。

たとえば、次のステートメントではエラーが発生します。

SQLAI によって生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パイプの作成

```
copyintomytable (col1, col2)
```

から (

```
selectcol1::date, col2from@mystage
```

);

受信データを他のテーブルと結合することはできません。これは、ステージングされたファイルのデータをサブクエリ内の別のテーブルのデータと結合することを意味します。たとえば、次のステートメントではエラーが発生します。

SQLAI によって生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パイプの作成

```
copyintomytable (col1, col2, col3)
```

から (

```
selects.col1, s.col2, t.col3from@mystages
```

```
joinothertable tons.col1=t.col1
```

);

ON ERROR - ABORT ステートメント コマンドは使用できません。つまり、エラーが発生した場合はロード操作全体が中止されます。このコマンドは COPY INTO <table> ステートメントでのみ使用でき、サブクエリでは使用できません。たとえば、次のステートメントではエラーが発生します。

SQLAI によって生成されたコード。慎重に確認して使用してください。詳細については FAQ をご覧ください。

パイプの作成

```
コピーイントミーテーブル  
から (  
select*from@mystage  
エラーによる中止  
);
```

参考文献:

- 1: パイプを作成する | スノーフレークのドキュメント
- 2: ロード中のデータの変換 | スノーフレークのドキュメント

### 最新問題: 22

Snowflake Architect は、災害復旧計画をサポートするためにデータベース レプリケーションを設定しています。プライマリ データベースには外部テーブルがあります。データベースはどのように複製する必要がありますか？

- A. プライマリ データベースのクローンを作成し、データベースをレプリケートします。
- B. 外部テーブルをレプリケートされていないデータベースに移動し、プライマリ データベースをレプリケートします。
- C. データベースをレプリケートし、レプリケートされたデータベースが外部テーブルと同じリージョンにあることを確認します。
- D. プライマリ データベースを、データベースのレプリケート先と同じリージョン内のアカウントと共有します。

**Answer: B** ([メッセージを残す](#))

データベース レプリケーションは、災害復旧やビジネス継続性を目的として、別のアカウント、リージョン、またはクラウド プラットフォームにデータベースのコピーを作成できる機能です。ただし、すべてのデータベース オブジェクトを複製できるわけではありません。外部テーブルは、Snowflake の一部ではない外部ステージに格納されたデータ ファイルを参照するため、例外の 1 つです。したがって、外部テーブルを含むデータベースをレプリケートするには、外部テーブルをレプリケートされていない別のデータベースに移動してから、他のオブジェクトを含むプライマリ データベースをレプリケートする必要があります。こうすることで、レプリケーション エラーを回避し、プライマリ データベースとセカンダリ データベース間の整合性を確保できます。他のオプションは、外部テーブルの問題に対処していないか、Snowflake でサポートされていない代替方法を使用しているため、正しくありません。レプリケーションは元のデータベースに対してのみ機能し、そのクローンに対しては機能しないため、プライマリ データベースのクローンを作成してからそれをレプリケートすることはできません。また、共有はデータベースのコピーを作成するのではなく、共有オブジェクトへのアクセスを許可する別の機能であるため、プライマリ データベースを別のアカウントと共有することもできません。最後に、ステージ URL が有効でアクセス可能である限り、外部テーブルは任意のリージョンまたはクラウド プラットフォームに保存されているデータ ファイルにアクセスできるため、レプリケートされたデータベースが外部テーブルと同じリージョンにあることを確認する必要はありません。参考文献:

[レプリケーションとフェイルオーバー/フェイルバック] 1

[外部テーブルの概要] 2

[外部テーブルの操作] 3

[レプリケーション: Snowflake の 1 つのクラウド プラットフォームまたはリージョンから別のクラウド プラットフォームまたはリージョンにアカウントを移行する方法] 4

### 最新問題: 23

既存のクラスター キーを持つテーブルの代替クラスター キーを定義する機能を提供する機能はどれですか？

- A. 外部テーブル
- B. マテリアライズド ビュー
- C. 検索の最適化
- D. 結果キャッシュ

**Answer:** ([解答を表示する](#))

マテリアライズド ビューは、既存のクラスター キーを持つテーブルの代替クラスター キーを定義する機能を提供します。マテリアライズド ビューは、Snowflake に保存される事前計算された結果セットであり、通常のテーブルと同様にクエリを実行できます。マテリアライズド ビューには、ベース テーブルとは異なるクラスター キーを含めることができます。これにより、マテリアライズド ビューでのクエリのパフォーマンスと効率が向上します。マテリアライズド ビューは、ベース テーブル データの集計、結合、およびフィルターもサポートできます。AUTO\_REFRESH パラメータが true<sup>1</sup> に設定されている限り、ベース テーブル内の基になるデータが変更されると、マテリアライズド ビューが自動的にリフレッシュされます。

参考文献:

マテリアライズド ビュー | スノーフレイクのドキュメント

### 最新問題: 24

アーキテクトは、ORDER\_ADMIN ユーザーのグループに、テーブルに対する権限を付与せずに、ORDERS テーブル内の古いデータをクリーンアップする (5 年より古いすべてのレコードを削除する) 権限を付与する必要があります。グループのマネージャー (ORDER\_MANAGER) は、テーブルに対する完全な DELETE 権限を持っています。ORDER\_MANAGER ロールが保持する DELETE 権限を必要とせずに、ORDER\_ADMIN ロールを有効にしてこのデータ クリーンアップを実行するにはどうすればよいですか？

- A. 適切な 「5 years」ビジネス ロジックを含む、呼び出し元の権限で実行されるストアード プロシージャを作成し、このプロシージャの USAGE を ORDER\_ADMIN に付与します。ORDER\_MANAGER ロールはプロシージャを所有します。
- B. 呼び出し元と所有者の両方の権限を使用して実行できるストアード プロシージャを作成し (ユーザーは実行時に使用する権限を指定できます)、このプロシージャの USAGE を ORDER\_ADMIN に付与します。ORDER\_MANAGER ロールはプロシージャを所有します。
- C. 適切な 「5 年」ビジネス ロジックを含む、所有者の権限で実行されるストアード プロシージャを作成し、このプロシージャの USAGE を ORDER\_ADMIN に付与します。ORDER\_MANAGER ロールはプロシージャを所有します。
- D. このシナリオは実際には Snowflake では不可能です。テーブルに対して DELETE を実行するユーザーには、使用しているロールに DELETE 権限が付与されている必要があります。

**Answer: C** ([メッセージを残す](#))

これは正しい答えです。これにより、ORDER\_ADMIN ロールは、ORDERS テーブルに対する DELETE 権限を必要とせずにデータ クリーンアップを実行できるようになります。ストアード プロシージャは、Snowflake で SQL ステートメントまたはストアード プロシージャをスケジュールして実行できる機能です。ストアード プロシージャは、

呼び出し元の権限または所有者の権限のいずれかで実行できます。呼び出し元の権限のストアド プロシージャは、ストアド プロシージャを呼び出したロールの権限で実行されますが、所有者の権限のストアド プロシージャは、ストアド プロシージャを作成したロールの権限で実行されます。所有者の権限で実行されるストアド プロシージャを作成すると、ORDER\_MANAGER ロールは、ORDER\_ADMIN ロールに ORDERS テーブルに対するより一般的な権限を付与しなくても、古いデータを削除する特定のタスクを ORDER\_ADMIN ロールに委任できます。ストアド プロシージャには、5 年より古いレコードのみを削除する適切なビジネス ロジックが含まれている必要があります。ORDER\_MANAGER ロールはストアド プロシージャに対する USAGE 権限を ORDER\_ADMIN ロールに付与する必要があります。その後、ORDER\_ADMIN ロールはストアド プロシージャを実行してデータ クリーンアップ 12 を実行できます。

参考文献:

Snowflake ドキュメント: ストアド プロシージャ

Snowflake ドキュメント: 呼び出し元の権利と所有者の権利ストアド プロシージャについて

### 最新問題: 25

アーキテクトは、いくつかのオブジェクトを回復するために、1 週間前の 6 月 1 日火曜日午前 8 時にスキーマ STAGING のクローンを作成するように依頼されました。

STAGING スキーマの保存期間は 50 日間です。

アーキテクトは次のステートメントを実行します。

CREATE SCHEMA STAGING\_CLONE CLONE STAGING at (timestamp => '2021-06-01 08:00:00'); アーキテクトは次のエラーを受け取ります: タイム トラベル データはスキーマ STAGING では使用できません。要求された時刻は、許可されたタイムトラベル期間を超えているか、オブジェクトの作成時刻より前です。

次に、アーキテクトはスキーマ履歴をチェックし、次のことを確認します。

```
CREATED_ON|NAME|DROPPED_ON
```

```
2021-06-02 23:00:00 | ステージング | ヌル
```

```
2021-05-01 10:00:00 | ステージング | 2021-06-02 23:00:00
```

STAGING スキーマのクローンを作成するにはどうすればよいですか?

**A.** STAGING スキーマの削除を解除し、CLONE ステートメントを再実行します。

**B.** ステートメントを変更します: CREATE SCHEMA STAGING\_CLONE CLONE STAGING at (timestamp => '2021-05-01 10:00:00');

**C.** STAGING スキーマの名前を変更し、UNDROP を実行して以前の STAGING スキーマのバージョンを取得してから、CLONE ステートメントを実行します。

**D.** 提案されたタイム トラベル期間中に STAGING スキーマ バージョンがアクティブではなかったため、クローン作成を実行できません。

**Answer: C** ([メッセージを残す](#))

このエラー メッセージは、スキーマの現在のバージョンが 2021-06-02 23:00:00 (タイムスタンプ 2021-06 より後) に作成されたため、スキーマ STAGING には要求されたタイムスタンプに使用できるタイム トラベル データがないことを示します。-01 08:00:00。したがって、CLONE ステートメントは、その時点でのスキーマの履歴データにアクセスできません。

オプション A は不正解です。STAGING スキーマを削除しても、2021-06-01 08:00:00 にアクティブだった以前のバージョンのスキーマは復元されません。代わりに、同じ名前でデータやオブジェクトのない新しいバージョンのスキーマが作成されます。

オプション B は不正解です。タイムスタンプを 2021-05-01 10:00:00 に変更すると、スキーマは 1 週間前の状態では複製されず、最初に作成された状態で複製されます。これは、スキーマとそのオブジェクトの望ましい状態を反映していない可能性があります。

オプション C は正解です。ステー징 スキーマの名前を変更し、UNDROP を実行して以前のステーディング スキーマ バージョンを取得すると、2021 年 6 月 2 日に削除されたスキーマが復元されます。

23:00:00。このスキーマには、要求されたタイムスタンプ 2021-06-01 で利用可能なタイムトラベル データがあります。

08:00:00 であり、CLONE ステートメントを使用してクローンを作成できます。

選択肢 D は不正解です。UNDROP コマンドを使用して、提案されたタイムトラベル期間中にアクティブだった以前のバージョンのスキーマにアクセスすることでクローンを作成できます。

参考資料: : クローン作成に関する考慮事項 : タイムトラベルの理解と使用 : CREATE <object> ... CLONE

#### 最新問題: 26

ユーザーは次のどのロールを使用してオブジェクト パラメータを変更できますか?

- A. アカウント管理者、セキュリティ管理者
- B. システム管理者、セキュリティ管理者
- C. ACCOUNTADMIN、特権を持つユーザー
- D. SECURITYADMIN、特権を持つユーザー

**Answer:** ([解答を表示する](#))

Snowflake のドキュメントによると、オブジェクト パラメーターは、データベース、スキーマ、テーブル、ステージなどの個々のオブジェクトに設定できるパラメーターです。オブジェクトパラメータは、オブジェクトに対する適切な権限を持つユーザーが設定できます。たとえば、テーブルにオブジェクト パラメータ AUTO\_REFRESH を設定するには、ユーザーはテーブルに対する MODIFY 権限を持っている必要があります。ACCOUNTADMIN ロールは、アカウント内のすべてのオブジェクトに対して最高レベルの権限を持っているため、任意のオブジェクトに対して任意のオブジェクト パラメータを設定できます。ただし、SECURITYADMIN や SYSADMIN などの他のロールは、すべてのオブジェクトに対して同じレベルの権限を持っていないため、所有していないオブジェクトや必要な権限を持っていないオブジェクトに対してオブジェクト パラメータを設定することはできません。

したがって、正解は C. ACCOUNTADMIN、PRIVILEGE を持つ USER です。

参考文献:

[パラメータ | スノーフレークのドキュメント](#)

[オブジェクトパラメータ | スノーフレークのドキュメント](#)

[オブジェクトの権限 | スノーフレークのドキュメント](#)

#### 最新問題: 27

メディア企業には、顧客レビュー データを Snowflake テーブルに取り込み、いくつかの変換を適用するデータ パイプラインが必要です。同社はまた、Amazon Comprehend を使用してセンチメント分析を行い、地域ごとに異なる

るクラウドプロバイダーを使用する広告会社が匿名化された最終データセットを公開できるようにする必要もあります。

データ パイプラインは、イベント通知を利用して新しいレコードがオブジェクトストレージに到着するたびに、継続的に効率的に実行する必要があります。また、運用の複雑さ、プラットフォームのアップグレードやセキュリティを含むインフラストラクチャのメンテナンス、および開発の労力は最小限に抑える必要があります。

これらの要件を満たすのはどの設計ですか？

**A.** COPY INTO を使用してデータを取り込み、ストリームとタスクを使用して変換を調整します。データを Amazon S3 にエクスポートして、Amazon Comprehend でモデル推論を行い、データを Snowflake テーブルに取り込みます。次に、Snowflake Marketplace にリストを作成して、他の企業がデータを利用できるようにします。

**B.** Snowpipe を使用してデータを取り込み、ストリームとタスクを使用して変換を調整します。Amazon Comprehend でモデル推論を実行し、最終レコードを Snowflake テーブルに書き込む外部関数を作成します。次に、Snowflake Marketplace にリストを作成して、他の企業がデータを利用できるようにします。

**C.** Amazon EMR を使用して Snowflake にデータを取り込み、Snowflake Spark コネクタを使用して PySpark にデータを取り込みます。

別の Spark ジョブを使用して変換を適用します。Amazon Comprehend テキスト分析 API を利用してモデル推論を行う Python プログラムを開発します。次に、結果を Snowflake テーブルに書き込み、Snowflake Marketplace にリストを作成して、他の企業がデータを利用できるようにします。

**D.** Snowpipe を使用してデータを取り込み、ストリームとタスクを使用して変換を調整します。データを Amazon S3 にエクスポートして、Amazon Comprehend でモデル推論を行い、データを Snowflake テーブルに取り込みます。次に、Snowflake Marketplace にリストを作成して、他の企業がデータを利用できるようにします。

**Answer: B** ([メッセージを残す](#))

この設計は、データ パイプラインのすべての要件を満たしています。Snowpipe は、イベント通知を使用してオブジェクトストレージから Snowflake への継続的なデータの読み込みを可能にする機能です。効率的でスケラブルでサーバーレスです。つまり、ユーザーによるインフラストラクチャやメンテナンスは必要ありません。ストリームとタスクは、変更データのキャプチャとスケジュールされた実行を使用して、Snowflake 内で自動化されたデータ パイプラインを可能にする機能です。

また、効率的でスケラブルでサーバーレスでもあり、データ変換プロセスを簡素化します。外部関数は、Snowflake 内から外部サービスまたは API を呼び出すことができる関数です。これらを使用して、Amazon Comprehend と統合し、データのセンチメント分析を実行できます。結果は、標準の SQL コマンドを使用して Snowflake テーブルに書き戻すことができます。Snowflake Marketplace は、データ プロバイダーがさまざまなアカウント、リージョン、クラウド プラットフォームにわたってデータ コンシューマーとデータを共有できるようにするプラットフォームです。これは、他の企業がデータを公開できるようにする安全かつ簡単な方法です。

参考文献:

[スノーパイプの概要 | スノーフレイクのドキュメント](#)

[データ パイプラインの概要 | スノーフレイクのドキュメント](#)

[外部関数の概要 | スノーフレイクのドキュメント](#)

[Snowflake データ マーケットプレイスの概要 | スノーフレイクのドキュメント](#)

CURRENT\_TIME () または CURRENT\_TIMESTAMP() のいずれかのデフォルト値を持つ列のロード時間をキャプチャするテーブルにデータをロードすると、何が起こりますか？

- A. 特定の COPY ステートメントを使用してロードされたすべての行には、行が挿入された時間に基づいて異なるタイムスタンプが設定されます。
- B. 特定の COPY ステートメントを使用してロードされた行には、行がソースから読み取られた時間に基づいて異なるタイムスタンプが設定されます。
- C. 特定の COPY ステートメントを使用してロードされた行には、ソース内で行が作成されたときに基づいて異なるタイムスタンプが設定されます。
- D. 特定の COPY ステートメントを使用してロードされたすべての行は、同じタイムスタンプ値を持ちます。

**Answer: D (メッセージを残す)**

Snowflake のドキュメントによると、CURRENT\_TIME () または CURRENT\_TIMESTAMP() のいずれかのデフォルト値を持つ列のロード時間をキャプチャするテーブルにデータをロードする場合、デフォルト値は行ごとではなく、COPY ステートメントごとに 1 回評価されます。したがって、特定の COPY ステートメントを使用してロードされたすべての行は、同じタイムスタンプ値を持ちます。この動作により、タイムスタンプ値には、データがソースから読み取られたときやソース内で作成されたときではなく、データがテーブルにロードされたときの時刻が反映されるようになります。

参考文献:

Snowflake ドキュメント: デフォルト値を使用したテーブルへのデータのロード

Snowflake ドキュメント: COPY INTO テーブル

### 最新問題: 29

水の使用量を測定する IoT デバイス用のテーブルを作成します。テーブルはすぐに大きくなり、20 億行を超える行が含まれます。

```
create table water_iot (  
  UniqueId number,  
  DeviceId varchar(20),  
  DeviceManufacturer varchar(50)  
  CustomerId varchar(20),  
  IOT_timestamp timestamp_ntz,  
  City varchar(80),  
  Location varchar(50)  
)
```

テーブルの一般的なクエリ パターンは次のとおりです。

1. DeviceId、IOT\_timestamp、および CustomerId は、select ステートメントのフィルター述語で頻繁に使用されます
2. City 列と DeviceManufacturer 列がよく取得されます
3. UniqueId が頼りになることが多い

クラスタリング キーにはどのフィールドを使用する必要がありますか？

- A. IOT\_timestamp
- B. 都市およびデバイス製造者
- C. DeviceId と CustomerId
- D. 固有の値

**Answer: C** ([メッセージを残す](#))

クラスタリング キーは、Snowflake のストレージの単位である同じマイクロパーティションにデータを同じ場所に配置するために使用される列または式のサブセットです。クラスタリングにより、スキャンする必要があるデータの量が削減されるため、クラスタリング キー列でフィルタリングするクエリのパフォーマンスが向上します。クラスタリング キーの最適な選択は、クエリ パターンとテーブル内のデータ分布によって異なります。この場合、列 DeviceId、IOT\_timestamp、および CustomerId は、select ステートメントのフィルター述語で頻繁に使用されません。つまり、これらの列がクラスタリング キーの適切な候補であることを意味します。City 列と DeviceManufacturing 列は頻繁に取得されますが、フィルター処理されないため、クラスタリング キーにとってはそれほど重要ではありません。

Uniqueld 列はカウントに使用されますが、カーディナリティが高く均一な分布を持つ可能性が高いため、クラスタリング キーとしては適切な選択ではありません。つまり、データを同じ場所に配置するのに役立ちません。したがって、マイクロパーティションをプルーニングしてクエリを高速化するのに役立つため、DeviceId と CustomerId をクラスタリング キーとして使用するのが最善のオプションです。参考文献: クラスタリング キーとクラスタ化テーブル、マイクロパーティションとデータ クラスタリング、スノーフレイク クラスタリングの完全ガイド

**最新問題: 30**

Snowflake Architect は、強力な法的分離ルールとマルチテナントが要件となる組織向けに、アプリケーションとテナント戦略を設計しています。

ロールベースのアクセス ポリシー (RBAC) がテナントを分離するための実行可能なオプションである場合、これらの要件を満たすのはどのアプローチですか？

- A. Snowflake 組織内の各テナントのアカウントを作成します。
- B. 行レベルのセキュリティがテナントの分離に有効な場合は、テナント戦略ごとにオブジェクトを作成します。
- C. テナントを分離するために行レベルのセキュリティが実行できない場合は、テナント戦略ごとにオブジェクトを作成します。
- D. テナントを分離するために行レベルのセキュリティが実行できない場合は、マルチテナント テーブル戦略を作成します。

**Answer: (**[解答を表示する](#)**)**

このアプローチは、強力な法的分離とマルチテナンシーの要件を満たします。テナントごとに個別のアカウントを作成することで、アプリケーションは各テナントが独自の専用ストレージ、コンピューティング、メタデータ リソースに加え、独自の暗号化キーとセキュリティ ポリシーを確実に持つことができます。これにより、テナンシーモデルの中で最高レベルの分離とデータ保護が提供されます。さらに、同じ Snowflake 組織内にアカウントを作成することにより、アプリケーションは一元的な請求、アカウント管理、アカウント間のデータ共有などの Snowflake 組織の機能を活用できます。

参考文献:

最新問題: 31

次のコマンドのうち、ウェアハウス クレジットを使用するのはどれですか？

- A. 'SNOWFL%' のようなテーブルを表示します。
- B. SNOWFLAKE から MAX(FLAKE\_ID) を選択します。
- C. スノーフレークから COUNT(\*) を選択します。
- D. FLAKE\_ID による SNOWFLAKE グループから COUNT(FLAKE\_ID) を選択します。

Answer: ([解答を表示する](#))

ウェアハウス クレジットは、Snowflake の各仮想ウェアハウスで使用される処理時間の支払いに使用されます。仮想ウェアハウスは、クエリの実行、データのロード、およびその他の DML 操作の実行を可能にするコンピューティング リソースのクラスターです。ウェアハウス クレジットは、使用する仮想ウェアハウスの数、実行時間、サイズに基づいて課金されます<sup>1</sup>。

質問にリストされているコマンドのうち、次のコマンドはウェアハウス クレジットを使用します。

SELECT MAX(FLAKE\_ID) FROM SNOWFLAKE: このコマンドは、仮想ウェアハウスの実行が必要なクエリであるため、ウェアハウス クレジットを使用します。クエリは SNOWFLAKE テーブルをスキャンし、FLAKE\_ID 列 2 の最大値を返します。したがって、選択肢 B が正しいです。

SELECT COUNT(\*) FROM SNOWFLAKE: このコマンドは、仮想ウェアハウスの実行が必要なクエリであるため、ウェアハウス クレジットも使用します。クエリは SNOWFLAKE テーブルをスキャンし、テーブル内の行数を返します<sup>3</sup>。したがって、選択肢 C が正しいです。

SELECT COUNT(FLAKE\_ID) FROM SNOWFLAKE GROUP BY FLAKE\_ID: このコマンド

これは仮想ウェアハウスの実行が必要なクエリであるため、ウェアハウス クレジットも使用されます。クエリは SNOWFLAKE テーブルをスキャンし、FLAKE\_ID 列の個別の値ごとに行数を返します<sup>4</sup>。したがって、選択肢 D が正しいです。

ウェアハウス クレジットを使用しないコマンドは次のとおりです。

SHOW TABLES LIKE 'SNOWFL%': このコマンドは、仮想ウェアハウスの実行を必要としないメタデータ操作であるため、ウェアハウス クレジットを使用しません。このコマンドは、現在のデータベースと schema5 のパターン 'SNOWFL%' に一致するテーブルの名前を返します。したがって、選択肢 A は不正解です。

参考資料: : コンピューティング コストについて : MAX 関数 : COUNT 関数 : GROUP BY 句 : SHOW TABLES

有効な **ARA-R01** 問題集は GoShiken.com が提供された合格しやすい ARA-R01 試験問題集！ GoShiken.com が最新の **ARA-R01** 試験問題集を提供しています。GoShiken.com ARA-R01 試験問題は最新で、解答が正確でございます。最新の GoShiken.com ARA-R01 問題集をゲットする人はこちら:

<https://www.goshiken.com/Snowflake/ARA-R01-mondaishu.html> (16330%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 32

Snowflake環境を構築する際の有効なオブジェクト階層は何ですか？

- A. アカウント --> データベース --> スキーマ --> ウェアハウス
- B. 組織 --> アカウント --> データベース --> スキーマ --> ステージ
- C. アカウント --> スキーマ > テーブル --> ステージ
- D. 組織 --> アカウント --> ステージ --> テーブル --> ビュー

**Answer:** ([解答を表示する](#))

Snowflake のドキュメントと Web 検索結果によると、これは Snowflake 環境を構築する際の有効なオブジェクト階層です。Snowflake は、データベース、スキーマ、テーブル、ビュー、ステージ、ウェアハウスなど、さまざまなタイプのオブジェクトをサポートするクラウド データ プラットフォームです。これらのオブジェクトは、次のように階層構造に編成されています。

**組織:** 組織は、ビジネス ニーズまたは所有権によって関連付けられた Snowflake アカウントのグループを表す最上位のエンティティです。組織は 1 つ以上のアカウントを持つことができ、アカウント間のデータ共有、請求と使用状況のレポート、アカウント間のシングル サインオンなどの機能を有効にすることができます<sup>12</sup>。

**アカウント:** アカウントは、Snowflake 顧客を表す主要なエンティティです。アカウントには、1 つ以上のデータベース、スキーマ、ステージ、ウェアハウス、その他のオブジェクトを含めることができます。アカウントには 1 つ以上のユーザー、ロール、セキュリティ統合を含めることもできます。アカウントは、特定のクラウド プラットフォーム、リージョン、および Snowflake エディションに関連付けられます<sup>34</sup>。

**データベース:** データベースはスキーマを論理的にグループ化したものです。データベースには 1 つ以上のスキーマを含めることができ、構造化データ、半構造化データ、または非構造化データを保存できます。データベースには、保存期間、暗号化、所有権などのプロパティを持つこともできます<sup>56</sup>。

**スキーマ:** スキーマは、テーブル、ビュー、ステージ、およびその他のオブジェクトの論理的なグループです。スキーマには 1 つ以上のオブジェクトを含めることができ、オブジェクトの名前空間とアクセス制御を定義できます。スキーマには、所有権やデフォルトのウェアハウスなどのプロパティを含めることもできます。

**ステージ:** ステージは、外部ストレージまたは内部ストレージ内のファイルを参照する名前付きの場所です。ステージを使用すると、COPY INTO コマンドを使用して Snowflake テーブルにデータをロードしたり、COPY INTO LOCATION コマンドを使用して Snowflake テーブルからデータをアンロードしたりできます。ステージはアカウント、データベース、またはスキーマ レベルで作成でき、ファイル形式、暗号化、資格情報などのプロパティを持つことができます。

リストされている他のオプションは、構造内の一部のオブジェクトが省略されているか、誤って配置されているため、有効なオブジェクト階層ではありません。たとえば、オプション A では組織レベルが省略され、ウェアハウスがスキーマ レベルの下に配置されますが、これは正しくありません。オプション C では、組織、アカウント、およびステージのレベルが省略され、テーブルがスキーマ レベルの下に配置されますが、これは正しくありません。オプション D はデータベース レベルを省略し、ステージとテーブルをアカウント レベルの下に配置しますが、これは正しくありません。

参考文献:

Snowflake ドキュメント: 組織

Snowflake ブログ: Snowflake の組織の紹介

Snowflake ドキュメント: アカウント

Snowflake ブログ: Snowflake アカウント構造を理解する

Snowflake ドキュメント: データベース

Snowflake ブログ: Snowflake でデータベースを作成する方法

[Snowflake ドキュメント: スキーマ]

[Snowflake ブログ: Snowflake でスキーマを作成する方法]

[Snowflake ドキュメント: ステージ]

[Snowflake ブログ: Snowflake でのステージの使用法]

#### 最新問題: 33

APPLY MASKING POLICY、APPLY ROW ACCESS POLICY、および APPLY SESSION POLICY 権限はどのオブジェクト タイプ レベルで付与できますか?

- A. グローバル
- B. データベース
- C. スキーマ
- D. テーブル

**Answer: A** ([メッセージを残す](#))

APPLY MASKING POLICY、APPLY ROW ACCESS POLICY、および APPLY SESSION POLICY 権限を付与できるオブジェクト タイプ レベルはグローバルです。これらは、列、テーブル、ビュー、アカウント、ユーザーなどのオブジェクトに対してこれらのポリシーを適用または設定解除できるユーザーを制御するアカウント レベルの権限です。これらの権限はデフォルトで ACCOUNTADMIN ロールに付与され、必要に応じて他のロールに付与できます。

他のオプションは、これらの権限を付与できるオブジェクト タイプ レベルではないため、正しくありません。データベース、スキーマ、およびテーブルは、これらの権限をサポートしない下位レベルのオブジェクト タイプです。

参考資料: [アクセス制御権限 | Snowflake ドキュメント](#)、[動的データ マスキングの使用 | Snowflake ドキュメント](#)、[行アクセス ポリシーの使用 | Snowflake ドキュメント](#)、[セッション ポリシーの使用 | スノーフレイクのドキュメント](#)

#### 最新問題: 34

データ共有で使用できる Snowflake オブジェクトはどれですか? (2 つ選択してください)。

- A. 標準ビュー
- B. 安全なビュー
- C. ストアド プロシージャ
- D. 外部テーブル
- E. ストリーム

**Answer: A,B** ([メッセージを残す](#))

データ共有は、アカウント内のデータベース内で選択したオブジェクトを他の Snowflake アカウントと共有できるようにする機能です。次の Snowflake データベース オブジェクトを共有できます: 外部テーブル、動的テーブル、セキュア ビュー、セキュア マテリアライズド ビュー、セキュア UDF、テーブル。ただし、これらのオブジェクトのすべてをデータ共有で使用できるわけではありません。データ共有は、データベースを共有するために必要な情報をカプセル化する名前付きオブジェクトです。データベース ロールを介して、または共有に直接、オブジェクトに

対する権限を共有に付与できません。共有に直接権限を付与できるオブジェクトは、標準ビュー、安全なビュー、安全な UDF、およびテーブルです。

したがって、正解はAとBです。

他のオプションは、共有に直接権限を付与できないため、正しくありません。外部テーブル、動的テーブル、およびストリームは、データベース ロールを介してのみ共有できます。ストアド プロシージャはまったく共有できません。参考文献:

[安全なデータ共有の概要] 1

[共有の操作] 2

[データベースオブジェクトの共有方法の選択] 3

### 最新問題: 35

ある企業には、さまざまな LOT 操作に JSON レコードを提供するソース システムがあります。JSON は、バリエーション フィールドを持つ永続テーブルに直接ロードされます。データは急速に数億レコードに増加しており、パフォーマンスが問題になっています。バリエーション フィールド内の create\_date キーでフィルタリングするために使用される一般的なアクセス パターンがあります。

パフォーマンスを向上させるために何ができるでしょうか？

**A.** JSON レコードから取得した追加フィールドを含めるようにターゲット テーブルを変更します。これには、タイムスタンプのデータ型を持つ create\_date フィールドが含まれます。このフィールドがフィルターで使用されると、パーティションのプルーニングが発生します。

**B.** JSON レコードから取得した追加フィールドを含めるようにターゲット テーブルを変更します。これには、varchar のデータ型を持つ create\_date フィールドが含まれます。このフィールドがフィルターで使用されると、パーティションのプルーニングが発生します。

**C.** 使用されているウェアハウスのサイズを検証します。レコード数が数億に近づく場合、サイズ XL がこの量のデータを処理するために必要な最小サイズになります。

**D.** 日付範囲で分割された複数のテーブルの使用を組み込みます。ユーザーまたはプロセスが特定の日付範囲をクエリする必要がある場合は、適切なベース テーブルが使用されていることを確認してください。

**Answer: A (メッセージを残す)**

正解は A です。これにより、スキャンおよび処理されるデータ量が削減され、クエリのパフォーマンスが向上します。タイムスタンプ データ型の create\_date フィールドを追加することで、Snowflake はこのフィールドに基づいてテーブルを自動的にクラスタ化し、フィルター条件に一致しないマイクロパーティションをプルーニングできます。これにより、JSON データを解析してすべてのレコードのバリエーション フィールドにアクセスする必要がなくなります。

オプション B は、クエリのパフォーマンスが向上しないため、不正解です。varchar データ型で create\_date フィールドを追加すると、Snowflake はこのフィールドに基づいてテーブルを自動的にクラスタリングしたり、フィルター条件に一致しないマイクロパーティションをプルーニングしたりすることができません。この場合も、JSON データを解析し、すべてのレコードのバリエーション フィールドにアクセスする必要があります。

オプション C は、パフォーマンス問題の根本原因に対処していないため、不正解です。使用されているウェアハウスのサイズを検証することで、Snowflake はデータ量に合わせてコンピューティング リソースを調整し、クエリの

実行を並列化できます。ただし、これによってスキャンおよび処理されるデータの量は減りません。これが、JSON データに対するクエリの主なボトルネックです。

オプション D は、データのロードとクエリのプロセスに不必要な複雑さとオーバーヘッドを追加するため、不正解です。Snowflake は、日付範囲でパーティション化された複数のテーブルの使用を組み込むことで、日付範囲を指定するクエリでスキャンおよび処理されるデータの量を削減できます。ただし、これには、複数のテーブルを作成および維持し、日付に基づいて適切なテーブルにデータをロードし、複数の日付範囲にわたるクエリのためにテーブルを結合する必要があります。参考文献:

Snowflake ドキュメント: Snowpipe を使用したデータのロード: このドキュメントでは、Snowpipe を使用して外部ソースから Snowflake テーブルにデータを継続的にロードする方法について説明します。また、ON\_ERROR、PURGE、SKIP\_FILE など、読み込み動作を制御するさまざまなオプションとパラメーターをサポートする COPY INTO コマンドの構文と使用方法についても説明します。

Snowflake ドキュメント: 日付と時刻のデータ型と関数: このドキュメントでは、Snowflake で日付と時刻の値を操作するためのさまざまなデータ型と関数について説明します。また、セッション タイムゾーンとシステム タイムゾーンを設定および変更する方法についても説明します。

Snowflake ドキュメント: メタデータのクエリ: このドキュメントでは、さまざまな関数、ビュー、テーブルを使用して、Snowflake のオブジェクトと操作のメタデータをクエリする方法について説明します。また、COPY\_HISTORY 関数または COPY\_HISTORY ビューを使用してコピー履歴情報にアクセスする方法についても説明します。

Snowflake ドキュメント: JSON データのロード: このドキュメントでは、COPY INTO コマンド、INSERT コマンド、PUT コマンドなどのさまざまな方法を使用して JSON データを Snowflake テーブルにロードする方法について説明します。また、ドット表記、FLATTEN 関数、または LATERAL 結合を使用して JSON データにアクセスし、クエリを実行する方法についても説明します。

Snowflake ドキュメント: パフォーマンスのためのストレージの最適化: このドキュメントでは、クエリのパフォーマンスを向上させるために Snowflake テーブル内のデータのストレージを最適化する方法について説明します。また、自動クラスタリング、検索最適化サービス、マテリアライズド ビューの概念と利点についても説明します。

### 最新問題: 36

ある小売会社では 3000 を超える店舗があり、すべて同じ POS (Point of Sale) システムを使用しています。同社は、ほぼリアルタイムの販売結果をカテゴリ マネージャーに提供したいと考えています。店舗はさまざまなタイムゾーンで営業しており、毎分さまざまな取引が行われており、一部の店舗は他の店舗よりも高い売上高を示しています。

販売結果は、複雑なデータ パイプラインで計算されるデータ エンジニアリング フィールドを使用して、統一された方法で提供されます。計算には、例外、集計、およびスコアリング アルゴリズムに接続された外部関数を使用したスコアリングが含まれます。集計のソース データには 1 億行を超えます。

POS は、ファイルに含まれる一連のトランザクションを識別するための店舗番号とタイムスタンプを含む命名規則に従って、すべての販売トランザクション ファイルを 1 分ごとにクラウドストレージの場所に送信します。通常、ファイルのサイズは 10MB 未満です。

ほぼリアルタイムの結果をカテゴリ マネージャーに提供するにはどうすればよいですか? (2 つ選択してください)。

- A. マイクロインジェストを避けるために、Snowflake に取り込む前にすべてのファイルを連結する必要があります。
- B. Snowpipe を作成し、AUTO\_INGEST = true で構成する必要があります。ストア番号とタイムスタンプを通知するストリーム メタデータを使用して、INSERTS を単一のターゲット テーブルに処理するには、ストリームを作成する必要があります。
- C. ほぼリアルタイムのデータを蓄積するためにストリームを作成し、リアルタイム分析のニーズに一致する頻度で実行するタスクを作成する必要があります。
- D. 外部スケジューラは、クラウド ストレージの場所の内容を調べ、SnowSQL コマンドを発行して、リアルタイム分析のニーズに一致する頻度でデータを処理する必要があります。
- E. ほぼリアルタイムの要件を達成するには、1 秒ごとに実行するようにスケジュールされたタスクを含む copy into コマンドを使用する必要があります。

**Answer:** ([解答を表示する](#))

ほぼリアルタイムの販売結果をカテゴリ マネージャーに提供するには、アーキテクトは次の手順を使用できます。POS が販売トランザクション ファイルを送信するクラウド ストレージの場所を参照する外部ステージを作成します。外部ステージでは、ソース ファイルと一致するファイル形式と暗号化設定を使用する必要があります<sup>2</sup> 外部ステージからファイルを Snowflake のターゲット テーブルにロードする Snowpipe を作成します。Snowpipe は AUTO\_INGEST = true で構成する必要があります。これは、新しいファイルが外部ステージに到着すると自動的に検出して取り込むことを意味します。また、Snowpipe では、重複した取り込みを避けるために、読み込み後にコピー オプションを使用して外部ステージからファイルをパージする必要があります<sup>3</sup>。Snowpipe によって作成された INSERTS をキャプチャするターゲット テーブル上にストリームを作成します。ストリームには、ファイル名、パス、サイズ、最終変更時刻に関する情報を提供するメタデータ列が含まれている必要があります。ストリームには、リアルタイム分析のニーズに一致する保存期間も必要です<sup>4</sup> ストリームに対してクエリを実行して、ほぼリアルタイムのデータを処理するタスクを作成します。クエリでは、ストリーム メタデータを使用してファイル名とパスからストア番号とタイムスタンプを抽出し、外部関数を使用して例外、集計、スコアリングの計算を実行する必要があります。クエリは、カテゴリ マネージャーがアクセスできる別のテーブルまたはビューに結果を出力する必要もあります。タスクは、リアルタイム分析のニーズに一致する頻度 (1 分ごと、5 分ごとなど) で実行されるようにスケジュールする必要があります。

他のオプションは、ほぼリアルタイムの結果を提供するには最適ではなく、実現可能でもありません。

マイクロインジェストを避けるために、Snowflake に取り込む前にすべてのファイルを連結する必要があります。

このオプションは、データ パイプラインに追加の遅延と複雑性が生じるため、推奨されません。

ファイルを連結するには、クラウド ストレージの場所を監視し、ファイルの結合操作を実行する外部プロセスまたはサービスが必要になります。これにより、Snowflake への新しいファイルの取り込みが遅れ、データの損失または破損のリスクが増加します。さらに、Snowpipe は依然として各連結ファイルを個別のロードとして取り込むため、ファイルを連結してもマイクロ取り込みは回避できません。

外部スケジューラは、クラウド ストレージの場所の内容を調べ、SnowSQL コマンドを発行して、リアルタイム分析のニーズに一致する頻度でデータを処理する必要があります。Snowpipe は外部トリガーやスケジューラを必要とせず外部ステージから新しいファイルを自動的に取り込むことができるため、このオプションは必要ありません。外部スケジューラを使用すると、データ パイプラインへのオーバーヘッドと依存関係がさらに追加され、ポーリング間隔と外部スケジューラの可用性に依存するため、ほぼリアルタイムの取り込みは保証されません。

ほぼリアルタイムの要件を達成するには、1 秒ごとに実行するようにスケジュールされたタスクを含む copy into コマンドを使用する必要があります。Snowflake ではタスクを毎秒実行するようにスケジュールできないため、このオプションは実行できません。タスクの最小間隔は 1 分ですが、タスクはスケジュール遅延や同時実行制限の影響を受けるため、これさえも保証されません。さらに、タスクで copy into コマンドを使用すると、ファイルの自動検出、負荷分散、マイクロパーティションの最適化などの Snowpipe の利点を活用できなくなります。参考文献:

- 1: SnowPro Advanced: アーキテクト | 学習ガイド
- 2: スノーフレークのドキュメント | ステージの作成
- 3: スノーフレークのドキュメント | Snowpipe を使用したデータのロード
- 4: スノーフレークのドキュメント | ELT のストリームとタスクの使用
- : スノーフレークのドキュメント | タスクの作成
- : スノーフレークのドキュメント | データロードのベストプラクティス
- : スノーフレークのドキュメント | Snowpipe REST API の使用
- : スノーフレークのドキュメント | タスクのスケジュール設定
- : SnowPro Advanced: アーキテクト | 学習ガイド
- : ステージの作成
- : Snowpipe を使用したデータのロード
- : ELT のストリームとタスクの使用
- : [タスクの作成]
- : [データロードのベストプラクティス]
- : [Snowpipe REST API の使用]
- : [タスクのスケジュール設定]

#### 最新問題: 37

Snowpark\_opt\_wh ウェアハウスで実行すると、Snowpark ストアド プロシージャのメモリと計算リソースを最大化する SQL 変更コマンドはどれですか?

A.

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 2;
```

B.

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 8;
```

C.

D.

```
alter warehouse snowpark_opt_wh set max_concurrency_level = 16;
```

#### Answer: A (メッセージを残す)

Snowpark ストアド プロシージャのメモリとコンピューティング リソースを最大化するには、ストアド プロシージャを実行するウェアハウスの MAX\_CONCURRENCY\_LEVEL パラメータを設定する必要があります。このパラメータは、単一のウェアハウスで実行できる同時クエリの最大数を決定します。これを 16 に設定すると、ウェアハウスは単一ノード上の利用可能なすべての CPU コアとメモリを確実に使用できるようになり、Snowpark に最適化されたウェアハウスにとって最適な構成になります。これにより、他のクエリやノードとリソースを共有する必

要がなくなるため、ストアド プロシージャのパフォーマンスと効率が向上します。他のオプションは、MAX\_CONCURRENCY\_LEVEL パラメーターを変更しないか、16 より低い値に設定しているため、間違っています。これにより、ストアド プロシージャのメモリと計算リソースが削減されます。参考文献:

[スノーパークに最適化された倉庫]1

[Snowpark Python を使用した機械学習モデルのトレーニング] 2

[スノーフレイク ショーツ: スノーパークに最適化された倉庫] 3

### 最新問題: 38

アーキテクトは、大量の構造化データおよび半構造化データを保存および分析するために、Snowflake アカウントとデータベース戦略を設計する必要があります。社内には多くの事業部や部門があります。要件は、スケーラビリティ、セキュリティ、コスト効率です。

どのようなデザインを使用する必要がありますか？

- A. データの量や複雑さに関係なく、すべてのデータ ストレージと分析のニーズに対応する単一の Snowflake アカウントとデータベースを作成します。
- B. データの分離とセキュリティを確保するために、部門または事業単位ごとに個別の Snowflake アカウントとデータベースを設定します。
- C. Snowflake のデータ レイク機能を使用して、構造化スキーマやインデックスを必要とせずに、すべてのデータを中央の場所に保存して分析します。
- D. コア ビジネス データには一元化された Snowflake データベースを使用し、部門またはプロジェクト固有のデータには別のデータベースを使用します。

**Answer: D (メッセージを残す)**

さまざまな事業単位や部門の大量の構造化データおよび半構造化データを保存および分析するための最適な設計は、コア ビジネス データには一元化された Snowflake データベースを使用し、部門またはプロジェクト固有のデータには別のデータベースを使用することです。この設計では、次のような Snowflake の機能を活用することで、スケーラビリティ、セキュリティ、コスト効率を実現します。

データベースのクローン作成: データベースのクローンを作成すると、元のデータベースと同じデータ ファイルを共有するゼロコピー クローンが作成されますが、個別に変更できます。これにより、ストレージ コストが削減され、さまざまな目的での高速かつ一貫したデータ レプリケーションが可能になります。

データベース共有: データベースを共有すると、データベース内のデータのサブセットへの安全で管理されたアクセスを他の Snowflake アカウントまたはコンシューマーに許可できます。これにより、さまざまなビジネスユニットや外部パートナー間でのデータコラボレーションと収益化が可能になります。

ウェアハウスのスケーリング: ウェアハウスをスケーリングすると、さまざまなワークロードのパフォーマンスとコストの要件に合わせてウェアハウスのサイズと同時実行性を調整できます。これにより、さまざまなデータ分析ニーズに合わせた最適リソース利用と柔軟性が可能になります。参考文献: Snowflake ドキュメント: データベースのクローン作成、Snowflake ドキュメント: データベース共有、[Snowflake ドキュメント: ウェアハウスのスケーリング]

### 最新問題: 39

ある企業は、従来のデータベース プラットフォームからの移行の一環として、Snowpipe を使用して 10 TB の CSV データを Snowflake テーブルに取り込もうとしています。レコードは、最もパフォーマンスが高く、コスト効率の高い方法で取り込まれる必要があります。

これらの要件はどのように満たされるのでしょうか？

- A. copy into コマンドで ON\_ERROR = continue を使用します。
- B. copy into コマンドで purge = TRUE を使用します。
- C. copy into コマンドで PURGE = FALSE を使用します。
- D. copy into コマンドのエラー = SKIP\_FILE で使用します。

**Answer: D (メッセージを残す)**

Snowpipe を使用して大量の CSV データを Snowflake に取り込む場合、特に 10 TB などの大量の場合、COPY INTO コマンドの on error = SKIP\_FILE オプションが非常に効果的です。このアプローチにより、Snowpipe は取り込みプロセス中にエラーを引き起こすファイルをスキップできるため、全体のデータ読み込みが停止したり大幅に遅くなったりすることがなくなります。問題のあるファイルの再処理を回避し、他のデータの取り込みを続行することで、パフォーマンスとコスト効率を維持するのに役立ちます。

#### 最新問題: 40

アーキテクトは、2 つの別個の Snowflake アカウントを使用して、Snowflake 実稼働環境と QA 環境を分離することを選択しました。

QA アカウントは、運用アカウントに変更をプッシュする前に、データとデータベース オブジェクトに対する変更を実行してテストすることを目的としています。QA アカウント内のすべてのデータベース オブジェクトとデータは、少なくとも毎晩、Production アカウント内の権限とデータを含め、データベース オブジェクトの正確なコピーである必要があることが要件です。

QA アカウントに運用アカウントのデータとデータベース オブジェクトを毎晩入力するために使用する最も複雑でないアプローチはどれですか？

- A. 1) 各データベースの運用アカウントに共有を作成します  
2) 消費者として QA アカウントへのアクセスを共有する  
3) QA アカウントは各共有から直接データベースを作成します  
4) それらのデータベースのクローンを毎晩作成する  
5) クローン作成されたデータベースで直接テストを実行します。
- B. 1) Production アカウントでステージを作成します  
2) 同じ外部オブジェクト ストレージの場所を指す QA アカウント内にステージを作成します。  
3) 毎晩実行するタスクを作成して、本番アカウントの各テーブルをステージにアンロードします  
4) Snowpipe を使用して QA アカウントに入力します
- C. 1) 運用アカウントの各データベースのレプリケーションを有効にします。  
2) QA アカウントにレプリカ データベースを作成する  
3) レプリカ データベースのクローンを毎晩作成する  
4) クローン作成されたデータベースで直接テストを実行します。
- D. 1) Production アカウントで、QA アカウントに接続し、1 つの特定のテーブルのすべてのデータを返す外部関数を作成します。

2) ストアド プロシージャの一部として外部関数を実行し、運用アカウントの各テーブルをループし、QA アカウントの各テーブルにデータを設定します。

**Answer:** ([解答を表示する](#))

このアプローチは、Snowflake の組み込みレプリケーション機能を使用してデータとデータベース オブジェクトを運用アカウントから QA アカウントにコピーするため、最も複雑ではありません。レプリケーションは、アカウント、リージョン、クラウド プラットフォーム間でデータを同期するための高速かつ効率的な方法です。また、レプリケートされたオブジェクトの権限とメタデータも保持されます。レプリカ データベースのクローンを作成すると、QA アカウントは元のデータに影響を与えることなく、クローン データに対してテストを実行できます。クローンはゼロコピーでもあり、データが変更されない限り、追加のストレージ領域を消費しません。このアプローチでは、データ転送プロセスに複雑さとオーバーヘッドを追加する可能性がある外部ステージ、タスク、Snowpipe、または外部関数は必要ありません。

参考文献:

レプリケーションとフェイルオーバーの概要

複数のアカウント間でのデータベースのレプリケーション

クローン作成に関する考慮事項

最新問題: 41

アーキテクトは、ON\_ERROR=SKIP\_FILE オプションを指定した COPY INTO を使用して、テーブル ステージを使用して CSV ファイルを TABLEA というテーブルに一括ロードします。file5.csv という名前の 1 つのファイルのロードに失敗します。アーキテクトはファイルを修正し、以前とまったく同じファイル名でステージに再ロードします。

アーキテクトはステージから file5.csv ファイルのみをロードするためにどのコマンドを使用する必要がありますか? (2つお選びください。)

A. `@%tablea から tablea にコピー RETURN_FAILED_ONLY = TRUE;`

B. `@%tablea から tablea にコピー`します。

C. `@%tablea FILES = ('file5.csv');` から tablea にコピーします。

D. `@%tablea から tablea にコピー`します。FORCE = TRUE;

E. `@%tablea から tablea にコピー NEW_FILES_ONLY = TRUE;`

F. `COPY INTO tablea FROM @%tablea MERGE = TRUE;`

**Answer:** B,C ([メッセージを残す](#))

オプション A (RETURN\_FAILED\_ONLY) は、以前にロードに失敗したファイルのみをロードします。file5.csv は同じ名前でもステージにすでに存在するため、新しいファイルとは見なされず、ロードされません。

オプション D (FORCE) は、テーブル内の既存のデータを上書きします。file5.csv からデータをロードするだけなので、これは望ましくありません。

オプション E (NEW\_FILES\_ONLY) は、最後の COPY コマンド以降にステージに追加されたファイルのみをロードします。file5.csv は修正される前の段階にすでに存在しているため、これは機能しません。

オプション F (MERGE) は、ステージからのデータを既存のテーブルにマージし、まだ存在しないデータに対して新しい行を作成するために使用されます。この場合、単に file5.csv からデータをロードしたいだけなので、これは必要ありません。

したがって、アーキテクトは COPY INTO tablea FROM @%tablea または COPY INTO tablea FROM を使用できません。

@%tablea FILES = ('file5.csv') ステージから file5.csv のみをロードします。どちらのオプションでも、既存のデータを上書きしたり、追加の構成を必要とせずに、指定されたファイルからデータがロードされます。

#### 最新問題: 42

アーキテクトは、外部ステージから Snowflake への 2 つのファイルの毎日のインポートを自動化する必要があります。1 つのファイルには Parquet 形式のデータが含まれ、もう 1 つのファイルには CSV 形式のデータが含まれます。

最終的な結果セットを生成するには、データをどのように結合および集計する必要がありますか？

- A. Snowpipe を使用して 2 つのファイルを取り込み、マテリアライズド ビューを作成して最終的な結果セットを生成します。
- B. Snowflake スクリプトを使用してファイルをインポートするタスクを作成し、ユーザー定義関数 (UDF) を呼び出して最終的な結果セットを生成します。
- C. 読み取る JavaScript ストアド プロシージャを作成します。外部ステージから直接データを結合して集計し、結果をテーブルに保存します。
- D. マテリアライズド ビューを作成して、外部ステージから直接データを読み取り、結合し、集計し、そのビューを使用して最終結果セットを生成します。

**Answer: B** ([メッセージを残す](#))

Snowflake のドキュメントによると、タスクは、Snowflake で SQL ステートメントまたは JavaScript ユーザー定義関数 (UDF) のスケジュールと実行を可能にするオブジェクトです。タスクを使用すると、データのロード、変換、メンテナンス操作を自動化できます。Snowflake スクリプトは、SQL ステートメントと JavaScript UDF を使用して手続き型ロジックを作成できる機能です。Snowflake スクリプトを使用すると、複雑なワークフローを作成し、タスクを調整できます。したがって、外部ステージから Snowflake への 2 つのファイルの毎日のインポート、データの結合と集計、および最終的な結果セットの生成を自動化する最良のオプションは、COPY INTO コマンドを使用してファイルをインポートする Snowflake スクリプトを使用してタスクを作成することです。次に、UDF を呼び出して結合および集計ロジックを実行します。UDF は、最終的な結果セットとしてテーブルまたはバリエーション値を返すことができます。参考文献:

タスク

スノーフレイク スクリプト

ユーザー定義関数

#### 最新問題: 43

Snowflake のどの組み込み機能がテーブルの変更追跡メタデータを利用しますか? (2つお選びください。)

- A. MERGE コマンド
- B. UPSERT コマンド
- C. CHANGES 句
- D. STREAM オブジェクト
- E. CHANGE\_DATA\_CAPTURE コマンド

**Answer: C,D** ([メッセージを残す](#))

テーブルの変更追跡メタデータを利用する組み込みの Snowflake 機能は、CHANGES 句と STREAM オブジェクトです。CHANGES 句を使用すると、明示的なトランザクション オフセット 1 を使用してストリームを作成しなくても、指定された時間間隔内でテーブルまたはビューの変更追跡メタデータをクエリできます。STREAM オブジェクトは、テーブルに対して行われたデータ操作言語 (DML) の変更 (挿入、更新、削除など) と各変更に関するメタデータを記録し、変更されたデータを使用してアクションを実行できるようにします。このプロセスは、変更データ キャプチャ (CDC)<sup>2</sup> と呼ばれます。他のオプションは、テーブルの変更追跡メタデータを利用しないため、正しくありません。MERGE コマンドは、ソース テーブルとの結合の結果に基づいて、ターゲット テーブルに対して挿入、更新、または削除操作を実行します<sup>3</sup>。UPSERT コマンドは有効な Snowflake コマンドではありません。CHANGE\_DATA\_CAPTURE コマンドは有効な Snowflake コマンドではありません。参考資料: 変更点 | Snowflake ドキュメント、テーブル ストリームを使用した変更追跡 | Snowflake ドキュメント、MERGE | スノーフレイクのドキュメント

#### 最新問題: 44

次の DDL コマンドは、ストリームに基づいてタスクを作成するために使用されました。

```
CREATE TASK ts_insert_new_customers
  WAREHOUSE = MY_WH
  Schedule = '5 minute'
WHEN
  System$STREAM_HAS_DATA('MYSTREAM')
AS
  INSERT INTO new_customers(id, name) SELECT id, name
FROM mystream WHERE METADATA$ACTION = 'INSERT';
```

MY\_WH が auto\_suspend - 60 に設定され、このタスク専用で使用されると仮定すると、どのステートメントが真ですか？

- A. ストリームをチェックするために、ウェアハウス MY\_WH が 5 分ごとにアクティブになります。
- B. ウェアハウス MY\_WH は、ストリームに結果がある場合にのみアクティブになります。
- C. ウェアハウス MY\_WH は一時停止されません。
- D. ウェアハウス MY\_WH は、ストリームのサイズに合わせて自動的にサイズ変更されます。

**Answer: B** ([メッセージを残す](#))

ウェアハウス MY\_WH は、ストリームに結果がある場合にのみアクティブになります。これは、タスクがストリームに基づいて作成されるためです。つまり、タスクはストリームに新しいデータがある場合にのみ実行されます。さらに、ウェアハウスは auto\_suspend - 60 に設定されます。これは、60 秒間の非アクティブ状態が続くとウェアハウスが自動的に一時停止することを意味します。したがって、ウェアハウスはストリームに結果がある場合にのみアクティブになります。参考文献:

[タスクの作成 | スノーフレイクのドキュメント]

[ストリームとタスクの使用 | スノーフレイクのドキュメント]

[倉庫を作成 | スノーフレイクのドキュメント]

#### 最新問題: 45

Snowpipe のイベント通知の特徴は何ですか？

- A. ロード履歴はターゲットテーブルのメタデータに保存されます。
- B. 通知は、クラウドストレージ イベントとファイル内の実際のデータを識別します。
- C. Snowflake は、一時停止されたパイプが再開されると、すべての古い通知を処理できます。
- D. パイプが一時停止されると、パイプに対して受信されたイベント メッセージは制限された保存期間に入ります。

**Answer:** ([解答を表示する](#))

Snowpipe のイベント通知は、ステージ内の新しいファイルまたは変更されたファイルを Snowflake に通知するためにクラウドストレージ プロバイダーによって送信されるメッセージです。Snowpipe はこれらの通知を使用して、ステージからターゲット テーブルへのデータの読み込みをトリガーします。パイプが一時停止されると、パイプに対して受信されたイベント メッセージは制限された保存期間に入ります。この期間はクラウドストレージ プロバイダーによって異なります。保持期間内にパイプが再開されない場合、イベント メッセージは破棄され、データは自動的にロードされません。データをロードするには、パイプを再開し、COPY コマンドを手動で実行する必要があります。これは、Snowpipe のイベント通知の特徴であり、他のオプションとは異なります。参考資料:

Snowflake ドキュメント: Snowpipe の使用、Snowflake ドキュメント: パイプの一時停止と再開

#### 最新問題: 46

Snowflake オブジェクト階層に基づいて、どのセキュリティ保護可能なオブジェクトが Snowflake アカウントに直接属しますか？

(3 つを選択)。

- A. データベース
- B. スキーマ
- C. テーブル
- D. ステージ
- E. 役割
- F. 倉庫

**Answer:** A,E,F ([メッセージを残す](#))

セキュリティ保護可能なオブジェクトは、Snowflake でアクセスを許可できるエンティティです。セキュリティ保護可能なオブジェクトには、データベース、スキーマ、テーブル、ビュー、ステージ、パイプ、関数、プロシージャ、シーケンス、タスク、ストリーム、ロール、ウェアハウス、共有などがあります<sup>1</sup>。

Snowflake オブジェクト階層は、セキュリティ保護可能なオブジェクトをネストされた方法で編成する論理構造です。最上位のコンテナはアカウントであり、顧客組織のすべてのデータベース、ロール、ウェアハウスが含まれています。各データベースにはスキーマが含まれており、スキーマにはテーブル、ビュー、ステージ、パイプ、関数、プロシージャ、シーケンス、タスク、ストリームが含まれています。各ロールには、他のロールまたはセキュリティ保護可能なオブジェクトに対する権限を付与できます。各ウェアハウスは、セキュリティ保護可能なオブジェクトに対するクエリの実行に使用できます<sup>2</sup>。

Snowflake オブジェクト階層に基づいて、Snowflake アカウントに直接属するセキュリティ保護可能なオブジェクトは、データベース、ロール、ウェアハウスです。これらのオブジェクトはアカウント レベルで作成および管理さ

れ、他のセキュリティ保護可能なオブジェクトには依存しません。他のオプションは次の理由により正しくありません。

スキーマはアカウントではなくデータベースに属します。スキーマは既存のデータベース内に作成する必要があります3。

テーブルはアカウントではなくスキーマに属します。テーブルは既存のスキーマ内に作成する必要があります4。

ステージはアカウントではなく、スキーマまたはテーブルに属します。ステージは既存のスキーマまたはテーブル内に作成する必要があります。

参考文献:

1: アクセス制御の概要 | スノーフレークのドキュメント

2: セキュリティ保護可能なオブジェクト | スノーフレークのドキュメント

3: スキーマの作成 | スノーフレークのドキュメント

4: テーブルの作成 | スノーフレークのドキュメント

[5]: ステージの作成 | スノーフレークのドキュメント

有効な **ARA-R01** 問題集は GoShiken.com が提供された合格しやすい ARA-R01 試験問題集！ GoShiken.com が最新の **ARA-R01** 試験問題集を提供しています。GoShiken.com ARA-R01 試験問題は最新で、解答が正確でございます。最新の GoShiken.com ARA-R01 問題集をゲットする人はこちら:

<https://www.goshiken.com/Snowflake/ARA-R01-mondaishu.html> (16330%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

最新問題: 47

ヘルスケア企業は医療機関とデータを共有したいと考えています。この研究所は Snowflake の Standard エディションを実行しています。ヘルスケア会社は Business Critical エディションを実行しています。

このデータはどのように共有できますか？

A. 医療会社は、アカウント パネルで研究所の Snowflake エディションを変更する必要があります。

B. デフォルトでは、共有は Business Critical Snowflake エディションから Standard エディションまでサポートされています。

C. Snowflake に連絡すると、ヘルスケア会社の共有リクエストを実行します。

D. 共有オブジェクトの share\_restriction パラメータを false に設定します。

**Answer: D (メッセージを残す)**

デフォルトでは、Snowflake は Business Critical エディションから非 Business Critical エディションへのデータの共有を許可しません。これは、Business Critical エディションでは、下位エディションでは利用できない強化されたセキュリティおよびデータ保護機能が提供されるためです。ただし、この制限は、共有オブジェクト (データベース、スキーマ、またはテーブル) の share\_restriction パラメータを false に設定することでオーバーライドできます。このパラメータを使用すると、データ プロバイダーは下位エディションのアカウントとのデータ共有を明示的に許可できます。このパラメータはデータ プロバイダーのみが設定でき、データ コンシューマーは設定できないことに注意してください。また、このパラメータを false に設定すると、共有データのセキュリティとデータ保護のレベルが低下する可能性があります。

#### 参考文献:

データ共有を有効にする: ビジネス クリティカル アカウントから下位エディションへのデータ共有を有効にする  
BUSINESS CRITICAL Edition のアカウントから下位エディションのアカウントへの共有は許可されない SQL 実行エラー: BUSINESS CRITICAL Edition のアカウントから下位エディションのアカウントへの共有は許可されません。スノーフレイクのドキュメント

#### 最新問題: 48

アーキテクトは、接続と切断のタイムスタンプ、ユーザーのユーザー名、および要約統計情報を含む VPN\_ACCESS\_LOGS テーブルを SECURITY\_LOGS スキーマに持っています。

このテーブルで Snowflake 検索最適化サービスを有効にするには、アーキテクトは何をすべきですか？

- A. 今後のテーブルで OWNERSHIP を持つロールを引き受け、SECURITY\_LOGS スキーマで ADD SEARCH OPTIMIZATION を引き受けます。
- B. SECURITY LOGS スキーマの ADD SEARCH OPTIMIZATION を含むすべての権限を持つ役割を引き受けます。
- C. VPN\_ACCESS\_LOGS の OWNERSHIP と SECURITY\_LOGS スキーマの ADD SEARCH OPTIMIZATION を持つロールを引き受けます。
- D. VPN\_ACCESS\_LOGS に対する ALL PRIVILEGES と SECURITY\_LOGS スキーマの ADD SEARCH OPTIMIZATION を持つロールを引き受けます。

**Answer: C (メッセージを残す)**

SnowPro Advanced: Architect Exam Study Guide によると、テーブルで検索最適化サービスを有効にするには、ユーザーはテーブルとスキーマに対する ADD SEARCH OPTIMIZATION 権限を持っている必要があります。権限は明示的に付与することも、データベースやロールなどの上位レベルのオブジェクトから継承することもできます。テーブルに対する OWNERSHIP 権限は ADD SEARCH OPTIMIZATION 権限を意味するため、テーブルを所有するユーザーはテーブルに対する検索最適化サービスを有効にすることができます。したがって、正しい答えは、VPN\_ACCESS\_LOGS の OWNERSHIP と SECURITY\_LOGS スキーマの ADD SEARCH OPTIMIZATION を持つロールを引き受けることです。これにより、ユーザーは VPN\_ACCESS\_LOGS テーブルおよび SECURITY\_LOGS スキーマで作成される今後のテーブルで検索最適化サービスを有効にすることができます。他のオプションは、過剰な権限を付与するか、テーブルまたはスキーマに必要な権限を付与しないため、正しくありません。参考文献:

SnowPro Advanced: Architect 試験学習ガイド、11 ページ、セクション 2.3.1

Snowflake ドキュメント: 検索最適化サービスの有効化

#### 最新問題: 49

次のテーブルが実稼働データベースに存在します。

規制要件では、企業はデータがクエリされるときに、現在の日付に基づいて 6 か月以上前のイベントのユーザー名をマスクする必要があると規定されています。

イベント データを複製せず、テーブルを使用してビューを作成するときやテーブルのクローンを作成するときイベント データが確実に適用されるようにすることなく、どのように要件を満たすことができるでしょうか？

- A. 有効な日付を持つ資格テーブルを使用して、ユーザー名列にマスキング ポリシーを使用します。
- B. 有効な日付を持つ資格テーブルを使用して、user\_events テーブルで行レベルのポリシーを使用します。

- C. 条件列としてevent\_timestampを持つユーザー名列にマスキング ポリシーを使用します。
- D. username 列の case ステートメントを使用して、user\_events テーブルで安全なビューを使用します。

**Answer:** ([解答を表示する](#))

マスキング ポリシーは、ユーザーの役割とデータの状態に基づいてクエリ結果内の機密データをマスキングできる Snowflake の機能です。マスキング ポリシーはテーブルまたはビュー内の列に適用でき、同じテーブルまたはビュー内の別の列を条件付き列として使用できます。条件列は、値1に基づいて秘匿化ポリシーを適用するかどうかを決定する列です。

この場合、event\_timestamp を条件列としてユーザー名列のマスキング ポリシーを使用することで要件を満たすことができます。マスキング ポリシーでは、event\_timestamp が現在の日付に基づいて 6 か月以上古い場合にユーザー名をマスクし、それ以外の場合は元のユーザー名を返す関数を使用できます。マスキング ポリシーは user\_events テーブルに適用でき、テーブルを使用してビューを作成するとき、または table2 を複製するときにも適用されます。

他のオプションは次の理由により正しくありません。

A). 有効な日付を含む資格テーブルを使用してユーザー名列にマスキング ポリシーを使用するには、各ユーザー名の有効な日付を格納する別のテーブルを作成し、それをマスキング ポリシー関数で user\_events テーブルと結合する必要があります。これにより、マスキング ポリシーに複雑さとオーバーヘッドが追加され、マスキングの条件としてevent\_timestamp列が使用されなくなります。

B). 有効な日付を持つ資格テーブルを使用して user\_events テーブルで行レベルのポリシーを使用するには、各ユーザー名の有効な日付を保存する別のテーブルを作成し、それを行アクセス ポリシー関数で user\_events テーブルと結合する必要があります。これにより、username 列をマスクするのではなく、有効な日付に基づいて、event\_timestamp が 6 か月より古い行がフィルターで除外されます。これはユーザー名をマスクするという要件を満たさず、イベント データの可視性も低下します。

D). username 列の case ステートメントを使用して user\_events テーブルで安全なビューを使用するには、case 式を使用して、event\_timestamp 列に基づいて username 列をマスクするビューを作成する必要があります。これはユーザー名をマスクするという要件を満たしますが、テーブルのクローン作成時には適用されません。安全なビューは、ビューに対するクエリによって基になるデータが公開されるのを防ぐビューです。ただし、安全なビューであっても、テーブルのクローンを作成することによって基になるデータが公開されるのを防ぐことはできません<sup>3</sup>。

参考文献:

- 1: マスキングポリシー | スノーフレイクのドキュメント
- 2: マスキング ポリシーでの条件付き列の使用 | スノーフレイクのドキュメント
- 3: 安全なビュー | スノーフレイクのドキュメント

**Valid ARA-R01 Dumps** shared by GoShiken.com for Helping Passing ARA-R01 Exam! GoShiken.com now offer the **newest ARA-R01 exam dumps**, the GoShiken.com ARA-R01 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com ARA-R01 dumps with Test Engine here:

<https://www.goshiken.com/Snowflake/ARA-R01-mondaishu.html> (163 Q&As Dumps, 30%OFF Special

Discount: **Freepdfdumps**)