Snowflake.ARA-C01.v2026-07-27.q156

試験コード：	ARA-C01
試験名称：	SnowPro Advanced Architect Certification
認証ベンダー：	Snowflake
無料問題の数：	156
バージョン：	v2026-07-27
ページの閲覧量：	107
問題集の閲覧量：	1762

https://www.jpnshiken.com/shiken/Snowflake.ARA-C01.v2026-07-27.q156.html

質問: **1**
Snowflakeのタスクでは、夏時間による現地時間の変更はどのように処理されますか？ 2つ選択してください。）
A. UTCベースのスケジュールでスケジュールされたタスクは、時間の変更による問題は発生しません。
B. タスクスケジュールは、時間変更に対応するために、指定されたタイムゾーンまたは現地のタイムゾーンに従うように設計できます。
C. 夏時間への変更中は、タスクは一時停止状態になります。
D. 数分ごとの頻繁なタスク実行スケジュールは問題を引き起こさないかもしれませんが、タスク履歴に影響を与えます。
E. タスク スケジュールは指定された時間のみに従い、時間の損失や重複には対応できません。
正解: **A,B (**コメントを発表する**)**
Snowflakeのドキュメント1とWeb検索結果2によると、Snowflakeタスクにおける夏時間による現地時間の変更の処理方法について、以下の2つの記述は正しい。タスクとは、SnowflakeでSQLステートメントまたはストアドプロシージャをスケジュールして実行できるようにする機能である。タスクは、タスクの実行頻度とタイムゾーンを指定するcron式を使用してスケジュールできる。
* UTCベースのスケジュールでスケジュールされたタスクは、時刻変更による影響を受けません。UTCは、夏時間を適用しない世界標準の時刻です。したがって、タイムゾーンとしてUTCを使用するタスクは、現地の時刻変更に関係なく、年間を通して同じ時刻に実行されます1。
タスクのスケジュールは、指定されたタイムゾーンまたはローカルタイムゾーンに従って設計することで、時刻の変更に対応できます。Snowflake は、タスクの cron 式で有効な IANA タイムゾーン識別子を使用することをサポートしています。これにより、タスクは指定されたタイムゾーンのローカルタイムに従って実行され、夏時間調整も含まれる場合があります。たとえば、タイムゾーンとして Europe/London を使用するタスクは、ローカルタイムが GMT と BST12 の間で切り替わると、1 時間早くまたは遅く実行されます。
Snowflakeドキュメント :タスクのスケジュール設定
Snowflakeコミュニティ :Snowflakeでタスクをスケジュールする際に使用されるタイムゾーンは、夏時間に対応していますか？

質問: **2**
クローンされたテーブルがセカンダリデータベースに複製されると、データもセカンダリデータベースに複製されます。
A. 偽
B. 真
正解: **(**正解を表示します**)**

質問: **3**
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すればよいでしょうか？

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を示します。

D. FILERONSはファイル形式の場所を示します。FILE_ROW_NUMBERはステージです。

正解: **(**正解を表示します**)**

ステージとは、Snowflake内でデータロードおよびアンロード用のファイルを保存できる名前付きの場所です。ステージは、ファイルの保存場所に応じて、内部ステージまたは外部ステージに分類されます。

質問にあるクエリは、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、FILE_ROW_NUMBER（ステージ内のファイルの行番号）を含む、さまざまな列を持つテーブルを返します。

したがって、このクエリは、FILEROWSという名前のステージ内のファイルを一覧表示し、そのステージ内の各ファイルの行番号を表示するものと解釈できます。

参照：

段階

：リスト関数

質問: **4**

スキーマ所有者は、通常のスキーマでオブジェクト権限を付与できます。

A. 真

B. 偽

正解: **B (**コメントを発表する**)**

質問: **5**

ある医療関連企業が、個人健康情報（PHI）を含む可能性のあるSnowflakeアカウントを導入しようとしている。

会社は、関連するすべてのプライバシー基準を遵守しなければならない。

データ保護およびコンプライアンス要件を満たすベストプラクティス推奨事項はどれですか？（3つ選択してください。）

A. 最低でもSnowflakeのBusiness Criticalエディションを使用してください。

B. 動的データマスキングポリシーを作成し、PHI を含む列に適用します。

C. 内部トークン化機能を使用して機密データを難読化します。

D. 外部トークン化機能を使用して機密データを難読化します。

E. current_role() に基づく PHI データの投影を排除するように SQL クエリを書き換えます。

F. 提携組織とのデータ共有は避けてください。

正解: **(**正解を表示します**)**
PHIデータを扱う医療関連企業は、HIPAA、HITRUST、GDPRなどの関連するプライバシー基準を遵守する必要があります。Snowflakeは、顧客がデータ保護とコンプライアンス要件を満たすのに役立ついくつかの機能とベストプラクティスを提供しています1。
推奨されるベストプラクティスの一つは、最低でもSnowflakeのBusiness Criticalエディションを使用することです。このエディションは、顧客管理キーによるエンドツーエンド暗号化、強化されたオブジェクトレベルのセキュリティ、HIPAAおよびHITRUST準拠など、最高レベルのデータ保護とセキュリティを提供します。したがって、選択肢Aが正解です。
もう1つのベストプラクティスの推奨事項は、動的データマスキングポリシーを作成し、PHIを含む列に適用することです。動的データマスキングは、現在のユーザーの役割に基づいて機密データをマスキングまたは編集できる機能です。これにより、承認されたユーザーのみがマスキングされていないデータを表示でき、他のユーザーはNULL、アスタリスク、ランダムな文字などのマスキングされた値を見ることができます3。したがって、オプションBが正解です。
3つ目のベストプラクティス推奨事項は、外部トークン化機能を使用して機密データを難読化することです。
外部トークン化とは、機密データを、Protegrityなどの外部サービスによって生成および保存されたトークンに置き換える機能です。この方法により、元のデータはSnowflakeによって保存または処理されることはなく、承認されたユーザーのみが外部サービス4を介してトークン化されたデータにアクセスできます。したがって、選択肢Dが正解です。
オプションCは誤りです。Snowflakeには内部トークン化機能がありません。Snowflakeはネイティブのトークン化機能を提供しておらず、外部トークン化サービスとの統合のみをサポートしています4。
オプションEは誤りです。current_role()に基づいてPHIデータの投影を排除するためにSQLクエリを書き換えることは、ベストプラクティスではありません。この方法はエラーが発生しやすく、非効率的で、保守も困難です。より良い代替策は、動的データマスキングポリシーを使用することです。これにより、クエリを変更することなく、ユーザーの役割に基づいてデータを自動的にマスキングできます3。
選択肢Fは誤りです。なぜなら、提携組織とのデータ共有を避けることは、ベストプラクティスではないからです。
Snowflakeは、事業部門、顧客、パートナーなどの社内外の利用者との間で、安全かつ統制されたデータ共有を可能にします。データ共有は、データのコピーや移動ではなく、共有オブジェクトへのアクセス権限の付与のみを伴います。また、データ共有では、動的データマスキングと外部トークン化機能を利用して、機密データを保護することもできます5。
Snowflakeのセキュリティとコンプライアンスに関するレポート :Snowflakeのエディション：動的データマスキング：外部トークン化：安全なデータ共有

質問: **6**
既存のクラスタキーを持つテーブルに対して、代替クラスタキーを定義する機能を提供するのはどの機能ですか？
A. 外部テーブル
B. 実物大ビュー
C. 検索最適化
D. 結果キャッシュ
正解: **B (**コメントを発表する**)**
マテリアライズド ビューは、既存のクラスタ キーを持つテーブルに対して代替クラスタ キーを定義できる機能です。マテリアライズド ビューは、Snowflake に格納される事前計算済みの結果セットであり、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 ビューは、ベース テーブルとは異なるクラスタ キーを持つことができ、これによりマテリアライズド ビューに対するクエリのパフォーマンスと効率が向上します。また、マテリアライズド ビューは、ベース テーブル データに対する集計、結合、フィルタをサポートできます。AUTO_REFRESH パラメータが true に設定されている場合、ベース テーブルの基となるデータが変更されると、マテリアライズド ビューは自動的に更新されます。

マテリアライズドビュー | Snowflake ドキュメント

質問: **7**
アーキテクトは、特定のテーブル上のさまざまなアクセスパスのパフォーマンスを向上させるために、最適なクラスタリングをどのように実現できるでしょうか？
A. テーブルに対して複数のクラスタリングキーを作成します。
B. 異なるクラスタキーを持つ複数のマテリアライズドビューを作成します。
C. クラスタリングを自動的に作成するスーパープロジェクションを作成します。
D. アクセスパスで使用されるすべての列を含むクラスタリングキーを作成します。
正解: **(**正解を表示します**)**
説明
SnowPro Advanced: Architect のドキュメントと学習リソースによると、特定のテーブル上のさまざまなアクセス パスのパフォーマンスを向上させるために最適なクラスタリングを有効にする最良の方法は、異なるクラスタ キーを持つ複数のマテリアライズド ビューを作成することです。マテリアライズド ビューは、1 つ以上のベース テーブルに対するクエリから派生する、事前に計算された結果セットです。マテリアライズド ビューは、クラスタリング キーを指定することでクラスタリングできます。クラスタリング キーとは、マテリアライズド ビュー内のデータがマイクロ パーティション内でどのように共存するかを決定する列または式のサブセットです。異なるクラスタ キーを持つ複数のマテリアライズド ビューを作成することで、アーキテクトは同じベース テーブル上の異なるアクセス パスを使用するクエリのパフォーマンスを最適化できます。たとえば、ベース テーブルに列 A、B、C、および D があり、A と B、C と D、または A と C でフィルタリングするクエリがある場合、アーキテクトは、それぞれ異なるクラスタ キーを持つ 3 つのマテリアライズド ビュー (A、B)、(C、D)、および (A、C) を作成できます。こうすることで、各クエリは対応するマテリアライズドビューの最適なクラスタリングを活用し、より高速なスキャン効率とより優れた圧縮率を実現できます。
参考文献：
* Snowflakeドキュメント :マテリアライズドビュー
* スノーフレーク学習 :マテリアライズドビュー
https://www.snowflake.com/blog/using-materialized-views-to-solve-multi-clustering-performance-problems/

質問: **8**
Snowpark_opt_whウェアハウス上で実行された場合、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SQL ALTERコマンドはどれですか？
A.
```
alter warehouse snowpark_opt_wh set max_concurrency_level = 1;
```
B.
```
alter warehouse snowpark_opt_wh set max_concurrency_level = 2;
```
C.
```
alter warehouse snowpark_opt_wh set max_concurrency_level = 8;
```
D.
```
alter warehouse snowpark_opt_wh set max_concurrency_level = 16;
```
正解: **(**正解を表示します**)**

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 パラメーターを設定する必要があります。このパラメーターは、単一のウェアハウスで実行できる同時実行クエリの最大数を決定します。これを 16 に設定すると、ウェアハウスが単一ノード上の利用可能なすべての CPU コアとメモリを使用できるようになります。これは、Snowpark に最適化されたウェアハウスにとって最適な構成です。これにより、ストアド プロシージャは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、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 より低い値に設定するため、ストアド プロシージャのメモリとコンピューティング リソースが減少するため、正しくありません。参照:
* [スノーパーク最適化倉庫] 1
* [Snowpark Python を使用した機械学習モデルのトレーニング] 2
* [スノーフレークショーツ :スノーパーク最適化倉庫] 3

質問**: 9**
あるメディア企業は、顧客レビューデータをSnowflakeテーブルに取り込み、いくつかの変換処理を適用するデータパイプラインを必要としています。また、同社はAmazon Comprehendを使用して感情分析を行い、匿名化された最終データセットを、異なる地域で異なるクラウドプロバイダーを利用する広告会社向けに公開する必要があります。
データパイプラインは、イベント通知を活用しながら、オブジェクトストレージに新しいレコードが到着するたびに、継続的かつ効率的に動作する必要があります。また、運用上の複雑さ、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保守、および開発作業は最小限に抑える必要があります。
どのデザインがこれらの要件を満たすでしょうか？
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オーケストレーションします。データをAmazon S3にエクスポートしてAmazon Comprehendでモデル推論を実行し、データをSnowflakeテーブルに再度取り込みます。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オーケストレーションします。Amazon Comprehendを使用してモデル推論を実行する外部関数を作成し、最終的なレコードをSnowflakeテーブルに書き込みます。次に、Snowflake Marketplaceにリストを作成し、他の企業がデータを利用できるようにします。
C. Amazon EMRとPySparkを使用してSnowflake Sparkコネクタ経由でデータをSnowflakeに取り込みます。別のSparkジョブを使用して変換処理を適用します。Amazon Comprehendテキスト分析APIを活用してモデル推論を行うPythonプログラムを開発します。その後、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オーケストレーションします。データをAmazon S3にエクスポートしてAmazon Comprehendでモデル推論を実行し、データをSnowflakeテーブルに取り戻します。その後、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。
正解**: (**正解を表示します**)**
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 にデータを継続的にロードできる機能です。効率的でスケーラブルなサーバーレスであるため、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不要です。ストリームとタスクは、変更データ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。これらも効率的でスケーラブルなサーバーレスであり、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、データに対して感情分析を実行できます。結果は、標準 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間でデータコンシューマーとデータを共有できるプラットフォームです。他の企業にデータを公開するための安全で簡単な方法です。
参照：
Snowpipeの概要 | Snowflakeドキュメント
データパイプライン入門 | Snowflakeドキュメント
外部機能の概要 | Snowflakeドキュメント
Snowflakeデータマーケットプレイスの概要 | Snowflakeドキュメント

質問: **10**
ある企業は、異なる地域に複数の拠点を持ち、そこからデータを取り込みたいと考えている。
以下のどれが、このタイプのデータ取り込みを可能にしますか？
A. 会社は各サイトにリーダーアカウントをプロビジョニングし、リーダーアカウントを通じてデータを取り込む必要があります。
B. 企業は、各クラウドリージョンにSnowflakeアカウントを所有している必要があり、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する必要があります。
C. 会社はSnowflakeアカウント間でデータを複製する必要があります。
D. 外部ステージにはストレージ統合を使用すべきです。
正解: **(**正解を表示します**)**

質問: **11**
建築家は、以下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はテーブルを見つけることができません。
最小権限の原則を用いてUSER1がテーブルを見つけるために、アーキテクトは以下のどのコマンドを実行する必要がありますか？ 2つ選択してください。）
A. 役割を 「公開」から 「インターン」に付与する。
B. ロール INTERN にデータベース サンドボックスの使用権限を付与します。
C. ロール INTERN にスキーマ SANDBOX.PUBLIC の使用権限を付与します。
D. ユーザー INTERN にデータベース サンドボックスの所有権を付与します。
E. データベースサンドボックスに対するすべての権限をロール INTERN に付与します。
正解: **(**正解を表示します**)**
最小権限の原則によれば、アーキテクトはUSER1がSANDBOXデータベース内のテーブルを見つけるために必要な最小限の権限のみを付与すべきである。
USER1は、PUBLICスキーマ内のテーブルにアクセスできるように、SANDBOXデータベースおよびSANDBOX.PUBLICスキーマに対するUSAGE権限を持っている必要があります。したがって、コマンドBとCを実行するのが正しいです。

コマンドAは正しくありません。なぜなら、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を持たないからです。
コマンドDは正しくありません。なぜなら、このコマンドはSANDBOXデータベースの所有権をアーキテクトからUSER1に移転してしまうからです。これは不要であり、最小権限の原則に違反します。
コマンドEは正しくありません。なぜなら、このコマンドはSANDBOXデータベースに対するすべての可能な権限をUSER1に付与してしまうからです。これは不必要であり、最小権限の原則に違反します。
Snowflake - 最小権限の原則 : Snowflake - アクセス制御権限 : Snowflake - パブリックロール :
Snowflake - 所有権と助成金

質問**: 12**
データベースDB1には、テーブルT1が1つ存在するスキーマS1があります。
DB1 --> S1 --> T1
EG1の保存期間は10日間に設定されています。
s: の保持期間は20日間に設定されています。
tの保持期間は30日間に設定されます。
ユーザーは以下のコマンドを実行します。
データベースDB1を削除する。
T1のタイムトラベル保持期間はどれくらいになりますか？
A. 10日間
B. 20日間
C. 30日間
D. 37日
正解**: C (**コメントを発表する**)**
T1 の Time Travel 保持期間は 30 日です。これはテーブル レベルで設定された保持期間です。Time Travel 保持期間は、オブジェクトが変更または削除された後、履歴データが保持され、アクセス可能な期間を決定します。Time Travel 保持期間は、アカウント レベル、データベース レベル、スキーマ レベル、またはテーブル レベルで設定できます。階層の最下位レベルで設定された保持期間が上位レベルよりも優先されます。したがって、テーブル レベルで設定された保持期間は、スキーマ レベル、データベース レベル、またはアカウント レベルで設定された保持期間よりも優先されます。ユーザーがデータベース DB1 を削除すると、テーブル T1 も削除されますが、履歴データはテーブル レベルで設定された保持期間である 30 日間保持されます。ユーザーは UNDROP コマンドを使用して、30 日以内にテーブル T1 を復元できます。その他のオプションは、次の理由により誤りです。
10日間はデータベースレベルで設定された保持期間であり、テーブルレベルで設定することで上書きされます。
20日間はスキーマレベルで設定された保持期間であり、テーブルレベルによっても上書きされます。
37日間は有効な選択肢ではありません。なぜなら、どのレベルにおいても、その期間は保持期間として設定されていないからです。
参照：
タイムトラベルの理解と活用
AT | BEFORE
スノーフレーク -タイムトラベル＆フェイルセーフ

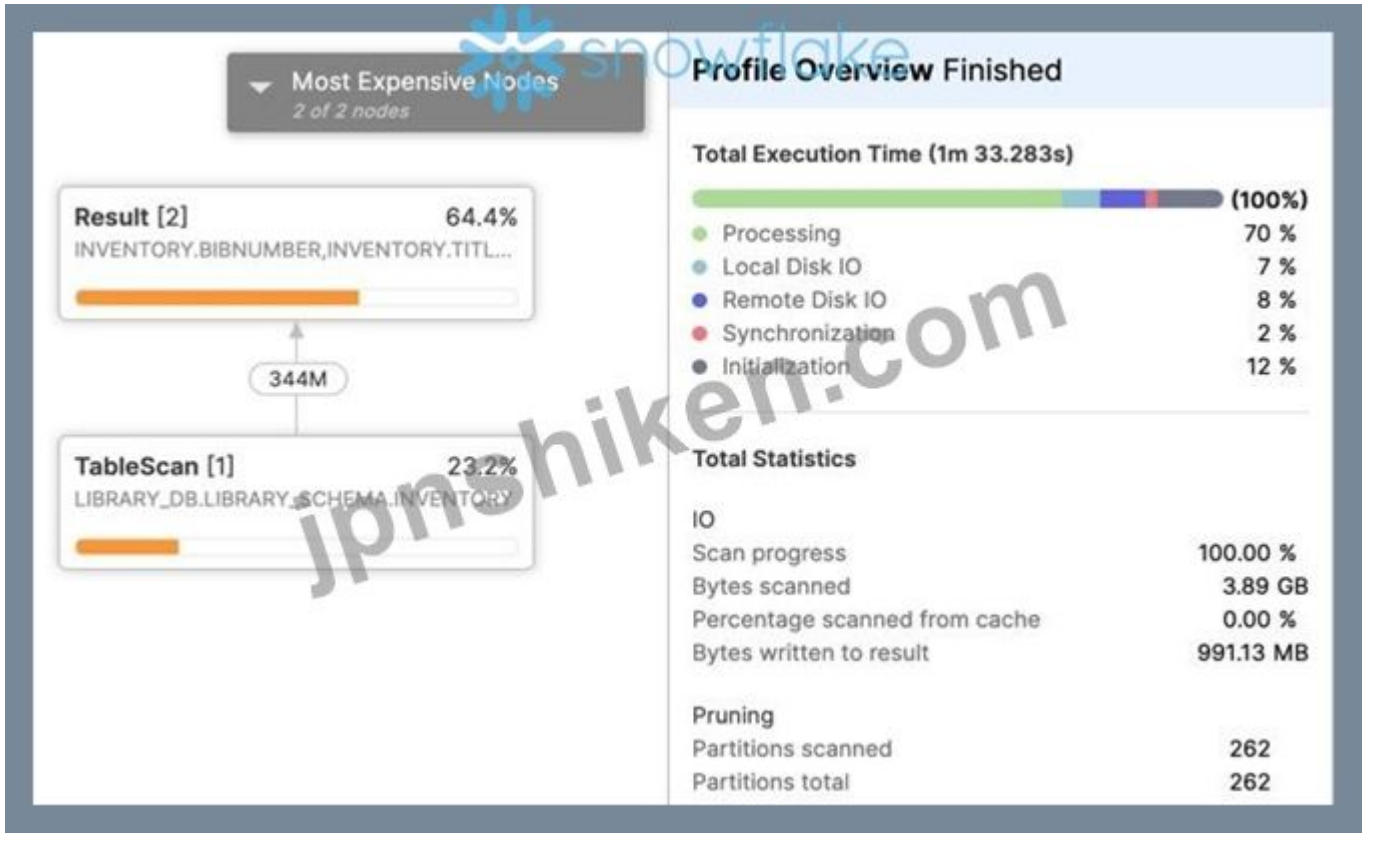
質問**: 13**

データはJSON形式でインポートされ、VARIANT型の列に保存されています。クエリのパフォーマンスは以前は良好でしたが、最近になってパフォーマンスの低下が報告されています。
何が原因なのでしょうか？
A. 最近のデータインポートにJSONのnull値が含まれていました。
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では、JSON値の文字列の長さにばらつきがありました。
正解: **(**正解を表示します**)**
データはJSON形式でVARIANT型の列にインポートされ、保存されています。クエリのパフォーマンスは以前は良好でしたが、最近になってパフォーマンスの低下が報告されています。これは以下の要因が原因であ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、最も一般的な要素については列のような構造で、残りの要素については 残り物」のような列で、半構造化データを内部的に保存します。JSON内のキーの順序は、Snowflakeが共通要素をどのように判断し、クエリのパフォーマンスを最適化するかに影響します。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て内部ストレージを再編成する必要があり、その結果、クエリのパフォーマンスが低下する可能性があります。
最近のデータインポートでは、JSON値の文字列長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型の列にロードされると文字列として格納されます。これらの値に対する操作は、対応するデータ型を持つ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はより多くの領域を割り当て、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があります。
その他の選択肢は、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のnull値が含まれていました。Snowflakeは、半構造化データにおいて、SQLのNULLとJSONのnullという2種類のnull値をサポートしています。SQLのNULLは値が欠落しているか不明であることを意味し、JSONの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、適切に処理することができます。最近のデータインポートにJSONのnull値が含まれていても、クエリのパフォーマンスに大きな影響はありません。
最近のデータインポートでは、通常よりもフィールド数が少なくなっています。Snowflakeは、スキーマやフィールドが異なる半構造化データを処理できます。最近のデータインポートでフィールド数が通常より少ない場合でも、Snowflakeは既存のフィールドに基づいてデータ取り込みとクエリ実行を最適化できるため、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。
参照：
VARIANTに格納される半構造化データに関する考慮事項
Snowflake Architect トレーニング
バリアント列内の一意要素に対するSnowflakeクエリのパフォーマンス
Snowflakeバリアントのパフォーマンス

質問**: 14**
あなたは以下のクエリを実行しました。私は自動停止が5秒に設定されている倉庫を持っています。
SELECT * FROM INVENTORY;
クエリプロファイルは以下のようになります。 ⊞キャッシュからスキャンされた割合」は0%です。

5秒経過する前にクエリを再度実行したところ、クエリプロファイルは以下のようになりました。「キャッシュのスキャン率」を見ると、75%となっています。

5秒後に再度クエリを実行しました。クエリプロファイルは以下のようになります。 「キャッシュからスキャンされた割合」を見ると、再びゼロになっています。



なぜこのようなことが起きているのか？
A. クエリの2回目の実行では、クエリ結果キャッシュが使用されました
B. クエリの2回目の実行では、データキャッシュを使用して結果の一部を取得しました。これは、データウェアハウスが停止される前に実行されたためです。
C. クエリの3回目の実行では、クエリ結果キャッシュが使用されました
正解: **B (**コメントを発表する**)**

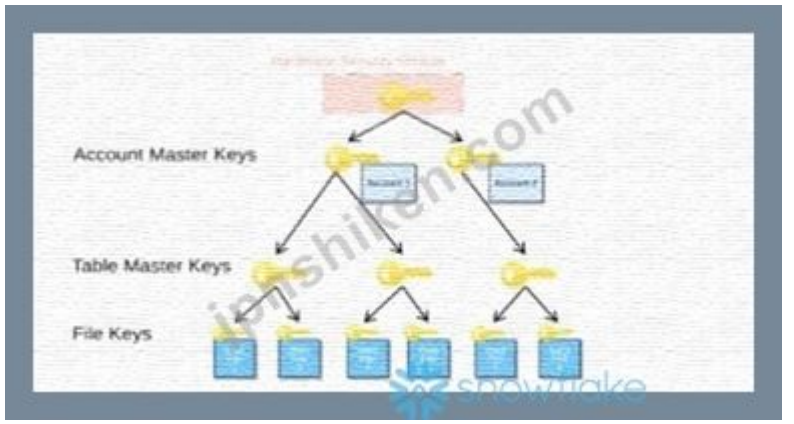
質問: **15**
Assuming all Snowflake accounts are using an Enterprise edition or higher, in which development and testing scenarios would be copying of data be required, and zero-copy cloning not be suitable? (Select TWO).
A. Developers create their own datasets to work against transformed versions of the live data.
B. Production and development run in different databases in the same account, and Developers need to see production-like data but with specific columns masked.
C. Data is in a production Snowflake account that needs to be provided to Developers in a separate development/testing Snowflake account in the same cloud region.
D. Developers create their own copies of a standard test database previously created for them in the development account, for their initial development and unit testing.
E. The release process requires pre-production testing of changes with data of production scale and complexity. For security reasons, pre-production also runs in the production account.
正解: **(**正解を表示します**)**
ゼロコピークローニングは、データを物理的にコピーすることなく、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。ゼロコピークローニングは、クローンされたオブジェクトが元のオブジェクトと同じデータとメタデータを持つ必要があり、クローン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ないシナリオに適しています。ゼロコピークローニングは、クローン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。ただし、ゼロコピークローニングは、クローンされたオブジェクトが元のオブジェクトとは異なるデータまたはメタデータを持つ必要がある場合、またはクローン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シナリオには適していません。
ゼロコピークローニングは、クローン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。このようなシナリオでは、COPY INTO コマンドを使用するか、セキュアビュー3 を使用したデータ共有を使用して、データのコピーが必要になります。以下は、データのコピーが必要で、ゼロコピークローニングが適さない開発およびテストシナリオの例です。
開発者は、ライブデータの変換バージョンに対して作業するために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、クローンされた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。ゼロコピーのクローンは、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のクローンを作成するため、適していません。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します。4
* データは本番環境の Snowflake アカウントにあり、開発者には別の方法で提供する必要があります。
* 同じクラウドリージョン内の開発/テスト用 Snowflake アカウント。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 クローニングは、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し、クローンを別のアカウントと共有できないため、適していません。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、セキュア ビューを使用したデータ共有または COPY INTO コマンドを使用できます。5 以下は、ゼロコピー クローニングが適しており、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、開発者は特定の列をマスクした本番環境と同様のデータを参照する必要があります。このシナリオでは、データは同じアカウント内で共有する必要があり、クローンされたオブジェクトは元のオブジェクトと異なるデータやメタデータを持つ必要がないため、ゼロコピークローニングを使用できます。ゼロコピークローニングを使用すると、開発データベースに本番データベースのクローンを作成でき、クローンは元のデータベースと同じデータとメタデータを持つことができます。特定の列をマスクするには、クローンの上にセキュアビューを作成し、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。
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コピーを、初期開発および単体テスト用に作成します。このシナリオでは、データは同じアカウント内で共有する必要があり、クローンされたオブジェクトは元のオブジェクトとは異なるデータやメタデータを持つ必要がないため、ゼロコピークローニングを使用できます。ゼロコピークローニングを使用すると、各開発者用に標準テストデータベースのクローンを作成でき、クローンは元のデータベースと同じデータとメタデータを持つことができます。開発者はクローンを初期開発および単体テストに使用でき、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。
* リリースプロセスでは、本番規模と複雑さのデータを使用して変更の事前テストを行う必要があります。セキュリティ上の理由から、事前テストは本番アカウントでも実行されます。このシナリオでは、データは同じアカウント内で共有する必要があり、クローンされたオブジェクトは元のオブジェクトとは異なるデータやメタデータを持つ必要がないため、ゼロコピークローニングを使用できます。ゼロコピークローニングを使用すると、本番データベースのクローンを事前プロダクションデータベースに作成でき、クローンは元のデータベースと同じデータとメタデータを持つことができます。事前プロダクションテストでは、クローンを使用して本番規模と複雑さのデータを使用して変更をテストでき、クローンに加えられた変更は元のデータベースや本番環境に影響を与えません。8 参考文献:
* 1: SnowPro Advanced: Architect | 学習ガイド 9
* 2: Snowflake ドキュメント | クローン作成の概要
* 3: Snowflake ドキュメント | COPY を使用してテーブルにデータをロードする
* 4: Snowflake ドキュメント | ロード中のデータ変換
* 5 :Snowflakeドキュメント | データ共有の概要
* 6: Snowflake ドキュメント | セキュア ビュー
* 7: Snowflake 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: Snowflake ドキュメント | テストと開発のためのクローン作成
* : SnowPro Advanced: Architect | 学習ガイド
* :クローニングの概要
* : COPYを使用してテーブルにデータをロードする
* : ロード中のデータ変換
* :データ共有の概要
* : セキュアビュー
* : データベース、スキーマ、およびテーブルのクローン作成
* : テストおよび開発のためのクローニング

質問**: 16**
ロールに付与されているすべての権限とロールを一覧表示するには、どのコマンドを実行しますか？
A. ロール<ロール名>の権限を表示
B. ロール<ロール名>の権限付与を表示する
C. ロール <ロール名> の権限を表示する
D. ロール <ロール名> への権限付与を表示

正解: (正解を表示します)

質問: 17
Snowflakeアカウントの階層型暗号化モデルでTri-Secret Secureを有効化する場合、顧客管理キーはどのレベルで使用されますか？



A. ルートレベル（HSM）
B. アカウントレベル（AMK）
C. テーブルレベル（TMK）
D. マイクロパーティションレベルで
正解: (正解を表示します)
Tri-Secret Secure は、Snowflake 管理キーに加えて、顧客が独自のキー (顧客管理キー (CMK) と呼ばれる) を使用して、Snowflake 内のデータを暗号化する複合マスターキーを作成できる機能です。複合マスターキーは、アカウントマスターキー (AMK) とも呼ばれ、アカウントごとに固有であり、データファイルを暗号化するファイルキーを暗号化するテーブルマスターキー (TMK) を暗号化します。顧客管理キーは、ルートレベル、テーブルレベル、またはマイクロパーティションレベルではなく、アカウントレベルで使用されます。ルートレベルはハードウェアセキュリティモジュール (HSM) によって保護され、テーブルレベルは TMK によって保護され、マイクロパーティションレベルはファイルキーによって保護されます 12。参考文献:
* Snowflakeにおける暗号化キー管理の理解
* AWS 上の Snowflake の Tri-Secret Secure FAQ

質問: 18
共有から作成されたSnowflakeデータベースは、共有から作成されていない標準データベースとどのように異なりますか？ 3つ選択してください。）
A. 共有データベースは読み取り専用です。
B. 新しいデータを表示するには、共有データベースを更新する必要があります。
C. 共有データベースはクローンできません。
D. タイムトラベルでは共有データベースはサポートされていません。

E. 共有データベースには、これらのスキーマを明示的に共有に付与しなくても、PUBLIC または INFORMATION_SCHEMA スキーマが存在します。
F. 共有データベースは、一時的なデータベースとしても作成できます。
正解: **A,C,D (**コメントを発表する**)**
SnowPro Advanced: Architectのドキュメントと学習リソースによると、共有から作成されたSnowflakeデータベースと、共有から作成されていない標準データベースとの違いは次のとおりです。
共有データベースは読み取り専用です。つまり、共有データベースにアクセスするデータ利用者は、データベース内のデータやオブジェクトを変更または削除することはできません。データベースを共有するデータ提供者は、データとオブジェクトを完全に制御でき、それらに対する権限を付与または取り消すことができます[1]。
* 共有データベースはクローンできません。つまり、共有データベースにアクセスするデータ利用者は、データベースまたはデータベース内のオブジェクトのコピーを作成できません。データベースを共有するデータ提供者は、データベースまたはオブジェクトをクローンできますが、クローンは自動的に共有されません。
* 共有データベースはタイムトラベルではサポートされていません。つまり、共有データベースにアクセスするデータコンシューマーは、AS OF句を使用して履歴データを照会したり、削除されたデータを復元したりすることはできません。データベースを共有するデータプロバイダーは、データベースまたはオブジェクトに対してタイムトラベルを使用できますが、履歴データはサポートされていません。
* データはデータ利用者には見えません3。
他のオプションは、共有から作成された Snowflake データベースと、共有から作成されていない標準データベースとの違いを説明するものではないため、誤りです。オプション B は、共有データベースでは新しいデータを表示するために更新する必要がないため誤りです。共有データベースにアクセスするデータ コンシューマーは、データ プロバイダがデータを更新するとすぐに最新のデータを見ることができます 1。オプション E は、共有データベースに PUBLIC スキーマまたは INFORMATION_SCHEMA スキーマを明示的に付与しない限り、共有データベースにはこれらのスキーマが存在しないため誤りです。共有データベースにアクセスするデータ コンシューマーは、データ プロバイダが共有に付与したオブジェクトのみを表示でき、PUBLIC スキーマと INFORMATION_SCHEMA スキーマはデフォルトでは付与されません 4。オプション F は、共有データベースを一時データベースとして作成できないため誤りです。一時データベースは、タイム トラベルまたはフェイル セーフをサポートしないデータベースであり、データの保持期間に影響を与えることなく削除できます。共有データベースは、ソース データベースの種類に関係なく、常に永続データベースとして作成されます 5。参照: セキュア データ共有の概要 | 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、タイムトラベル | Snowflakeドキュメント、共有の操作 | Snowflakeドキュメント、データベースの作成 | Snowflakeドキュメント

質問: **19**
Snowflakeのデータモデリング手法のうち、BIクエリ向けに設計されているのはどれですか？
A. 星型図
B. 3 NF
C. スノーフレークスキーマ
D. データ保管庫
正解: **(**正解を表示します**)**

質問: **20**
テーブルで自動クラスタリングを有効にしました。
特定のテーブルについて、過去1週間に自動クラスタリングで消費されたクレジット量をどのように監視しますか？

A. 以下のクエリを実行します。select * from table(information_schema.automatic_clustering_history(date_range_start=>dateadd(d, -7, current_date), date_range_end=>current_date, table_name=>'mydb.myschema.mytable'));
B. クラスタリング情報を表示
C. 自動クラスタリング情報の表示
正解: **(**正解を表示します**)**

質問: **21**
ある建築家がSnowflakeを使ってデータレイクを設計しています。同社は構造化データ、半構造化データ、非構造化データを保有しています。同社はSnowflakeシステム内のデータレイクにデータを保存したいと考えています。また、Snowflakeのデータ共有機能を利用して、各支社間でデータを共有することを計画しています。
Snowflake内で非構造化データを共有する際に考慮すべき点は何ですか？
A. 安全なビューを介してデータを共有するために、署名付きURLを使用して非構造化データをSnowflakeに保存する必要があります。URLには時間制限はありません。
B. 安全なビューを介してデータを共有するために、スコープ付き URL を使用して非構造化データを Snowflake に保存する必要があります。URL の有効期限は 24 時間です。
C. セキュアビューを介してデータを共有するために、非構造化データを Snowflake に保存するにはファイル URL を使用する必要があります。URL の有効期限は 7 日間です。
D. セキュアなビューを介してデータを共有するため、非構造化データを Snowflake に保存するにはファイル URL を使用する必要があります。URL の有効期限には「expiration_time」引数が定義されます。
正解: **D** (コメントを発表する)
Snowflakeのドキュメントによると、非構造化データファイルはセキュアビューとセキュアデータ共有を使用して共有できます。セキュアビューを使用すると、クエリの結果をテーブルのようにアクセスできます。セキュアビューは、特にデータプライバシーのために設計されています。スコープ付きURLは、ステージングされたファイルに一時的にアクセスできるようにするエンコードされたURLで、ステージングされたファイルに権限を付与しません。URLは、永続化されたクエリ結果の期間が終了すると期限切れになります。現在の有効期限は24時間です。スコープ付きURLは、ファイル管理者が同じアカウント内の特定のロールにデータファイルへのスコープ付きアクセスを付与する場合に推奨されます。Snowflakeは、スコープ付きURLを使用してファイルにアクセスしたユーザーとその日時に関する情報をクエリ履歴に記録します。したがって、スコープ付きURLは、セキュリティ、説明責任、およびデータアクセスの制御を提供するため、Snowflake内で非構造化データを共有するのに最適なオプションです。参照:
非構造化データを安全なビューで共有する
非構造化データのロード入門

質問: **22**
Snowflakeでクラスタキーの優先順位付けを行う際に推奨されるベストプラクティスはどれですか？ 2つ選択してください。）
A. 結合述語で頻繁に使用される列を選択します。
B. クラスタリングキーをサポートし、コスト効率を高めるために、カーディナリティの低い列を選択します。
C. 一意の行数を最大にするには、ナノ秒を含むTIMESTAMP列を選択してください。
D. 選択フィルタで最も頻繁に使用されるクラスタ列を選択します。
E. GROUP BY句で実際に使用されているクラスタ列を選択します。
正解: **(**正解を表示します**)**

Snowflakeのドキュメントによると、クラスタリングキーを選択する際のベストプラクティスは以下のとおりです。
結合述語で頻繁に使用される列を選択してください。これにより、スキャンおよび結合する必要のあるマイクロパーティションの数を減らし、結合パフォーマンスを向上させることができます。
* 選択フィルタで最も頻繁に使用される列を選択してください。これにより、フィルタ条件に一致しないマイクロパーティションをスキップできるため、スキャン効率が向上します。
性別や国など、カーディナリティの低い列をクラスタリングキーとして使用することは避けてください。クラスタリング精度が低下し、メンテナンスコストが高くなる可能性があります。
ナノ秒単位のTIMESTAMP列は、カーディナリティが非常に高く、他の列との相関が低い傾向があるため、使用を避けてください。これにより、クラスタリングの精度が低下したり、メンテナンスコストが高くなったりする可能性があります。
* 重複値やNULL値を含む列の使用は避けてください。これらの列はクラスタリングに偏りを生じさせ、枝刈りの効果を低下させる可能性があります。
クエリで複数のフィルタまたは結合述語を使用する場合は、複数の列でクラスタリングを実行します。これにより、より多くのマイクロパーティションが削除され、圧縮率が向上する可能性が高まります。
クラスタリングは、特に小規模または中規模のテーブル、あるいはクエリや更新の頻度が低いテーブルでは、必ずしも有効とは限りません。クラスタリングには、データの初期クラスタリングと、その後のクラスタリング維持のための追加コストが発生する可能性があります。
クラスタリングキーとクラスタ化テーブル | Snowflakeドキュメント
[テーブルのクラスタリングを選択する際の考慮事項 | Snowflakeドキュメント]

質問: **23**
データ共有は、データプロバイダーアカウントとデータコンシューマーアカウント間で行われます。プロバイダーアカウントの5つのテーブルがコンシューマーアカウントと共有されています。コンシューマーロールには、インポート権限が付与されています。
プロバイダースキーマに新しいテーブル（table_6）が追加された場合、消費者アカウントはどうなりますか？
A. 消費者ロールでは新しいテーブルが自動的に表示され、追加の権限付与は必要ありません。
B. 消費者側でこの助成金が付与された後にのみ、消費者の役割でこの表が表示されます。
データベース PSHARE_EDW_4TEST_DB に対するインポートされた権限を DEV_ROLE に付与します。
C. 消費者側は、プロバイダー側でこの許可が与えられた後にのみ、このテーブルを見ることができます。
ロール「accountadmin」を使用する。
PSHARE_EDW_4TEST にテーブル EDW.ACCOUNTING.Table_6 の選択権限を付与します。
D. 消費者側は、プロバイダー側でこの許可が与えられた後にのみ、このテーブルを見ることができます。
ロール「accountadmin」を使用する。
データベースEDW上での使用権限を共有PSHARE_EDW_4TESTに付与します。
スキーマ EDW.ACCOUNTING での使用権限を付与し、共有 PSHARE_EDW_4TEST を許可します。
データベース PSHARE_EDW_4TEST_DB に対して、テーブル EDW.ACCOUNTING.Table_6 に対する選択権限を付与します。
正解: **(**正解を表示します**)**
データ共有の一部であるプロバイダのアカウントのスキーマに新しいテーブル (table_6) が追加されても、コンシューマーは新しいテーブルを自動的に表示しません。コンシューマーは、プロバイダから適切な権限が付与された後にのみ、新しいテーブルにアクセスできるようになります。オプション D で説明されている正しい手順では、プロバイダの ACCOUNTADMIN ロールを使用してデータベースとスキーマに対する USAGE 権限を付与し、続いて新しいテーブルに対する SELECT 権限を付与します。具体的には、コンシューマーのデータベースを含む共有に

対して権限を付与します。これにより、確立されたデータ共有設定の下でコンシューマー アカウントが新しいテーブルにアクセスできるようになります。参照:
* Snowflakeのアクセス制御管理に関するドキュメント
* Snowflakeのデータ共有に関するドキュメント

質問: **24**
社内アプリケーションの中には、ブラウザアクセスやリダイレクト機能を使用せずにSnowflakeに接続する必要があるものが複数あります。Snowflakeにおける認証のベストプラクティスは何ですか？
A. Snowflake OAuthを使用します。
B. ユーザー名とパスワードを使用してください。
C. 外部OAuthを使用します。
D. サービスユーザーとのキーペア認証を使用します。
正解: **D (**コメントを発表する**)**
対話型ではないサービス間認証シナリオでは、Snowflakeはサービスユーザーを使用したキーペア認証を推奨しています（回答D）。この方法はパスワードのハードコーディングを回避し、認証情報の自動ローテーションをサポートし、セキュリティのベストプラクティスに準拠しています。
OAuthベースの認証方式は通常、ブラウザのリダイレクトやユーザー操作を必要としますが、このシナリオではそれらは利用できません。ユーザー名とパスワードによる認証は、セキュリティリスクと運用上の負担を伴います。
キーペア認証は、強力な証明書ベースのセキュリティを実現し、SnowPro Architectのアプリケーション、ETLツール、自動化ワークロードの設計において広く使用されています。

質問: **25**
5分間の自動停止が設定された中規模のデータウェアハウスがあります。テーブル#1に対してクエリを実行しました。10分後、テーブル#1とテーブル#2を結合するクエリを実行しましたが、クエリがデータを使用していないことがわかりました。
キャッシュ。
なぜ？
A. ジョインではいかなる種類のキャッシュも使用できません
B. 倉庫が停止すると、データキャッシュが失われる可能性があります。
C. 2回目に実行されたクエリは、テーブル#1で実行されたクエリと完全に異なるため、データキャッシュを使用できません。
正解: **(**正解を表示します**)**

質問: **26**
What are characteristics of Dynamic Data Masking? (Select TWO).
A. A masking policy that Is currently set on a table can be dropped.
B. A single masking policy can be applied to columns in different tables.
C. A masking policy can be applied to the value column of an external table.
D. The role that creates the masking policy will always see unmasked data In query results
E. A masking policy can be applied to a column with the GEOGRAPHY data type.
正解: **(**正解を表示します**)**

Dynamic Data Masking is a feature that allows masking sensitive data in query results based on the role of the user who executes the query. A masking policy is a user-defined function that specifies the masking logic and can be applied to one or more columns in one or more tables. A masking policy that is currently set on a table can be dropped using the ALTER TABLE command. A single masking policy can be applied to columns in different tables using the ALTER TABLE command with the SET MASKING POLICY clause. The other options are either incorrect or not supported by Snowflake. A masking policy cannot be applied to the value column of an external table, as external tables do not support column-level security. The role that creates the masking policy will not always see unmasked data in query results, as the masking policy can be applied to the owner role as well. A masking policy cannot be applied to a column with the GEOGRAPHY data type, as Snowflake only supports masking policies for scalar data types. References: Snowflake Documentation:
Dynamic Data Masking, Snowflake Documentation: ALTER TABLE

質問: **27**
景色が見えます。
ビュー内のすべてのオブジェクト参照を一覧表示するにはどうすればよいですか？
A. GET_VIEW_METADATA
B. GET_OBJECT_REFERENCES
C. GET_VIEW_REFERENCES
正解: **(**正解を表示します**)**

質問: **28**
あるアーキテクトは、Snowpipe の Snowflake Connector for Kafka を使用してデータをストリーミングする計画を立てています。
コストを最適化するには、どのような設定が適切でしょうか？
A. buffer.flush.time = 1 に設定します。
B. buffer.count.records = 1 に設定します。
C. buffer.size.bytes = 10 MB に設定します。
D. マイクロパーティションの数を最大化する。
正解: **(**正解を表示します**)**
Snowflake Connector for KafkaをSnowpipeと連携して使用する場合、コスト効率は、データを取り込む前に適切なサイズのファイルにバッチ処理するかどうかに左右されます。Snowflakeは、取り込まれたファイルの数とロードに使用されたコンピューティングリソースに基づいてSnowpipeの料金を請求します。ファイルサイズが非常に小さいと、オーバーヘッドとコストが大幅に増加します。
buffer.size.bytes を 10 MB に設定すると、コネクタはレコードを適切なサイズのファイルにバッチ処理してから Snowflake にフラッシュすることができます (回答 C)。これにより、取り込みの遅延とコスト効率のバランスが取れ、ストリーミング取り込みに関する Snowflake のベストプラクティスに準拠します。バッファサイズが極端に小さい場合や、フラッシュ間隔が非常に短い場合 (1 レコードまたは 1 秒など)、ファイルの作成が過剰になり、Snowpipe のコストが高くなります。
マイクロパーティションの最大化は直接設定できない上、コストとパフォーマンスの面で逆効果となる。
SnowPro Architectの受験者にとって、この質問は、KafkaとSnowpipeを使用したストリーミング取り込みパイプラインを設計する際に、バッチ処理とファイルサイズ戦略がいかに重要であるかを強調するものです。

質問: **29**
ある建築家は、過去6年間の四半期末の財務結果を保存したいと考えている。
アーキテクトは、これを実現するためにどのSnowflake機能を利用できますか？

A. 検索最適化サービス
B. 実物大ビュー
C. タイムトラベル
D. ゼロコピークローニング
E. 安全なビュー
正解: **(**正解を表示します**)**
説明
ゼロコピークローニングは、Snowflakeの機能の一つで、過去6年間の四半期末の財務結果を保存するために使用できます。ゼロコピークローニングを使用すると、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、データやメタデータを複製することなく作成できます。クローンは元のオブジェクトと同じデータファイルを共有しますが、クローンまたは元のオブジェクトに加えられた変更は個別に追跡されます。ゼロコピークローニングを使用すると、四半期末の財務結果など、異なる時点のデータのスナップショットを作成し、将来の分析や比較のために保存できます。ゼロコピークローニングは高速で効率的であり、データが変更されない限り、追加のストレージ容量を消費しません[1]。
参考文献：
* ゼロコピークローニング | Snowflake ドキュメント

質問: **30**
企業の毎日の Snowflake ワークロードは、次のタイミングでトリガーされる膨大な数の同時クエリで構成されています。
午後9時と午後11時。個々のレベルでは、これらのクエリは短時間で完了する小規模なステートメントです。
会社のアーキテクトは、このワークロードのパフォーマンスを向上させるために、どのような構成を実装できるでしょうか？
（2つ選択してください。）
A. ワークロード実行中は、マルチクラスタ仮想倉庫を最大モードで有効にします。
B. 仮想倉庫レベルで、MAX_CONCURRENCY_LEVEL をデフォルト値の 8 よりも高い値に設定します。
C. 仮想倉庫のサイズを特大サイズに拡大します。
D. Reduce the amount of data that is being processed through this workload.
E. Set the connection timeout to a higher value than its default.
正解: **A,B (**コメントを発表する**)**
Explanation
These two configuration options can enhance the performance of the workload that consists of a huge number of concurrent queries that are smaller and faster.
* Enabling a multi-clustered virtual warehouse in maximized mode allows the warehouse to scale out automatically by adding more clusters as soon as the current cluster is fully loaded, regardless of the number of queries in the queue. This can improve the concurrency and throughput of the workload by minimizing or preventing queuing. The maximized mode is suitable for workloads that require high performance and low latency, and are less sensitive to credit consumption1.
* Setting the MAX_CONCURRENCY_LEVEL to a higher value than its default value of 8 at the virtual warehouse level allows the warehouse to run more queries concurrently on each cluster. This can improve the utilization and efficiency of the warehouse resources, especially for smaller and faster queries that do not require a lot of processing power. The MAX_CONCURRENCY_LEVEL parameter can be set when creating or modifying a warehouse, and it can be changed at any time2.
References:
* Snowflake Documentation: Scaling Policy for Multi-cluster Warehouses
* Snowflake Documentation: MAX_CONCURRENCY_LEVEL

質問: **31**
企業は、Snowflakeアカウントで以下の機能を利用できる必要があります。
1. 多要素認証（MFA）のサポート
2. 最低2ヶ月間のタイムトラベルの空き状況
3. 異なるリージョン間でのデータベースのレプリケーション
4. JDBCおよびODBCのネイティブサポート
5. Tri-Secret Secureを使用した顧客管理型暗号化キー
6. 決済カード業界データセキュリティ基準（PCI DSS）への対応
上記すべてのサービスを提供するために、アカウント作成時に選択すべき最低限のSnowflakeエディションは何ですか？
A. 企業
B. ビジネス上重要な
C. 仮想プライベートサーバー（VPS）
D. 標準
正解: **(**正解を表示します**)**

質問: **32**
アーキテクトは、データアナリストによる迅速なレポート作成をサポートするために、Snowflakeアーキテクチャを設計しています。コストを最適化するため、仮想ウェアハウスは2分間アイドル状態が続くと自動的に停止するように構成されています。クエリは更新後の朝に一度実行されますが、それ以降のクエリは実行速度が遅くなります。
なぜこのようなことが起こるのでしょうか？
A. 倉庫が十分な大きさではありません。
B. 倉庫はマルチクラスタ倉庫として構成されていませんでした。
C. 倉庫は USE_CACHE = TRUE で作成されていません。
D. 倉庫が停止されたとき、キャッシュがドロップされました。
正解: **(**正解を表示します**)**
Snowflakeの仮想ウェアハウスは、ウェアハウスが稼働している間のみ、ローカルの結果キャッシュとデータキャッシュを保持します。
ウェアハウスが手動または自動で一時停止されると、ローカルキャッシュがクリアされます。その結果、後続のクエリはキャッシュされたデータを利用できなくなり、リモートストレージからデータを再スキャンする必要があるため、実行速度が低下します（正解はD）。
Snowflakeはクラウドサービス層でグローバルな結果キャッシュを保持していますが、これは全く同じクエリテキストが再実行され、基となるデータが変更されていない場合にのみ使用されます。多くの分析ワークロードでは、クエリがわずかに異なるため、結果キャッシュの再利用が妨げられます。
倉庫の規模とマルチクラスタ構成は、キャッシュの永続性ではなく、同時実行性とスループットに影響を与えます。

SnowflakeにはUSE_CACHEパラメータはありません。この問題は、Snowflakeのキャッシング動作と、コスト管理のための積極的な自動サスペンドとパフォーマンスのためのキャッシュ再利用とのトレードオフに関するアーキテクトの理解度を問うものです。

質問: **33**
入力文字列をJSONドキュメントとして解釈し、VARIANT値を生成する半構造化データ関数はどれですか。
A. PARSE_XML
B. STRIP_JSON
C. PARSE_JSON
正解: **C (**コメントを発表する**)**

質問: **34**
Snowflakeのコンテキスト関数は、列レベルのセキュリティが適用されているデータに対してユーザーが閲覧権限を持っているかどうかを判断するのにどのように役立ちますか？ 2つ選択してください)。
A. 現在のセッションで使用されているロールを対象に、current_roleを使用してマスキングポリシー条件を設定します。
B. is_role_in_session を使用して、現在のアカウントで使用されているロールを対象としたマスキングポリシー条件を設定します。
C. マスキングポリシーに、任意の機能の使用を許可する所有権権限があるかどうかを判断します。
D. オブジェクトを実行するユーザーにアカウント管理者ロールを割り当てます。
E. SQL文で実行ロールを対象とするinvoker_roleを使用してマスキングポリシー条件を設定します。
正解: **A,E (**コメントを発表する**)**

質問: **35**
ある企業は、Snowflakeアカウントをインターネットから見えないように、社内ネットワーク内に展開したいと考えています。同社はSnowflakeユーザー向けにVPNインフラストラクチャと仮想デスクトップインフラストラクチャ (VDI) を使用しています。また、ログイン管理時の冗長性を排除するため、VDI用に設定したログイン認証情報を再利用したいと考えています。
これらの要件を満たすには、Snowflakeのどの機能を使用すべきでしょうか？ 2つ選択してください。)
A. ユーザーが社内VPNの外部から接続できるように、レプリケーションを設定します。
B. 独自の企業トライシークレットセキュアキーを提供します。
C. クラウドプロバイダーのプライベート接続を使用します。
D. フェデレーション認証のためのSSOを設定します。
E. VPN の外部でプロキシ Snowflake アカウントを使用し、ユーザー ログインのクライアント リダイレクトを有効にします。
正解: **C,D (**コメントを発表する**)**
説明
SnowPro Advanced: Architectのドキュメントと学習リソースによると、これらの要件を満たすために使用すべきSnowflakeの機能は以下のとおりです。
* クラウド プロバイダーのプライベート接続を使用します。この機能により、お客様はデータをパブリック インターネットに公開することなく、独自のプライベート ネットワークから Snowflake に接続できます。Snowflake は、AWS PrivateLink、Azure Private Link、および Google Cloud Private Service Connect と統合されており、お客様の VPC または VNet から Snowflake エンドポイントへのプライベート接続を提供します。お客様は、トラフィックが Snowflake エンドポイントに到達する方法を制御でき、プロキシやパブリック IP アドレスが不要になります123。

* フェデレーション認証用のSSOを設定します。この機能により、お客様は既存のIDプロバイダー（IdP）を使用して、SnowflakeへのSSOアクセス用のユーザー認証を行うことができます。SnowflakeはほとんどのSAMLをサポートしています。
Okta、Microsoft AD FS、Google G Suite、Microsoft Azure Active Directory、OneLogin、Ping Identity、PingOneなど、2.0準拠のIdPベンダーが利用可能です。フェデレーション認証にSSOを設定することで、顧客は既存のユーザー認証情報とプロファイル情報を活用し、ユーザー名/パスワード認証よりも強力なセキュリティを提供できます4。
他の選択肢は、要件を満たしていないか、実現不可能であるため、誤りです。選択肢Aは、レプリケーションを設定すると、ユーザーが社内VPNの外部から接続できなくなるため、誤りです。
レプリケーションは、異なるリージョンやクラウド プラットフォームのアカウント間でデータベースをコピーできるようにする Snowflake の機能です。レプリケーションは、アカウントの接続性や可視性には影響しません 5。オプション B は、固有の企業 Tri-Secret Secure キーをプロビジョニングしても、ネットワークや認証の要件に影響しないため、誤りです。Tri-Secret Secure は、マスター キー、サービス キー、セキュリティ パスワードの 3 つの秘密を組み合わせて、Snowflake に保存されているデータの暗号化キーを顧客が管理できるようにする Snowflake の機能です。Tri-Secret Secure は、データの暗号化と復号化プロセスに対するセキュリティと制御の追加のレイヤーを提供しますが、プライベート接続や SSO を有効にするものではありません 6。オプション E は、VPN の外部でプロキシ Snowflake アカウントを使用し、ユーザー ログインのクライアント リダイレクトを有効にすることは、要件を満たすためのサポートされた方法でも推奨された方法でもないため、誤りです。クライアント リダイレクトは、接続文字列で指定されたものとは異なる Snowflake アカウントに顧客が接続できるようにする Snowflake の機能です。この機能は、リージョン間フェイルオーバー、データ共有、アカウント移行などのシナリオで役立ちますが、プライベート接続や SSO7 は提供しません。参照: AWS PrivateLink と Snowflake | Snowflake ドキュメント、Azure Private Link と Snowflake | Snowflake ドキュメント、Google Cloud Private Service Connect と Snowflake | Snowflake ドキュメント、フェデレーション認証と SSO の概要 | Snowflake ドキュメント、複数のアカウント間でのデータベースの複製 | Snowflake ドキュメント、Tri-Secret Secure | Snowflake ドキュメント、クライアント接続のリダイレクト | Snowflake ドキュメント

質問: **36**
既存のクラスタキーを持つテーブルに対して、代替クラスタキーを定義する機能を提供するのはどの機能ですか？
A. 外部テーブル
B. 実物大ビュー
C. 検索最適化
D. 結果キャッシュ
正解: **(**正解を表示します**)**
説明
マテリアライズド ビューは、既存のクラスタ キーを持つテーブルに対して代替クラスタ キーを定義できる機能です。マテリアライズド ビューは、Snowflake に格納される事前計算済みの結果セットであり、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 ビューは、ベース テーブルとは異なるクラスタ キーを持つことができ、これによりマテリアライズド ビューに対するクエリのパフォーマンスと効率が向上します。また、マテリアライズド ビューは、ベース テーブル データに対する集計、結合、フィルタをサポートできます。AUTO_REFRESH パラメータが true に設定されている場合、ベース テーブルの基となるデータが変更されると、マテリアライズド ビューは自動的に更新されます。
参考文献：
* マテリアライズドビュー | Snowflake ドキュメント

質問: **37**

ある企業は、Snowflakeアカウントをインターネットから見えないように、社内ネットワーク内に展開したいと考えています。同社はSnowflakeユーザー向けにVPNインフラストラクチャと仮想デスクトップインフラストラクチャ（VDI）を使用しています。また、ログイン管理時の冗長性を排除するため、VDI用に設定したログイン認証情報を再利用したいと考えています。
これらの要件を満たすには、Snowflakeのどの機能を使用すべきでしょうか？（2つ選択してください。）
A. ユーザーが社内VPNの外部から接続できるように、レプリケーションを設定します。
B. 独自の企業トライシークレットセキュアキーを提供します。
C. クラウドプロバイダーのプライベート接続を使用します。
D. フェデレーション認証のためのSSOを設定します。
E. VPN の外部でプロキシ Snowflake アカウントを使用し、ユーザー ログインのクライアント リダイレクトを有効にします。
正解: **(**正解を表示します**)**
SnowPro Advanced: Architectのドキュメントと学習リソースによると、これらの要件を満たすために使用すべきSnowflakeの機能は以下のとおりです。
クラウドプロバイダーのプライベート接続を使用します。この機能により、お客様はデータをパブリックインターネットに公開することなく、独自のプライベートネットワークからSnowflakeに接続できます。Snowflakeは、AWS PrivateLink、Azure Private Link、およびGoogle Cloud Private Service Connectと統合されており、お客様のVPCまたはVNetからSnowflakeエンドポイントへのプライベート接続を提供します。お客様は、トラフィックがSnowflakeエンドポイントに到達する方法を制御でき、プロキシやパブリックIPアドレスを必要とせずに済みます123。
フェデレーション認証用のSSOを設定します。この機能により、お客様は既存のIDプロバイダー（IdP）を使用して、SnowflakeへのSSOアクセス用のユーザー認証を行うことができます。Snowflakeは、Okta、Microsoft AD FS、Google G Suite、Microsoft Azure Active Directory、OneLogin、Ping Identity、PingOneなど、ほとんどのSAML 2.0準拠ベンダーをIdPとしてサポートしています。フェデレーション認証用のSSOを設定することで、お客様は既存のユーザー認証情報とプロファイル情報を活用し、ユーザー名よりも強力なセキュリティを提供できます。
/パスワード認証4。
他の選択肢は、要件を満たしていないか、実現不可能であるため、誤りです。選択肢Aは、レプリケーションを設定すると、ユーザーが社内VPNの外部から接続できなくなるため、誤りです。
レプリケーションは、異なるリージョンやクラウド プラットフォームのアカウント間でデータベースをコピーできるようにする Snowflake の機能です。レプリケーションは、アカウントの接続性や可視性には影響しません 5。オプション B は、固有の企業 Tri-Secret Secure キーをプロビジョニングしても、ネットワークや認証の要件に影響しないため、誤りです。Tri-Secret Secure は、マスター キー、サービス キー、セキュリティ パスワードの 3 つの秘密を組み合わせて、Snowflake に保存されているデータの暗号化キーを顧客が管理できるようにする Snowflake の機能です。Tri-Secret Secure は、データの暗号化と復号化プロセスに対するセキュリティと制御の追加のレイヤーを提供しますが、プライベート接続や SSO を有効にするものではありません 6。オプション E は、VPN の外部でプロキシ Snowflake アカウントを使用し、ユーザー ログインのクライアント リダイレクトを有効にすることは、要件を満たすためのサポートされた方法でも推奨された方法でもないため、誤りです。クライアント リダイレクトは、接続文字列で指定されたものとは異なる Snowflake アカウントに顧客が接続できるようにする Snowflake の機能です。この機能は、リージョン間フェイルオーバー、データ共有、アカウント移行などのシナリオで役立ちますが、プライベート接続や SSO7 は提供しません。参照: AWS PrivateLink と Snowflake | Snowflake ドキュメント、Azure Private Link と Snowflake | Snowflake ドキュメント、Google Cloud Private Service Connect と Snowflake | Snowflake ドキュメント、フェデレーション認証と SSO の概要 | Snowflake ドキュメント、複数のアカウント間でのデータベースの複製 | Snowflake ドキュメント、Tri-Secret Secure | Snowflake ドキュメント、クライアント接続のリダイレクト | Snowflake ドキュメント

質問**: 38**

プロバイダーアカウントとコンシューマーアカウント間でデータ共有が行われています。既に5つのテーブルが共有されており、コンシューマーロールにはインポート権限が付与されています。
プロバイダースキーマに新しいテーブルが追加された場合、どうなりますか？
A. 消費者は自動的に新しいテーブルを見ることができます。
B. 消費者側で輸入特権を再度付与した後、消費者は表を見ることができます。
C. コンシューマーは、プロバイダ側で共有に対して新しいテーブルに対するSELECT権限が付与された後にのみ、そのテーブルを見ることができます。
D. コンシューマーは、データベースとスキーマに対して USAGE が付与され、テーブルに対して SELECT がコンシューマー データベースに付与された後にテーブルを見ることができます。
正解: **C** (コメントを発表する)
Snowflake Secure Data Sharingでは、既存のスキーマに新しいテーブルを追加しても、自動的にコンシューマーから見えるようになるわけではありません。プロバイダは、共有に対して新しいテーブルへのSELECT権限を明示的に付与する必要があります（正解はC）。
これにより、意図的かつ管理されたデータ公開が保証されます。
プロバイダー側で権限が付与されると、既にインポートされた権限を持つコンシューマーロールは、コンシューマーアカウントに追加の権限を付与することなく、新しいテーブルを自動的に表示できるようになります。このような責任の分離は、Snowflakeのガバナンスにおける重要な原則です。
この質問は、SnowPro Architectがセキュアなデータ共有におけるプロバイダーとコンシューマーの責任について理解を深めるためのものです。

質問: **39**

図のアーキテクチャに基づいて、DB1のデータをTBL2にコピーするにはどうすればよいでしょうか？ 2つ選択してください)。

A.
```
use database DB1;
use schema SH1;
copy into DB1.SH2.TBL2
  from @STAGE1
 file_format = (format_name = FF_PIPE_1);
```

B.
```
use database DB1;
use schema SH2;
copy into TBL2
  from @STAGE1
 file_format = (format_name = FF_PIPE_1);
```

C.
```
use database DB1;
use schema SH2;
copy into DB1.SH2.TBL2
  from @STAGE1
 file_format = (format_name = DB1.SH1.FF_PIPE_1);
```

D.
```
use database DB1;
use schema SH2;
copy into DB1.SH2.TBL2
  from @DB1.SH1.STAGE1
 file_format = (format_name = FF_PIPE_1);
```

E.
```
use database DB1;
use schema SH2;
copy into TBL2
  from @DB1.SH1.STAGE1
 file_format = (format_name = DB1.SH1.FF_PIPE_1);
```

正解: **B,E** (コメントを発表する)

* 図のアーキテクチャは、DB1とDB2という2つのデータベースと、SH1とSH2という2つのスキーマを持つSnowflakeデータプラットフォームを示しています。DB1にはテーブルTBL1とステージSTAGE1が含まれています。DB2にはテーブルTBL2が含まれています。また、図には、ファイル形式FF PIPE 1を使用してSTAGE1からTBL2にデータをコピーするSQL言語で記述されたコードスニペットも示されています。

* DB1からTBL2にデータをコピーするには、提示された選択肢の中から2つの方法があります。

* オプション B: STAGE1 を参照する名前付き外部ステージを使用します。このオプションでは、DB1.SH1 の STAGE1 と同じ場所を指す外部ステージ オブジェクトを DB2.SH2 に作成する必要があります。外部ステージは、URL パラメーターで STAGE11 の場所を指定して CREATE STAGE コマンドを使用して作成できます。例:

SQLAIが生成したコードです。内容をよく確認の上、ご使用ください。詳細はFAQをご覧ください。

データベースDB2を使用する。

スキーマSH2を使用する。

ステージ作成 EXT_STAGE1
url=@DB1.SH1.STAGE1;
* 次に、COPY INTOコマンドを使用して、外部ステージ名をFROMパラメータで、ファイルフォーマット名をFILE FORMATパラメータで指定して、外部ステージからTBL2にデータをコピーできます。例：
SQLAIが生成したコードです。内容をよく確認の上、ご使用ください。詳細はFAQをご覧ください。
TBL2にコピー
@EXT_STAGE1 より
ファイル形式=(形式名=DB1.SH1.FF PIPE1);
* オプション E: クロス データベース クエリを使用して、TBL1 からデータを選択し、TBL2 に挿入します。このオプションでは、INSERT INTO コマンドと SELECT 句を使用して、DB1.SH1 の TBL1 からデータをクエリし、DB2.SH2 の TBL2 に挿入する必要があります。クエリでは、データベース名とスキーマ名を含む、テーブルの完全修飾名を使用する必要があります。例:
SQLAIが生成したコードです。内容をよく確認の上、ご使用ください。詳細はFAQをご覧ください。
データベースDB2を使用する。
スキーマSH2を使用する。
TBL2に挿入
select*fromDB1.SH1.TBL1;
* 他の選択肢は無効です。理由は以下のとおりです。
* オプション A: COPY INTO コマンドに無効な構文が使用されています。FROM パラメーターにはテーブル名を指定することはできません。ステージ名またはファイルロケーションのみを指定できます。
* オプション C: COPY INTO コマンドに無効な構文が使用されています。FILE FORMAT パラメーターではステージ名を指定できず、ファイル形式名またはオプション 2 のみを指定できます。
* オプション D: CREATE STAGE コマンドに無効な構文が使用されています。URL パラメータではテーブル名を指定できず、ファイルロケーションのみを指定できます。
1: ステージの作成 | Snowflake ドキュメント
2: テーブルへのコピー | Snowflakeドキュメント
3 :クロスデータベースクエリ | Snowflakeドキュメント

質問: **40**
Snowflakeにおけるトランザクション利用の特徴は何ですか？ 2つ選択してください。）
A. 明示的トランザクションには、DDL、DML、およびクエリステートメントを含めることができます。
B. 自動コミット設定はストアドプロシージャ内で変更できます。
C. トランザクションは、begin work ステートメントを実行することで明示的に開始でき、commit work ステートメントを実行することで明示的に終了できます。
D. トランザクションは、begin transaction ステートメントを実行することで明示的に開始でき、end transaction ステートメントを実行することで明示的に終了できます。
E. 明示的なトランザクションには、DML ステートメントとクエリ ステートメントのみを含める必要があります。すべての DDL ステートメントは、アクティブなトランザクションを暗黙的にコミットします。
正解: **(**正解を表示します**)**

Snowflakeでは、トランザクションとは、アトミックな単位として処理される一連のSQLステートメントのことです。トランザクション内のすべてのステートメントは、まとめて適用（コミット）されるか、取り消される（ロールバック）かのいずれかになります。
SnowflakeトランザクションはACID特性を保証します。トランザクションには読み取りと書き込みの両方を含めることができます1。
明示的トランザクションとは、BEGIN TRANSACTION、COMMIT、およびROLLBACKステートメントを使用して明示的に開始および終了されるトランザクションです。Snowflakeは、BEGIN WORKとBEGIN TRANSACTION、COMMIT WORKとROLLBACK WORKという同義語をサポートしています。明示的トランザクションには、DDL、DML、およびクエリステートメントを含めることができます。ただし、明示的トランザクションにはDMLステートメントとクエリステートメントのみを含める必要があります。これは、DDLステートメントがアクティブなトランザクションを暗黙的にコミットするためです。つまり、トランザクション内の前のステートメントによって行われた変更はすべて適用され、トランザクション内の後続のステートメントによって行われた変更は同じトランザクションの一部ではありません1。
他の選択肢が正しくない理由は以下のとおりです。
* B. 自動コミット設定はストアドプロシージャ内で変更できますが、これは Snowflake でのトランザクションの使用には影響しません。自動コミット設定は、各ステートメントが独自の暗黙的トランザクションで実行されるかどうかを決定します。自動コミットが有効になっている場合、各ステートメントは自動的にコミットされます。自動コミットが無効になっている場合、明示的な COMMIT または ROLLBACK が発行されるまで、各ステートメントは暗黙的トランザクションで実行されます。ストアドプロシージャ内で自動コミット設定を変更しても、ストアドプロシージャ内のステートメントにのみ影響し、ストアドプロシージャ外のステートメントには影響しません。
* C. トランザクションは、BEGIN WORK ステートメントを実行することで明示的に開始でき、COMMIT WORK ステートメントを実行することで明示的に終了できますが、これは Snowflake におけるトランザクションの使用上の特徴ではありません。これは、明示的なトランザクションを開始および終了するステートメントを記述する 1 つの方法にすぎません。Snowflake は、BEGIN TRANSACTION および COMMIT という同義語もサポートしており、BEGIN WORK および COMMIT WORK 1 よりもこれらの使用が推奨されます。
* D. トランザクションは、BEGIN TRANSACTION ステートメントを実行することで明示的に開始でき、END TRANSACTION ステートメントを実行することで明示的に終了できますが、これは Snowflake では有効な構文ではありません。
SnowflakeはEND TRANSACTIONステートメントをサポートしていません。明示的なトランザクションを終了する正しい方法は、COMMITまたはROLLBACKステートメントを使用することです。
参考文献：
* 1: トランザクション | Snowflake ドキュメント
* 2: AUTOCOMMIT | Snowflake ドキュメント

質問**: 41**
Snowflakeで動作するデータアーキテクチャを設計する際に、Snowflakeアーキテクトが3NFモデルではなくスター型スキーマモデルを使用する理由は何でしょうか？（2つ選択してください）。
A. Snowflake は 3NF データ モデルに含まれる結合を処理できません。
B. アーキテクトは、Snowflakeに保存されているデータからデータの重複を削除したいと考えています。
C. アーキテクトは、Snowflakeに生データを取り込むためのランディングゾーンを設計しています。
D. BIツールには、ユーザーがさまざまな次元にわたって事実を要約したり、要約からドリルダウンしたりできるデータモデルが必要です。
E. アーキテクトは、特定のエンドユーザーグループに、データのシンプルでフラットな単一ビューを提示したいと考えています。
正解**: D,E (**コメントを発表する**)**
スター型スキーマモデルは、単一のファクトテーブルと複数のディメンションテーブルで構成されるディメンションデータモデルの一種です。3NFモデルは、データの冗長性を排除し、参照整合性を保証する第3正規形に従うリレーショナルデータモデルの一種です。Snowflakeアー

キテクトは、Snowflakeで実行するデータアーキテクチャを設計する際に、3NFモデルではなくスター型スキーマモデルを使用する場合があります。その理由は次のとおりです。
スター型スキーマモデルは、BIツールなどで行われるような、異なる次元にわたってデータを集計・分割する分析クエリに適しています。一方、3NFモデルは、個々のレコードの挿入、更新、削除を行うトランザクションクエリに適しています。
スター型スキーマモデルは、結合が少なく、SQL文も複雑ではないため、3NFモデルよりもシンプルでクエリの実行速度が速い。一方、3NFモデルは結合が多く、SQL文も複雑であるため、クエリの実行速度が遅くなる。
スター型スキーマモデルは、ビジネスアナリストやデータサイエンティストなど、データの探索や視覚化を必要とする特定のエンドユーザーグループに対して、データのシンプルでフラットな単一ビューを提供できます。一方、3NFモデルは、アプリケーション開発者やデータエンジニアなど、データの保守や更新を必要とする別のエンドユーザーグループに対して、データのより詳細で正規化されたビューを提供できます。
Snowflakeにおいて、3NFモデルではなくスター型スキーマモデルを選択する正当な理由として、他の選択肢は認められません。
SnowflakeはANSI SQLをサポートし、複雑なクエリを効率的に最適化および実行できる強力なクエリエンジンを備えているため、3NFデータモデルに含まれる結合を処理できます。
アーキテクトは、スター型スキーマと3NFモデルの両方を使用して、Snowflakeに保存されているデータからデータの重複を排除できます。どちらのモデルもデータの整合性を確保し、データの異常を回避できるためです。ただし、トレードオフとして、スター型スキーマモデルはクエリのパフォーマンスを向上させるためにデータを非正規化するため、3NFモデルよりもデータの冗長性が高くなる可能性があり、一方、3NFモデルはデータの保守を容易にするためにデータを正規化するため、スター型スキーマモデルよりもデータの冗長性が低くなる可能性があります。
* 設計者は、スター型スキーマと3NFモデルの両方を使用して、Snowflakeに生データを取り込むためのランディングゾーンを設計できます。どちらのモデルも、さまざまな種類のデータソースとフォーマットに対応できるためです。
しかし、モデルの選択は、ランディングゾーンの目的と範囲によって異なる場合があり、例えば、一時的なストレージか永続的なストレージか、ステージングエリアかデータレイクか、単一ソース統合か複数ソース統合かなどによって決まる。
Snowflake Architect トレーニング
データモデリング :スター型スキーマとスノーフレーク型スキーマの理解
データボールト、スタースキーマ、第三正規形 :どのデータモデルを使用すべきか？
スター型スキーマとスノーフレーク型スキーマ :5つの重要な違い
ディメンションデータモデリング - Snowflakeスキーマ
スター型スキーマとスノーフレーク型スキーマの比較

質問: 42
A table contains five columns and it has millions of records. The cardinality distribution of the columns is shown below:

Column	Number of Distinct Values
C1	10,790
C2	108
C3	302,605
C4	1.117,736
C5	2.205,400

Column C4 and C5 are mostly used by SELECT queries in the GROUP BY and ORDER BY clauses. Whereas columns C1, C2 and C3 are heavily used in filter and join conditions of SELECT queries.
The Architect must design a clustering key for this table to improve the query performance.
Based on Snowflake recommendations, how should the clustering key columns be ordered while defining the multi-column clustering key?
A. C5, C4, C2
B. C3, C4, C5
C. C1, C3, C2
D. C2, C1, C3
正解**: C (**コメントを発表する**)**
According to the Snowflake documentation, the following are some considerations for choosing clustering for a table1:
Clustering is optimal when either:
You require the fastest possible response times, regardless of cost.
Your improved query performance offsets the credits required to cluster and maintain the table.
Clustering is most effective when the clustering key is used in the following types of query predicates:
Filter predicates (e.g. WHERE clauses)
Join predicates (e.g. ON clauses)
Grouping predicates (e.g. GROUP BY clauses)
Sorting predicates (e.g. ORDER BY clauses)
Clustering is less effective when the clustering key is not used in any of the above query predicates, or when the clustering key is used in a predicate that requires a function or expression to be applied to the key (e.g. DATE_TRUNC, TO_CHAR, etc.).
For most tables, Snowflake recommends a maximum of 3 or 4 columns (or expressions) per key. Adding more than 3-4 columns tends to increase costs more than benefits.
Based on these considerations, the best option for the clustering key columns is C. C1, C3, C2, because:
These columns are heavily used in filter and join conditions of SELECT queries, which are the most effective types of predicates for clustering.
These columns have high cardinality, which means they have many distinct values and can help reduce the clustering skew and improve the compression ratio.
These columns are likely to be correlated with each other, which means they can help co-locate similar rows in the same micro-partitions and improve the scan efficiency.
These columns do not require any functions or expressions to be applied to them, which means they can be directly used in the predicates without affecting the clustering.

質問**: 43**
ストリームに基づいてタスクを作成するために、以下のDDLコマンドが使用されました。

```
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、このタスク専用に使用されていると仮定した場合、次のうち正しい記述はどれですか？

A. 倉庫MY_WHは、ストリームをチェックするために5分ごとにアクティブになります。

B. 倉庫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. 倉庫MY_WHは決して停止しません。

D. 倉庫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。

正解: **B** (コメントを発表する)

説明

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ているため、ストリームに新しいデータがある場合にのみタスクが実行されるためです。さらに、ウェアハウスは auto_suspend - 60 に設定されているため、60 秒間操作がないと自動的に一時停止します。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。参照:

* [タスクの作成 | Snowflake ドキュメント]
* [ストリームとタスクの使用 | Snowflakeドキュメント]
* [倉庫の作成 | Snowflake ドキュメント]

質問: **44**

Snowflakeで動作するデータアーキテクチャを設計する際に、Snowflakeアーキテクトが3NFモデルではなくスター型スキーマモデルを使用する理由は何でしょうか？ 2つ選択してください)。

A. Snowflake は 3NF データ モデルに含まれる結合を処理できません。

B. アーキテクトは、Snowflakeに保存されているデータからデータの重複を削除したいと考えています。

C. アーキテクトは、特定のエンドユーザーグループに、データのシンプルでフラットな単一ビューを提示したいと考えています。

D. BIツールには、ユーザーがさまざまな次元にわたって事実を要約したり、要約からドリルダウンしたりできるデータモデルが必要です。

E. アーキテクトは、Snowflakeに生データを取り込むためのランディングゾーンを設計しています。

正解: (正解を表示します)

質問: **45**

データ構築ツール (dbt)は、どのような機能を提供しますか？ 2つ選択してください。)

A. データ読み込み

B. データテスト

C. データ可視化
D. データ変換
E. データ複製
正解: **(**正解を表示します**)**
dbt は、データウェアハウス内で動作する、変換に特化した分析エンジニアリングツールです。SQL ベースのデータ変換を可能にし、モデル間の依存関係を管理します (回答 D)。また、dbt は組み込みのデータテスト機能を提供し、一意性、NULL 制約、参照整合性チェックなど、データ品質のテストを定義および実行できるようにします (回答 B)。
dbt は、データのロード、可視化、レプリケーションは処理しません。これらの機能は通常、データ取り込みツール、BI プラットフォーム、またはレプリケーション サービスによって処理されます。SnowPro Architect の候補者は、Snowflake エコシステムにおける dbt の役割と、それが最新の ELT アーキテクチャにどのように適合するかを理解していることが求められます。

質問: **46**
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る。
この要件を満たすには、ユーザーの役割にどの権限が必要ですか？ 2つ選択してください。）
A. 輸入シェア
B. データベースを作成します。
C. 輸入特権
D. 作成 -共有;
E. データベースのインポート;
正解: **(**正解を表示します**)**

質問: **47**
ユーザーがセッションに対してプライマリロールとセカンダリロールを有効にしました。
Snowflakeでセカンダリロールを使用する際に、ユーザーがSQLアクションの一部として使用することを禁止されている操作は何ですか？
A. 挿入
B. 作成
C. 削除
D. 切り詰める
正解: **(**正解を表示します**)**
Snowflakeでは、ユーザーがセッション中にセカンダリロールをアクティブ化すると、DDL（データ定義言語）操作に関連する特定の権限が制限されます。DDL操作に含まれるCREATEステートメントは、セカンダリロールでは実行できません。この制限は、ロールベースのアクセス制御を徹底し、スキーマの変更が慎重に管理されるようにするためのものです。スキーマの変更は通常、データベース構造を変更する明示的な権限を持つプライマリロールにのみ許可されます。

参考資料 :Snowflakeのセキュリティおよびアクセス制御に関するドキュメント。セッション管理におけるプライマリロールとセカンダリロールの制限事項と機能について詳述しています。

質問: **48**
アーキテクトは、アプリケーションサーバーに追加のソフトウェアをインストールすることなく、Snowflakeとの間でデータの読み書きを行う必要があるアプリケーションを統合しようとしています。
この要件を満たすにはどうすればよいでしょうか？
A. SnowSQLを使用する。
B. Snowflake SQL REST API を使用します。
C. Snowpipe REST API を使用します。
D. Snowflake ODBC ドライバーを使用します。
正解: **(**正解を表示します**)**

質問: **49**
クエリの1つが完了するまでに時間がかかっています。クエリプロファイラを開くと、大量のデータがリモートディスクに流出していることがわかります（リモートストレージに流出したバイト数）。
その原因は何でしょうか？
A. 仮想倉庫に接続されているディスクの数が処理に十分ではありません
B. データを保持するために使用されるAWSバケットのサイズがクエリに対して不十分です
C. 操作を実行するサーバーで使用可能なメモリ容量が、中間結果を保持するのに十分でない可能性があります。
正解: **(**正解を表示します**)**

質問: **50**
データアナリストのグループにアナリストの役割が付与されました。彼らは、テーブル、ビュー、その他のオブジェクトを作成および変更して独自のデータをロードできるSnowflakeデータベースを必要としています。アナリストは、自分の役割以外のSnowflakeユーザーにこのデータへのアクセス権を付与する権限を持たないようにする必要があります。
これらの要件はどのように満たすべきでしょうか？
A. データベース内のすべてのスキーマを管理アクセス スキーマにし、SYSADMIN が所有するようにし、作成する必要のあるオブジェクトの種類ごとに、各スキーマに対する作成権限を ANALYST_ROLE に付与します。
B. ANALYST_R0LEにデータベースの所有権を付与しますが、ANALYST_ROLEがアカウントに対してMANAGE GRANTS権限を持たないようにしてください。
C. SYSADMIN にデータベースの所有権を付与しますが、データベースに対するスキーマ作成権限は ANALYST_ROLE に付与します。
D. データベースに対する所有権を ANALYST_ROLE に付与しますが、データベース権限内の将来の [オブジェクト タイプ] の所有権は SYSADMIN に付与します。
正解: **A (**コメントを発表する**)**

質問: **51**
ある企業は、AWS us-east-1リージョンにACCOUNTAという名前のSnowflakeアカウントを保有しています。この企業は、MARKET_DBという名前のSnowflakeデータベースにマーケティングデータを保存しています。同社のビジネスパートナーの1社は、Azure East US 2リージョンに

PARTNERBという名前のアカウントを保有しています。マーケティング目的で、同社はMARKET_DBデータベースをパートナーアカウントと共有することに同意しました。
アカウントPARTNERBがMARKET_DBデータベースからデータを取得するために、以下の手順のうちどれを必ず実行する必要がありますか？
A. Azure East US 2 リージョンに新しいアカウント (AZABC123 という名前) を作成します。アカウント ACCOUNTA からデータベース MARKET_DB の共有を作成し、この共有から AWS us-east-1 リージョンにローカルで新しいデータベースを作成し、この新しいデータベースを AZABC123 アカウントにレプリケートします。次に、PARTNERB アカウントとのデータ共有を設定します。
B. アカウントACCOUNTAからデータベースMARKET_DBの共有を作成し、AWS us-east-1リージョンにこの共有から新しいデータベースをローカルに作成します。次に、このデータベースをプロバイダーとして設定し、PARTNERBアカウントと共有します。
C. Azure East US 2 リージョンに新しいアカウント (AZABC123 という名前) を作成します。アカウント ACCOUNTA からデータベース MARKET_DB を AZABC123 にレプリケートし、このアカウントから PARTNERB アカウントへのデータ共有を設定します。
D. データベース MARKET_DB の共有を作成し、AWS us-east-1 リージョンでこの共有から新しいデータベースをローカルに作成します。次に、このデータベースをパートナーのアカウント PARTNERB にレプリケートします。
正解: **(**正解を表示します**)**
* Snowflakeは、アカウントレプリケーションと共有レプリケーション機能を使用して、リージョン間およびクラウドプラットフォーム間でのデータ共有をサポートします。アカウントレプリケーションを使用すると、ソースアカウントから同じ組織内の1つ以上のターゲットアカウントにオブジェクトを複製できます。共有レプリケーションを使用すると、ソースアカウントから同じ組織内の1つ以上のターゲットアカウントに共有を複製できます1。
* AWS us-east-1 リージョンの ACCOUNTA アカウントにある MARKET_DB データベースのデータを、Azure East US 2 リージョンの PARTNERB アカウントと共有するには、以下の手順を実行する必要があります。
* Azure East US 2 リージョンに新しいアカウント (AZABC123 という名前) を作成します。このアカウントは、ソース アカウントとターゲット アカウント間のブリッジとして機能します。新しいアカウントは、組織 2 を使用して ACCOUNTA アカウントにリンクする必要があります。
* アカウントレプリケーション機能を使用して、ACCOUNTA アカウントから MARKET_DB データベースを AZABC123 アカウントに複製します。これにより、AZABC123 アカウントに、ACCOUNTA アカウントのプライマリ データベースのレプリカであるセカンダリ データベースが作成されます。
* AZABC123 アカウントから、共有レプリケーション機能を使用して PARTNERB アカウントへのデータ共有を設定します。これにより、AZABC123 アカウントにセカンダリ データベースの共有が作成され、PARTNERB アカウントにアクセス権が付与されます。PARTNERB アカウントは、共有からデータベースを作成し、データをクエリできます。
したがって、選択肢Cが正解です。
リージョンとクラウド プラットフォーム間での共有の複製: 組織とアカウントの操作: 複数のアカウント間でのデータベースの複製: 複数のアカウント間での共有の複製

質問**: 52**
Snowflake上で大規模な結合処理を実行しています。中規模のデータウェアハウスで実行したところ、完了までに1時間近くかかりました。その後、大規模なデータウェアハウスで実行してみましたが、パフォーマンスは改善されませんでした。
この原因として最も考えられるのは何でしょうか。
A. 自動停止値を低く設定した倉庫のため、倉庫が頻繁に停止しています。
B. 倉庫のメモリが不足しています
C. データに歪みが生じている可能性があります。列内の値のうち1つが、他の値よりも著しく大きい場合です。

正解: **C** (コメントを発表する)

質問: **53**
生テーブル上に作成されたテーブルストリームをクエリし、行のサブセットを複数のテーブルに挿入するタスクを作成しています。以下の手順に従って作業を進めていますが、タスクを再開するステップに到達した際に、以下のようなエラーメッセージが表示されました。
なぜこのエラーが発生するのか、また、必要な権限を付与できるのは誰なのか？

このエラーを発生させるための手順
-- 生の JSON データを保存するランディング テーブルを作成します。
-- Snowpipe はこのテーブルにデータをロードできます。create or replace table raw (var variant);
-- ランディングテーブルへの挿入をキャプチャするためのストリームを作成します。
-- タスクはこのストリームから一連の列を消費します。create or replace stream rawstream1 on table raw;
-- ランディングテーブルへの挿入をキャプチャするための2つ目のストリームを作成します。
-- 2 番目のタスクは、このストリームから別の列セットを消費します。create or replace stream rawstream2 on table raw;
-- 生データで識別されたオフィス訪問者の名前を格納するテーブルを作成します。create or replace table names (id int, first_name string, last_name string);
-- 生データで特定されたオフィス訪問者の訪問日を格納するテーブルを作成します。
CREATE OR REPLACE TABLE visits (id int, dt date);
-- rawstream1 ストリームから新しい名前レコードを names テーブルに挿入するタスクを作成します
-- ストリームにレコードが含まれている場合、1分ごとに。
-- 'mywh' 倉庫を、あなたのロールが USAGE 権限を持つ倉庫に置き換えてください。create or replace task raw_to_names
倉庫 = etl_wh
スケジュール = '1分'
いつ
system$stream_has_data('rawstream1')
として
名前nにマージする
using (select var:id id, var:fname fname, var:lname lname from rawstream1) r1 on n.id = to_number(r1.id)
一致した場合、n.first_name = r1.fname、n.last_name = r1.lname を更新します。
一致しない場合は、(id、first_name、last_name) の値 (r1.id、r1.fname、r1.lname) を挿入します。
;
-- rawstream1 ストリームからの訪問記録を visits テーブルにマージする別のタスクを作成します
-- ストリームにレコードが含まれている場合、1分ごとに。
-- 新しいIDを持つレコードが訪問テーブルに挿入されます。

-- visits テーブルに存在する ID を持つレコードは、テーブルの DT 列を更新します。
-- 'mywh' 倉庫を、あなたのロールが USAGE 権限を持つ倉庫に置き換えてください。create or replace task raw_to_visits
倉庫 = etl_wh スケジュール = '1 分' の場合
system$stream_has_data('rawstream2')
訪問に統合 v
using (select var:id id, var:visit_dt visit_dt from rawstream2) r2 on v.id = to_number(r2.id) when matched then update set v.dt = r2.visit_dt
一致しない場合は、(id, dt) の値 (r2.id, r2.visit_dt) を挿入します。
両方のタスクを再開します。
alter task raw_to_names resume;
A. タスクを再開するために使用するロールには、EXECUTE TASK 権限がありません。この権限をロールに付与できるのは ACCOUNTADMIN のみです。
B. タスクを再開するために使用されたロールには、EXECUTE TASK権限がありません。その権限をロールに付与できるのはTASK OWNERのみです。
C. タスクを再開するために使用するロールには、EXECUTE TASK 権限がありません。SECURITYADMIN と ACCOUNTADMIN の両方が、そのロールに権限を付与できます。
正解: **(**正解を表示します**)**

質問: **54**
アカウントにはfin_dbとhr_dbという2つのデータベースがあり、それぞれ給与データと従業員データが格納されています。社内の会計担当者とアナリストは、業務を遂行するためにこれらのデータベース内のオブジェクトに対して異なるアクセス権限を必要とします。会計担当者はfin_dbへの読み書きアクセスが必要ですが、hr_dbは人事担当者が管理しているため、読み取り専用アクセスのみで十分です。
アーキテクトは、人事部に勤務する特定の従業員に対して、読み取り専用のロールを作成する必要があります。
このロールには、どの権限セットを付与する必要がありますか？
A. データベースhr_dbに対するUSAGE、データベースhr_db内のすべてのスキーマに対するUSAGE、データベースhr_db内のすべてのテーブルに対するSELECT
B. データベースhr_dbに対するUSAGE、データベースhr_db内のすべてのスキーマに対するSELECT、データベースhr_db内のすべてのテーブルに対するSELECT
C. データベースhr_dbに対するMODIFY、データベースhr_db内のすべてのスキーマに対するUSAGE、データベースhr_db内のすべてのテーブルに対するUSAGE
D. データベースhr_dbに対するUSAGE、データベースhr_db内のすべてのスキーマに対するUSAGE、データベースhr_db内のすべてのテーブルに対するREFERENCES
正解: **(**正解を表示します**)**
* 人事部で働く特定の従業員向けに読み取り専用ロールを作成するには、そのロールにhr_dbデータベースに対する以下の権限が必要です。
* データベースの使用権限: これにより、ロールはデータベースにアクセスし、そのスキーマとオブジェクトを表示できるようになります。
* データベース内のすべてのスキーマに対する使用権限: これにより、ロールはスキーマにアクセスし、そのオブジェクトを表示できるようになります。
* データベース内のすべてのテーブルに対する SELECT: これにより、ロールはテーブル内のデータをクエリできるようになります。
* オプションAが正解です。なぜなら、hr_dbデータベースに対する読み取り専用ロールに必要な最小限の権限を付与するからです。
* オプションBは、スキーマに対するSELECT権限が有効な権限ではないため、誤りです。スキーマはUSAGE権限とCREATE権限のみをサポートしています。

* オプションCは、データベースに対するMODIFY権限が有効な権限ではないため、誤りです。データベース
は、USAGE、CREATE、MONITOR、OWNERSHIPの権限のみをサポートしています。さらに、テーブルに対するUSAGE権限だけでは、データのクエリを実行するには不十分です。テーブルは、SELECT、INSERT、UPDATE、DELETE、TRUNCATE、REFERENCES、OWNERSHIPの権限をサポートしています。
* オプション D は、テーブルの REFERENCES はデータのクエリには関係がないため、誤りです。
REFERENCES権限により、ロールはテーブルに対して外部キー制約を作成できます。
参考文献：
* : https://docs.snowflake.com/en/user-guide/security-access-control-privileges.html#database-privileges
* : https://docs.snowflake.com/en/user-guide/security-access-control-privileges.html#schema-privileges
* : https://docs.snowflake.com/en/user-guide/security-access-control-privileges.html#table-privileges

質問: **55**
Company A has recently acquired company B. The Snowflake deployment for company B is located in the Azure West Europe region.
As part of the integration process, an Architect has been asked to consolidate company B's sales data into company A's Snowflake account which is located in the AWS us-east-1 region.
How can this requirement be met?
A. Replicate the sales data from company B's Snowflake account into company A's Snowflake account using cross-region data replication within Snowflake. Configure a direct share from company B's account to company A's account.
B. Export the sales data from company B's Snowflake account as CSV files, and transfer the files to company A's Snowflake account. Import the data using Snowflake's data loading capabilities.
C. Migrate company B's Snowflake deployment to the same region as company A's Snowflake deployment, ensuring data locality. Then perform a direct database-to-database merge of the sales data.
D. Build a custom data pipeline using Azure Data Factory or a similar tool to extract the sales data from company B's Snowflake account. Transform the data, then load it into company A's Snowflake account.
正解: **(**正解を表示します**)**
B社の売上データをA社のSnowflakeアカウントに統合するという要件を満たす最善の方法は、Snowflakeのリージョン間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、異なるリージョンやクラウドプラットフォーム間で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B社のアカウントからAWS us-east-1リージョンのA社のアカウントに売上データをレプリケートすることで、データは同期され、利用可能になります。データレプリケーションを有効にするには、アカウントをリンクし、ORGADMINロールを持つユーザーがレプリケーションを有効にする必要があります。次に、レプリケーショングループを作成し、売上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ために、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設定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で安全です。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。
参照：
地域やクラウドプラットフォームをまたい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
複製に関する考慮事項
アカウントオブジェクトの複製

質問: **56**
ある企業は、Snowflakeアカウントをインターネットから見えないように、社内ネットワーク内に展開したいと考えています。同社はSnowflakeユーザー向けにVPNインフラストラクチャと仮想デスクトップインフラストラクチャ（VDI）を使用しています。また、ログイン管理時の冗長性を排除するため、VDI用に設定したログイン認証情報を再利用したいと考えています。
これらの要件を満たすには、Snowflakeのどの機能を使用すべきでしょうか？（2つ選択してください。）
A. 独自の企業トライシークレットセキュアキーを提供します。
B. フェデレーション認証のためのSSOを設定します。
C. クラウドプロバイダーのプライベート接続を使用する。
D. VPN の外部で Snowflake プロキシ アカウントを使用し、ユーザー ログインのクライアント リダイレクトを有効にします。
E. ユーザーが社内VPNの外部から接続できるように、レプリケーションを設定します。
正解: **(**正解を表示します**)**

質問: **57**
Snowflakeでクラスタキーの優先順位付けを行う際に推奨されるベストプラクティスはどれですか？（2つ選択してください。）
A. 結合述語で頻繁に使用される列を選択します。
B. クラスタリングキーをサポートし、コスト効率を高めるために、カーディナリティの低い列を選択します。
C. 一意の行数を最大にするには、ナノ秒を含むTIMESTAMP列を選択してください。
D. 選択フィルタで最も頻繁に使用されるクラスタ列を選択します。
E. GROUP BY句で実際に使用されているクラスタ列を選択します。
正解: **A,D (**コメントを発表する**)**
Snowflakeのドキュメントによると、クラスタリングキーを選択する際のベストプラクティスは以下のとおりです。
結合述語で頻繁に使用される列を選択してください。これにより、スキャンおよび結合する必要のあるマイクロパーティションの数を減らし、結合パフォーマンスを向上させることができます。
* 選択フィルタで最も頻繁に使用される列を選択してください。これにより、フィルタ条件に一致しないマイクロパーティションをスキップできるため、スキャン効率が向上します。
性別や国など、カーディナリティの低い列をクラスタリングキーとして使用することは避けてください。クラスタリング精度が低下し、メンテナンスコストが高くなる可能性があります。
ナノ秒単位のTIMESTAMP列は、カーディナリティが非常に高く、他の列との相関が低い傾向があるため、使用を避けてください。これにより、クラスタリングの精度が低下したり、メンテナンスコストが高くなったりする可能性があります。
* 重複値やNULL値を含む列の使用は避けてください。これらの列はクラスタリングに偏りを生じさせ、枝刈りの効果を低下させる可能性があります。
クエリで複数のフィルタまたは結合述語を使用する場合は、複数の列でクラスタリングを実行します。これにより、より多くのマイクロパーティションが削除され、圧縮率が向上する可能性が高まります。
クラスタリングは、特に小規模または中規模のテーブル、あるいはクエリや更新の頻度が低いテーブルでは、必ずしも有効とは限りません。クラスタリングには、データの初期クラスタリングと、その後のクラスタリング維持のための追加コストが発生する可能性があります。
参考文献：
* クラスタリングキーとクラスタ化テーブル | Snowflake ドキュメント
* [テーブルのクラスタリングを選択する際の考慮事項 | Snowflakeドキュメント]

質問: **58**
アーキテクトは、外部テーブルを使用して外部ステージからデータを読み込み、変更されたデータに対して変換を実行し、ディメンションテーブルと結合し、結果をターゲットテーブルにロードする自動化されたETLパイプラインを構築したいと考えています。
建築家は何を使うべきか？
A. ストリームとタスク
B. 動的テーブル
C. マテリアライズドビュー
D. 自動取り込み機能付きスノーパイプ
正解: **(**正解を表示します**)**
Snowflakeでは、ストリームとタスクによって、自動化された増分型ETLパイプラインを構築するためのネイティブフレームワークが提供されています。
ストリームは外部テーブル またはステージングテーブル）の変更を追跡し、タスクは変換ロジックの実行とターゲットテーブルへのロードを調整します（回答A）。
このアプローチは、複雑な変換、ディメンションテーブルとの結合、制御されたスケジュール設定またはイベント駆動型実行をサポートします。マテリアライズドビューは任意のテーブルとの結合を実行できず、複雑なETLには適していません。動的テーブルは変換を簡素化しますが、外部テーブルから変更データを直接取り込むようには設計されていません。Snowpipeは取り込みのみに特化しており、下流の変換はサポートしていません。
SnowPro Architectの試験では、ストリームやタスクをいつ使用すべきか、あるいは動的テーブルのような新しい抽象化をいつ使用すべきかという理解度が頻繁に問われます。

質問: **59**
アーキテクトがQUERY関数を使用してパフォーマンスの低いクエリのトラブルシューティングを行っています。アーキテクトは、COMPILATION_TIMEがEXECUTION_TIMEよりも大きいことに気づきました。
その理由は？
A. このクエリは非常に大きなデータセットを処理しています。
B. クエリのロジックが過度に複雑です。
C. クエリは実行待ちのキューに追加されました。
D. クエリはリモートストレージから読み取っています
正解: **(**正解を表示します**)**
正解はBです。コンパイル時間とは、オプティマイザがクエリを効率的に実行するための最適なクエリプランを作成するのにかかる時間です。コンパイル時間は、テーブル数、列数、結合数、フィルタ数、集計数、サブクエリ数など、クエリの複雑さに依存します。クエリが複雑になるほど、コンパイルに時間がかかります。
オプションAは誤りです。クエリ処理時間はデータセットのサイズではなく、仮想ウェアハウスのサイズによって影響を受けます。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自動的にスケーリングし、クエリ実行を並列化します。データセットのサイズは実行時間に影響を与える可能性がありますが、コンパイル時間には影響しません。
選択肢Cは誤りです。クエリキュー時間はコンパイル時間や実行時間の一部ではありません。クエリキュー時間は、クエリが実行を開始する前にウェアハウスのスロットを待機する時間を示す独立した指標です。クエリキュー時間は、ウェアハウスの負荷、同時実行性、および優先度設定によって異なります。

オプションDは誤りです。クエリのリモートI/O時間はコンパイル時間や実行時間の一部ではありません。これは、クエリがS3やAzure Blob Storageなどのリモートストレージからデータを読み取るのに要する時間を示す別のメトリックです。クエリのリモートI/O時間は、ネットワーク遅延、帯域幅、およびキャッシュ効率に依存します。参照：
Snowflakeにおけるコンパイル時間が実行時間よりも長くなる理由を理解する :この記事では、Snowflakeにおけるクエリパフォーマンスを測定する上で、合計時間（コンパイル時間＋実行時間）が重要な指標となる理由を説明します。クエリの複雑さやテーブル数、列数など、コンパイル時間が長くなる原因についても解説します。
実行時間の調査 :このドキュメントでは、Snowsight を使用するか、ACCOUNT_USAGE スキーマのビューに対してクエリを作成することで、クエリとタスクの過去のパフォーマンスを調査する方法について説明します。また、実行時間、コンパイル時間、実行時間、キュー時間、リモート I/O 時間など、クエリのパフォーマンスに影響を与えるさまざまなメトリックとディメンションについても説明します。
「コンパイル時間」とは何か、そしてそれを最適化するにはどうすればよいか？ :このコミュニティ投稿では、クエリロジックの簡素化、ビューや共通テーブル式の使用、不要な列や結合の回避など、コンパイル時間を短縮するためのヒントとベストプラクティスを紹介します。

質問: **60**
デフォルトのスケーリングポリシーを使用した場合、マルチクラスタウェアハウスはいつシャットダウンしますか？
A. 5～6回連続してチェックが成功した後 (1 分間隔で実行)、最も負荷の低いクラスターの負荷を、クラスターを再度起動することなく他のクラスターに再分配できるかどうかを判断します。
B. クエリ実行直後
C. 2～3回連続してチェックが成功した後（1分間隔で実行）、最も負荷の低いクラスターの負荷を、クラスターを再度起動することなく他のクラスターに再分配できるかどうかを判断します。
正解: **(**正解を表示します**)**

質問: **61**
アーキテクトは、SECURITY_LOGSスキーマ内にVPN_ACCESS_LOGSテーブルを持ち、そこには接続と切断のタイムスタンプ、ユーザーのユーザー名、および概要統計情報が含まれています。
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しょうか？
A. VPN_ACCESS_LOGS の OWNERSHIP ロールを引き受け、SECURITY_LOGS スキーマに SEARCH OPTIMIZATION を追加します。
B. セキュリティログスキーマで、検索最適化の追加を含むすべての権限を持つロールを引き受けます。
C. VPN_ACCESS_LOGS に対して ALL PRIVILEGES を持つロールを引き受け、SECURITY_LOGS スキーマに SEARCH OPTIMIZATION を追加します。
D. 将来のテーブルに対してOWNERSHIPロールを引き受け、SECURITY_LOGSスキーマに対して検索最適化を追加します。
正解: **(**正解を表示します**)**

質問: **62**

ある大手製造会社は、事業部門全体で12個の個別のSnowflakeアカウントを運用している。同社は、サプライチェーンの最適化を支援し、複数のベンダーとの購買交渉力を高めるために、データ共有のレベルを向上させたいと考えている。
同社のSnowflakeアーキテクトは、各事業部門が共有する情報を決定できると同時に、構成と管理にかかる労力を最小限に抑えるソリューションを設計する必要がある。欧州を拠点とする一部の部門を除き、ほとんどの事業部門は同じクラウド環境でSnowflakeアカウントを使用している。
Snowflakeが推奨するベストプラクティスによると、これらの要件はどのように満たすべきでしょうか？
A. 欧州のアカウントをグローバルリージョンに移行し、接続されたグラフアーキテクチャで共有を管理します。データ交換をデプロイします。
B. 欧州アカウントのデータ共有と組み合わせたプライベートデータ交換を導入する。
C. すべてのセキュアビューでinvoker_share()が使用されていることを確認して、Snowflake Marketplaceにデプロイします。
D. プライベートデータ交換を導入し、レプリケーションを使用して交換内でヨーロッパのデータ共有を可能にします。
正解: **B** (コメントを発表する)
説明
Snowflake が推奨するベストプラクティスによると、大規模製造企業の要件は、欧州アカウントのデータ共有と組み合わせてプライベートデータエクスチェンジを導入することで満たされるはずです。プライベートデータエクスチェンジは、Snowflake Data Cloud プラットフォームの機能で、組織間でデータを安全かつ統制された方法で共有することを可能にします。これにより、Snowflake の顧客は独自のデータハブを作成し、組織の他の部門や外部パートナーを招待してデータセットにアクセスし、提供することができます。プライベートデータエクスチェンジは、エクスチェンジで共有されるデータに対して、一元管理、きめ細かなアクセス制御、およびデータ使用状況のメトリクスを提供します 1。データ共有は、データをコピーしたり移動したりすることなく、Snowflake アカウント間でデータを安全かつ直接的に共有する方法です。データ共有を使用すると、データプロバイダーは、アカウント内の選択したオブジェクトに対する権限を、他のアカウントの 1 つ以上のデータコンシューマーに付与できます 2。プライベートデータエクスチェンジとデータ共有を組み合わせて使用することで、企業は次のメリットを実現できます。
* 各事業部門は、共有するデータを決定し、プライベートデータ交換プラットフォームに公開できます。公開されたデータは、交換プラットフォームの他のメンバーが検索してアクセスできます。これにより、複数のデータ共有関係や構成を管理する手間と複雑さが軽減されます。
* 企業は、同じクラウド環境にある既存のSnowflakeアカウントを活用してプライベートデータエクスチェンジを作成し、メンバーを招待することができます。これにより、移行とセットアップのコストを最小限に抑え、既存のSnowflakeの機能とセキュリティを活用できます。
* 同社はデータ共有機能を利用して、異なる地域やクラウドプラットフォームにある欧州のアカウントとデータを共有できます。これにより、同社はデータ主権とプライバシーに関する地域および規制上の要件を遵守しつつ、組織全体でのデータ連携を実現できます。
同社はSnowflake Data Cloudプラットフォームを利用して、共有データに対するデータ分析や変換を実行できるほか、他のデータソースやアプリケーションとの統合も可能です。これにより、サプライチェーンを最適化し、複数のベンダーとの購買交渉力を高めることができます。
他のオプションは、要件を満たしていないか、ベストプラクティスに従っていないため、誤りです。オプションAは、欧州アカウントをグローバルリージョンに移行すると、データ主権とプライバシー規制に違反する可能性があり、データエクスチェンジを導入しても、企業が必要とするレベルの制御と管理が実現できない可能性があるため、誤りです。オプションCは、Snowflake Marketplaceに導入すると、企業のデータが望ましくない消費者に公開される可能性があり、セキュアビューでinvoker_share()を使用しても、望ましいレベルのセキュリティとガバナンスが実現できない可能性があるため、誤りです。オプションDは、エクスチェンジで欧州のデータ共有を可能にするためにレプリケーションを使用すると、追加のコストと複雑さが発生する可能性があり、代わりにデータ共有を使用できる場合は不要になる可能性があるため、誤りです。参照：プライベートデータエクスチェンジ｜Snowflakeドキュメント、セキュアデータ共有の概要｜Snowflakeドキュメント

質問: **63**
アーキテクトは、COPY INTO で使用される Snowflake のデータアンロード戦略を設計する必要があります。

<location> コマンド。
どの設定が有効ですか？
A. ファイルの場所: Snowflake の内部ロケーション。ファイル形式: CSV、XML。ファイルエンコーディング: UTF-8。
暗号化 :128ビット
B. ファイルの場所: Amazon S3。ファイル形式: CSV、JSON。ファイルエンコーディング: Latin-1 (ISO-8859)。
暗号化 :128ビット
C. ファイルの場所: Google Cloud Storage。ファイル形式: Parquet。ファイルエンコーディング: UTF-8 圧縮:
gzip
D. ファイルの場所: Azure ADLS。ファイル形式: JSON、XML、Avro、Parquet、ORC。圧縮: bzip2。暗号化: ユーザー指定のキー
正解: **(**正解を表示します**)**
Snowflakeでのデータアンロードの設定において、提示された選択肢の中で有効なのは 「C」です。これは、SnowflakeがCOPY INTOを使用してGoogle Cloud Storageにデータをアンロードすることをサポートしているためです。
<location> コマンドに特定の構成を指定します。オプション C に記載されている構成（UTF-8 エンコーディングと gzip 圧縮を使用した Parquet ファイル形式など）はすべて Snowflake でサポートされています。特に Parquet はカラム型ストレージファイル形式であり、Snowflake での高性能データ処理タスクに最適です。UTF-8 ファイルエンコーディングと gzip 圧縮はどちらも標準的で広く使用されている設定であり、クラウドストレージプラットフォームへのデータアンロードに関する Snowflake の機能と互換性があります。
参考文献：
SnowflakeのCOPY INTOコマンドに関するドキュメント
Snowflakeでサポートされているファイル形式に関するドキュメント
Snowflakeの圧縮およびエンコードオプションに関するドキュメント

質問: **64**
ある企業は、Amazon S3バケットからファイルを取り込むためにSnowpipeを使用したデータパイプラインを構築しました。Snowpipeは、データをステージングデータベーステーブルにロードするように設定されています。その後、タスクが実行され、ステージングデータベーステーブルからレポート用データベーステーブルにデータがロードされます。
同社は、レポート用データベーステーブルのデータ可用性には満足しているものの、レポート用テーブルのデータ削減が効果的に行われていない。現在、レポート用データベース内のすべてのテーブルを照会するために、4X-Largeサイズの仮想データウェアハウスが使用されている。
レポートテーブルのプルーニングを改善するために、どのような対策を講じることができますか？
A. Snowpipeの使用を廃止し、PUTコマンドを使用してファイルを内部ステージにロードします。
B. 仮想倉庫のサイズを5X-Largeに拡大します。
C. ORDER BY <cluster_key (s) > コマンドを使用してレポートテーブルをロードします。
D. Snowpipeが取り込むためのより大きなファイルを作成し、ステージングの頻度が1分を超えないようにしてください。
正解: **(**正解を表示します**)**
Snowflake における効果的なデータプルーニングは、マイクロパーティション内のデータの整理に依存します。レポートテーブルにデータをロードする際にクラスタリングキーを含む ORDER BY 句を使用することで、Snowflake はマイクロパーティション内のデータをより適切に整理できます。この整理により、Snowflake はクエリ実行時に無関係なマイクロパーティションをスキップできるため、クエリのパフォーマンスが向上し、スキャンされるデータ量が削減されます 12。
参照
* Snowflakeのマイクロパーティションとデータクラスタリングに関するドキュメント2

* 不適切な剪定を認識し、改善するためのコミュニティ記事1

質問: **65**
ファイルは、独自のシステムから10秒ごとに外部ステージに届きます。ファイルのサイズは500KBから3MBまでです。データは到着後すぐにダッシュボードからアクセスできる必要があります。
Snowflakeアーキテクトは、最小限のコーディングでこの要件を満たすにはどうすればよいでしょうか？ 2つ選択してください。）
A. Snowpipeを自動取り込み機能付きで使用する。
B. タスクで COPY コマンドを使用します。
C. 外部テーブルのマテリアライズドビューを使用します。
D. COPY INTOコマンドを使用します。
E. タスクとストリームを組み合わせて使用します。
正解: **A,E (**コメントを発表する**)**
要件は、データが外部ステージに到着した後、最小限のコーディング作業でできるだけ早くアクセスできるようにすることです。
オプションA：自動取り込み機能を備えたSnowpipeは、データがステージに到着すると同時に継続的にデータをロードするサービスです。自動取り込み機能により、Snowpipeはクラウドステージに到着した新しいファイルを自動的に検出し、最小限の遅延で、かつ介入なしで指定されたSnowflakeテーブルにデータをロードします。これは、ファイルが非常に高い頻度で到着するシナリオにおいて、メンテナンスの手間を最小限に抑えた理想的なソリューションです。
オプションE :タスクとストリームを組み合わせることで、Snowflakeでリアルタイムの変更データを取得できます。
ストリームはテーブルに加えられた変更（挿入、更新、削除）を記録し、非常に短い間隔で実行されるようにタスクをスケジュール設定することで、変更が発生した時点でダッシュボードテーブルに反映されるようにすることができます。

質問: **66**
アーキテクトが長時間実行されているステートメントのトラブルシューティングを行っており、ブロックされているトランザクションと、それらをブロックしているクエリを特定する必要があります。
どのビューを使用すべきですか？ 2つ選択してください）
A. クエリ履歴
B. オブジェクトの依存関係
C. データ転送履歴
D. LOCK_WAIT_HISTORY
E. アクセス履歴
正解: **(**正解を表示します**)**
LOCK_WAIT_HISTORY は、ロック待ちのトランザクションに関する詳細情報を提供します。これには、ブロックされているトランザクションと、それらをブロックしているトランザクションが含まれます (解答 D)。このビューは、競合や同時実行の問題を診断するために不可欠です。
QUERY_HISTORYは、実行時間、ユーザー、SQLテキストなど、ブロッキングクエリに関する実行詳細を提供することで、これを補完します（回答A）。これらのビューを組み合わせることで、アーキテクトはブロックされたトランザクションと、それに関連するワークロードを関連付けることができます。
その他のビューはトランザクションロックの動作とは無関係です。この質問は、SnowPro Architectの並行処理とトランザクション管理に関するトラブルシューティングスキルを問うものです。

質問: **67**

Snowflakeでは、クラスタキーを選択する際に何を推奨していますか？
A. 選択フィルターで最も頻繁に使用されるクラスタ列
B. カーディナリティの高いキーを選択する
C. クラスタキーを追加する余地がある場合は、結合述語で頻繁に使用される列を検討してください。
正解: **(**正解を表示します**)**

質問: **68**
あるメディア企業は、顧客レビューデータをSnowflakeテーブルに取り込み、いくつかの変換処理を適用するデータパイプラインを必要としています。また、同社はAmazon Comprehendを使用して感情分析を行い、匿名化された最終データセットを、異なる地域で異なるクラウドプロバイダーを利用する広告会社向けに公開する必要があります。
データパイプラインは、イベント通知を活用し、オブジェクトストレージに新しいレコードが到着するたびに、継続的かつ効率的に動作する必要があります。また、運用上の複雑さ、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保守、および開発作業は最小限に抑える必要があります。
どのデザインがこれらの要件を満たすでしょうか？
A. コピーイント機能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オーケストレーションします。データをAmazon S3にエクスポートしてAmazon Comprehendでモデル推論を実行し、データをSnowflakeテーブルに取り戻します。その後、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オーケストレーションします。Amazon Comprehendを使用してモデル推論を実行する外部関数を作成し、最終的なレコードをSnowflakeテーブルに書き込みます。次に、Snowflake Marketplaceにリストを作成し、他の企業がデータを利用できるようにします。
C. Snowflake Sparkコネクタを使用して、Amazon EMRとPySparkを使用してデータをSnowflakeに取り込みます。
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 を活用してモデル推論を行う Python プログラムを開発します。その後、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オーケストレーションします。データをAmazon S3にエクスポートしてAmazon Comprehendでモデル推論を実行し、データをSnowflakeテーブルに取り戻します。その後、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。
正解: **B (**コメントを発表する**)**
オプション B は、イベント通知を活用し、オブジェクト ストレージに新しいレコードが到着するたびに Snowpipe を使用してデータを継続的かつ効率的に取り込むため、要件を満たす最適な設計です。Snowpipe は、外部ソースから Snowflake テーブルへのデータのロードを自動化するサービスです 1。また、ストリームとタスクを使用して、取り込まれたデータに対する変換をオーケストレーションします。ストリームはテーブルの変更履歴を格納するオブジェクトであり、タスクはスケジュールに基づいて、または別のタスクによってトリガーされたときに SQL ステートメントを実行するオブジェクトです 2。オプション B では、外部関数を使用して Amazon Comprehend でモデル推論を実行し、最終的なレコードを Snowflake テーブルに書き込みます。外部関数は、Snowflake でネイティブにサポートされていない計算を実行するために、Amazon Comprehend などの外部 API を呼び出すユーザー定義関数です 3。
最後に、オプションBでは、Snowflake Marketplaceを使用して、匿名化された最終データセットを、異なる地域で異なるクラウドプロバイダーを使用する広告会社に公開します。Snowflake Marketplaceは、データプロバイダーが使用するクラウドプラットフォームや地域に関係なく、データセットをリスト化してデータコンシューマーと共有できるようにするプラットフォームです4。

オプションAは、データの取り込みにコピーイントネーションを使用するため、最適な設計とは言えません。コピーイントネーションは、Snowpipeほど効率的で継続的ではないからです。コピーイントネーションは、単一のトランザクションでファイルからテーブルにデータをロードするSQLコマンドです。また、Amazon Comprehendでモデル推論を行うためにデータをAmazon S3にエクスポートするため、手順が一つ増え、インフラストラクチャの運用上の複雑さとメンテナンスが増大します。
オプションCは、データの取り込みと変換にAmazon EMRとPySparkを使用するため、インフラストラクチャの運用上の複雑さとメンテナンスが増加するという点で、最適な設計とは言えません。Amazon EMRは、大規模なデータセットを処理および分析するためのマネージドHadoopフレームワークを提供するクラウドサービスです。
PySparkは、Hadoop上で動作する分散コンピューティングフレームワークであるSpark用のPython APIです。オプションCでは、Amazon Comprehendテキスト分析APIを活用してモデル推論を行うPythonプログラムも開発するため、開発作業量が増加します。
オプションDは、データ取り込み方法以外はオプションAと全く同じであるため、最適な設計とは言えません。Amazon Comprehendでモデル推論を行うためにデータをAmazon S3にエクスポートする必要があるため、手順が一つ増え、インフラストラクチャの運用上の複雑さとメンテナンスが増大します。
参考文献: 1: Snowpipe の概要 2: ストリームとタスクを使用したデータパイプラインの自動化 3: 外部関数の概要 4: Snowflake データマーケットプレイスの概要 : [COPY INTO を使用したデータのロード] : [Amazon EMR とは？] : [PySpark の概要]

質問**: 69**
CREDITCARDINFO テーブルの CREDICARDND 列にマスキング ポリシーが適用される以下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) returns 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以下のとおりです。
氏名 有効期限 クレジットカード番号
ジョン ･ドゥ 2022-07-23 4321 5678 9012 1234
Snowflakeシステムのロールに、追加の役割が付与されていない場合、どのような結果になりますか？
A. システム管理者はCREDICARDND列のデータを平文で確認できます。
B. テーブルの所有者は、CREDICARDND 列のデータを平文で見ることができます。
C. PI_ANALYTICS ロールを持つユーザーは、dear テキスト内の CREDICARDND 列データの最後の 4 文字を見ることができます。
D. PI_ANALYTICS ロールを持つユーザーは、CREDICARDND 列を *** 'MASKED* **' として表示します。
正解**: (**正解を表示します**)**
画像に示されているマスキングポリシーでは、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文字を平文で表示できると規定されています。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ロールは付与されず、クレジットカード番号の最後の4文字を表示することはできません。
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、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します。

ALTER TABLE CREDITCARDINFO ALTER COLUMN CREDITCARDNO SET MASKING POLICY creditcardno_mask; Snowflake でマスキング ポリシーを作成および使用する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。
マスキングポリシーの作成 :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 の動的データマスキングの設定方法と使用方法について説明します。動的データマスキングとは、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キングできる機能です。
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。

質問: **70**
アーキテクトがQUERY_HIST0RY関数を使用して、パフォーマンスの低いクエリのトラブルシューティングを行っています。アーキテクトは、COMPILATIONJHMEがEXECUTIONJTIMEよりも大きいことに気づきました。
その理由は？
A. このクエリは非常に大きなデータセットを処理しています。
B. クエリのロジックが過度に複雑です。
C. クエリは実行待ちのキューに追加されました。
D. クエリはリモートストレージから読み込んでいます。
正解: **B** (コメントを発表する)
コンパイル時間とは、オプティマイザがクエリを効率的に実行するための最適なクエリ プランを作成するのにかかる時間です。これには、パーティション ファイルの剪定も含まれ、クエリの実行を効率化します。2 コンパイル時間が実行時間よりも長い場合、オプティマイザがクエリの分析に、実際にクエリを実行するよりも多くの時間を費やしたことを意味します。これは、クエリに複数の結合、サブクエリ、集計、式など、過度に複雑なロジックが含まれていることを示している可能性があります。クエリの複雑さは、クエリ プランのサイズと品質にも影響し、クエリのパフォーマンスに影響を与える可能性があります。3 コンパイル時間を短縮するために、アーキテクトはクエリ ロジックを簡素化したり、ビューまたは共通テーブル式 (CTE) を使用してクエリをより小さな部分に分割したり、ヒントを使用してオプティマイザをガイドしたりすることができます。アーキテクトは、EXPLAIN コマンドを使用してクエリ プランを調べ、潜在的なボトルネックや非効率性を特定することもできます。4 参考資料:
1 :SnowPro Advanced :Architect｜学習ガイド5
2 :Snowflakeドキュメント｜クエリプロファイルの概要 6
3 :Snowflakeにおけるコンパイル時間が実行時間よりも長くなる理由を理解する 7
4 :Snowflakeドキュメント｜クエリパフォーマンスの最適化 8
SnowPro 上級者向け :アーキテクト | 学習ガイド
クエリプロファイルの概要
Snowflakeにおけるコンパイル時間が実行時間よりも長くなる理由を理解する クエリパフォーマンスの最適化

質問: **71**
アーキテクトは、JSONという名前の単一の列を含むleader_followerというテーブルを持っています。このテーブルには、次の構造を持つ行が1つあります。
{
「活動」: [
{ "activityNumber": 1, "winner": 5 },

{ "activityNumber": 2, "winner": 4 }
],
「フォロワー」:{
"名前": { "デフォルト": "マット" },
「番号」:4
},
"リーダー" :{
"name": { "default": "Adam" },
「番号」:5
}
}
以下の結果を生成するクエリはどれですか？
アクティビティ番号
勝者名
1
アダム
2
マット
A. SELECT lf.json:activities.activityNumber AS activity_number,
IFF(
lf.json:activities.activityNumber = lf.json:leader.number、
lf.json:リーダー名.デフォルト、
lf.json:follower.name.default
)::VARCHAR
リーダー_フォロワーlfから;
B. 選択
C. value:activityNumber AS activity_number、
IFF(
D. value:winner = lf.json:leader.number,
lf.json:リーダー名.デフォルト、
lf.json:follower.name.default
)::VARCHAR AS winner_name
リーダーフォロワーlfから、
LATERAL FLATTEN(input => json:activities) p;
E. 選択
F. value:activityNumber AS activity_number、
IFF(
G. value:winner = lf.json:leader.number,
lf.json:リーダー、
lf.json:フォロワー

)::VARCHAR AS winner_name
リーダーフォロワーlfから、
LATERAL FLATTEN(input => json:activities) p;
H. 選択
I. value:activityNumber AS activity_number、
IFF(
J. value:winner = lf.json:leader.number,
lf.json:リーダー名.デフォルト、
lf.json:follower.name.default
)::VARCHAR AS winner_name
リーダーフォロワーlfから、
LATERAL FLATTEN(input => json:activities, outer => TRUE) p;
正解: **(**正解を表示します**)**
この問題は、SnowPro Architect 試験範囲に明確に含まれる Snowflake のいくつかのコア半構造化データ概念、すなわち VARIANT データの操作、配列の処理、および LATERAL FLATTEN の使用についてテストします。JSON 構造の activities 要素は配列であるため、activityNumber や winner などの個々の属性にアクセスする前に、配列をフラット化する必要があります。オプション A は、配列をフラット化せずに配列内のフィールドを直接参照しようとしているため無効です。
オプションBは、json:activitiesに対してLATERAL FLATTENを正しく使用しており、アクティビティごとに1行が生成されます。エイリアスp.valueは各配列要素を表し、activityNumberとwinnerへのアクセスを可能にします。IFF式は、アクティビティのwinner値をleader.numberと比較します。一致する場合は、クエリはleader.name.defaultを返し、一致しない場合はfollower.name.defaultを返します。結果をVARCHARにキャストすることで、適切なスカラー出力が保証されます。
オプションCは、ネストされたname.default値ではなく、完全なJSONオブジェクト（リーダーまたはフォロワー）を返そうとしているため、誤りです。オプションDはOUTER => TRUEを使用していますが、アクティビティ配列が存在することが保証されているため、この場合は不要です。それでも動作はしますが、Snowflakeの試験問題では通常、最も正確で最小限の正解が求められます。

質問: **72**
企業のSnowflakeにおける日々のワークロードは、午後9時から午後11時の間に発生する膨大な数の同時実行クエリで構成されています。個々のクエリは、短時間で完了する小規模なステートメントです。
会社のアーキテクトは、このワークロードのパフォーマンスを向上させるために、どのような構成を実装できますか？（2つ選択してください。）
A. このワークロードで処理されるデータ量を削減してください。
B. 仮想倉庫のサイズを特大サイズに拡大します。
C. ワークロード実行中は、マルチクラスタ仮想倉庫を最大モードで有効にします。
D. 仮想倉庫レベルで、MAX_CONCURRENCY_LEVEL をデフォルト値の 8 よりも高い値に設定します。
E. 接続タイムアウトをデフォルト値よりも高い値に設定します。
正解: **(**正解を表示します**)**

質問: **73**
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ればよいですか？
A. ステージ統合

B. セキュリティ統合
C. ストレージ統合
D. API連携
正解: **(**正解を表示します**)**
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べき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 Okta、Duo、Google Authenticator などのサードパーティのセキュリティサービスとの間のインターフェースを提供する 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) を必須にしたり、エクスポート先を特定のネットワークまたはドメインに制限したりするなど、データのエクスポートに関するポリシーを適用できます。また、セキュリティ統合を使用して、Snowflake ユーザーに対して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 1。
The other options are incorrect because they are not integration objects that can be used to place restrictions on where data may be exported. Option A is incorrect because a stage integration is not a valid type of integration object in Snowflake. A stage is a Snowflake object that references a location where data files are stored, such as an internal stage, an external stage, or a named stage. A stage is not an integration object that provides an interface between Snowflake and third-party services2. Option C is incorrect because a storage integration is a Snowflake object that provides an interface between Snowflake and external cloud storage, such as Amazon S3, Azure Blob Storage, or Google Cloud Storage. A storage integration can be used to securely access data files from external cloud storage without exposing the credentials, but it cannot be used to place restrictions on where data may be exported3. Option D is incorrect because an API integration is a Snowflake object that provides an interface between Snowflake and third-party services that use REST APIs, such as Salesforce, Slack, or Twilio. An API integration can be used to securely call external REST APIs from Snowflake using the CALL_EXTERNAL_API function, but it cannot be used to place restrictions on where data may be exported4. References: CREATE SECURITY INTEGRATION | Snowflake Documentation, CREATE STAGE | Snowflake Documentation, CREATE STORAGE INTEGRATION | Snowflake Documentation, CREATE API INTEGRATION | Snowflake Documentation

質問: **74**
CSVファイル形式で`copy into <table>`コマンドを使用する場合、`match_by_column_name`パラメータはどのように動作しますか？
A. CSVファイルにはヘッダーが存在することが想定されており、そのヘッダーは大文字小文字を区別するテーブルの列名と照合されます。
B. このパラメータは無視されます。
C. コマンドはエラーを返します。
D. このコマンドを実行すると、ファイルに一致しない列があることを示す警告が表示されます。
正解: **B (**コメントを発表する**)**
Option B is the best design to meet the requirements because it uses Snowpipe to ingest the data continuously and efficiently as new records arrive in the object storage, leveraging event notifications. Snowpipe is a service that automates the loading of data from external sources into Snowflake tables1. It also uses streams and tasks to orchestrate transformations on the ingested data. Streams are objects that store the change history of a table, and tasks are objects that execute SQL statements on a schedule or when triggered by another task2.
Option B also uses an external function to do model inference with Amazon Comprehend and write the final records to a Snowflake table. An external function is a user-defined function that calls an external API, such as Amazon Comprehend, to perform computations that are not natively supported by Snowflake3. Finally, option B uses the Snowflake Marketplace to make the de-identified final data set available publicly for advertising companies who use different cloud providers in different regions. The Snowflake Marketplace is a platform that enables data providers to list and share their data sets with data consumers, regardless of the cloud platform or region they use4.

Option A is not the best design because it uses copy into to ingest the data, which is not as efficient and continuous as Snowpipe. Copy into is a SQL command that loads data from files into a table in a single transaction. It also exports the data into Amazon S3 to do model inference with Amazon Comprehend, which adds an extra step and increases the operational complexity and maintenance of the infrastructure.
Option C is not the best design because it uses Amazon EMR and PySpark to ingest and transform the data, which also increases the operational complexity and maintenance of the infrastructure. Amazon EMR is a cloud service that provides a managed Hadoop framework to process and analyze large-scale data sets.
PySpark is a Python API for Spark, a distributed computing framework that can run on Hadoop. Option C also develops a python program to do model inference by leveraging the Amazon Comprehend text analysis API, which increases the development effort.
Option D is not the best design because it is identical to option A, except for the ingestion method. It still exports the data into Amazon S3 to do model inference with Amazon Comprehend, which adds an extra step and increases the operational complexity and maintenance of the infrastructure.
References: 1: Snowpipe Overview 2: Using Streams and Tasks to Automate Data Pipelines 3: External Functions Overview 4: Snowflake Data Marketplace Overview : [Loading Data Using COPY INTO] : [What is Amazon EMR?] : [PySpark Overview]
* The copy into <table> command is used to load data from staged files into an existing table in Snowflake. The command supports various file formats, such as CSV, JSON, AVRO, ORC, PARQUET, and XML1.
* match_by_column_name パラメータは、半構造化データを、ソースデータで表現されている対応する列と一致するターゲットテーブルの個別の列にロードできるようにするコピーオプションです。このパラメータには、次のいずれかの値を指定できます。
* CASE_SENSITIVE: ソースデータの列名は、ターゲットテーブルの列名と大文字 •小文字を含めて完全に一致する必要があります。これがデフォルト値です。
* CASE_INSENSITIVE: ソースデータの列名はターゲットテーブルの列名と一致する必要がありますが、大文字と小文字は区別されません。
* なし: ソースデータの列名は無視され、ターゲットテーブルの列の順序に基づいてデータがロードされます。
* match_by_column_name パラメータは、JSON、AVRO、ORC、PARQUET、XML などの半構造化データにのみ適用されます。構造化データとみなされる CSV データには適用されません。
* CSVファイル形式で<table>にコピーするコマンドを使用する場合、match_by_column_nameパラメータは次のように動作します2:
* CSVファイルにはヘッダーが存在する必要があり、そのヘッダーは大文字小文字を区別するテーブルの列名と照合されます。つまり、CSV ファイルの最初の行には列名が含まれている必要があり、その列名は対象テーブルの列名と大文字小文字を含めて完全に一致している必要があります。ヘッダーが存在しない場合、または一致しない場合は、コマンドはエラーを返します。
* このパラメータは、NONE に設定されている場合でも無視されません。コマンドは、CSV ファイル内の列名と対象テーブルの列名を照合しようとし、一致しない場合はエラーを返します。
* このコマンドは、ファイルに一致しない列があることを示す警告を返しません。列名が一致する場合はデータが正常にロードされ、一致しない場合はエラーが返されます。
参考文献：
* 1: COPY INTO <table> | Snowflakeドキュメント
* 2: MATCH_BY_COLUMN_NAME | Snowflake ドキュメント

質問**: 75**
外部テーブルスキーマに含めることができる列はどれですか？ 3つ選択してください。）
A. METADATASROW_ID
B. 価値
C. メタデータ更新

D. メタデータ A$ ファイル名
E. メタデータ外部テーブルパーティション
F. メタデータ ファイル行番号
正解: **(**正解を表示します**)**

質問: **76**
外部テーブルに行アクセス ポリシーを適用する際の特徴として、次のうちどれが挙げられますか？ 3つ選択してください。）
A. 外部テーブルの上に作成されたビューには、行アクセス ポリシーを適用できません。
B. 行アクセス ポリシーを使用して外部テーブルを作成でき、そのポリシーを VALUE 列に適用できます。
C. 既存の外部テーブルのVALUE列に行アクセス ポリシーを適用できます。
D. データベースをクローンする際に、行アクセス ポリシーと外部テーブルの両方がクローンされます。
E. 行アクセス ポリシーは、外部テーブルの仮想列に直接追加することはできません。
F. 外部テーブルは、行アクセス ポリシーのマッピング テーブルとしてサポートされています。
正解: **(**正解を表示します**)**

質問: **77**
データベースのクローン作成中に、パイプ定義のCOPYステートメントで完全修飾テーブル（db_name.schema_name.table_nameまたはschema_name.table_nameの形式）を指定してSNOWPIPEをクローンした場合、どうなりますか？
A. すべてのファイルがステージに再度ロードされます
B. Snowpipeがトリガーされるとデータ読み込みが失敗します
C. Snowpipeは重複データをソーステーブルにロードします
正解: **(**正解を表示します**)**

質問: **78**
アーキテクトは、外部ステージからSnowflakeへ2つのファイルを毎日自動的にインポート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、もう1つのファイルにはCSV形式のデータが含まれています。
最終的な結果セットを作成するために、データはどのように結合および集計されるべきでしょうか？
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結果セットを生成します。
B. Snowflakeスクリプトを使用してタスクを作成し、ファイルをインポートしてから、ユーザー定義関数（UDF）を呼び出して最終結果セットを生成します。
C. 外部ステージから直接データを読み込み、結合、集計し、結果をテーブルに格納する JavaScript ストアドプロシージャを作成します。
D. 外部ステージから直接データを読み取り、結合、集計するためのマテリアライズドビューを作成し、そのビューを使用して最終結果セットを生成します。
正解: **B (**コメントを発表する**)**

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）のスケジュール設定と実行を可能にするオブジェクトです。タスクは、データのロード、変換、およびメンテナンス操作を自動化するために使用できます。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記述できる機能です。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、タスクをオーケストレーションできます。したがって、外部ステージからSnowflakeに2つのファイルを毎日インポートし、データを結合および集計して最終結果セットを生成する作業を自動化する最良の方法は、COPY INTOコマンドを使用してファイルをインポートし、次にUDFを呼び出して結合および集計ロジックを実行するSnowflakeスクリプトを使用したタスクを作成することです。UDFは、最終結果セットとしてテーブルまたはバリアント値を返すことができます。参照:
* タスク
* Snowflake スクリプト
* ユーザー定義関数

質問: **79**
以下の手順に従ってタスクとストリームを作成します。
system$stream_has_data('rawstream1') 条件が false を返すと、タスクはどうなりますか？
-- 生の JSON データを保存するランディング テーブルを作成します。
-- Snowpipe はこのテーブルにデータをロードできます。create or replace table raw (var variant);
-- ランディングテーブルへの挿入をキャプチャするためのストリームを作成します。
-- タスクはこのストリームから一連の列を消費します。create or replace stream rawstream1 on table raw;
-- ランディングテーブルへの挿入をキャプチャするための2つ目のストリームを作成します。
-- 2 番目のタスクは、このストリームから別の列セットを消費します。create or replace stream rawstream2 on table raw;
-- 生データで識別されたオフィス訪問者の名前を格納するテーブルを作成します。create or replace table names (id int, first_name string, last_name string);
-- 生データで特定されたオフィス訪問者の訪問日を格納するテーブルを作成します。
CREATE OR REPLACE TABLE visits (id int, dt date);
-- rawstream1 ストリームから新しい名前レコードを names テーブルに挿入するタスクを作成します
-- ストリームにレコードが含まれている場合、1分ごとに。
-- 'etl_wh' ウェアハウスを、あなたのロールが USAGE 権限を持つウェアハウスに置き換えてください。create or replace task raw_to_names
倉庫 = etl_wh スケジュール = '1 分' の場合
system$stream_has_data('rawstream1') as
名前nにマージする
using (select var:id id, var:fname fname, var:lname lname from rawstream1) r1 on n.id = to_number(r1.id)
一致した場合、n.first_name = r1.fname、n.last_name = r1.lname を更新します。
一致しない場合は、(id、first_name、last_name) の値 (r1.id、r1.fname、r1.lname) を挿入します。
;
-- rawstream1 ストリームからの訪問記録を visits テーブルにマージする別のタスクを作成します
-- ストリームにレコードが含まれている場合、1分ごとに。
-- 新しいIDを持つレコードが訪問テーブルに挿入されます。
-- visits テーブルに存在する ID を持つレコードは、テーブルの DT 列を更新します。
-- 'etl_wh' ウェアハウスを、あなたのロールが USAGE 権限を持つウェアハウスに置き換えてください。create or replace task raw_to_visits
倉庫 = etl_wh スケジュール = '1 分' の場合

system$stream_has_data('rawstream2')
訪問に統合 v
using (select var:id id, var:visit_dt visit_dt from rawstream2) r2 on v.id = to_number(r2.id) when matched then update set v.dt = r2.visit_dt
一致しない場合は、(id, dt) の値 (r2.id, r2.visit_dt) を挿入します。
;
両方のタスクを再開します。
alter task raw_to_names resume;
alter task raw_to_visits resume;
-- ランディングテーブルにレコードセットを挿入します。insert into raw
値からparse_json(column1)を選択します
('{"id": "123","fname": "Jane","lname": "Smith","visit_dt": "2019-09-17"}')
('{"id": "456","fname": "Peter","lname": "Williams","visit_dt": "2019-09-17"}');
-- テーブル streams 内の変更データキャプチャレコードをクエリします。select * from rawstream1;
rawstream2から*を選択します。
A. タスクは警告メッセージを返します
B. タスクは実行されますが、行はマージされません
C. タスクはスキップされます
正解: **(**正解を表示します**)**

質問: **80**
あるアーキテクトが、複数のソースから小さなCSVファイルを受信するデータパイプラインを設計しました。すべてのファイルは1つの場所に格納されます。特定のファイルは、copyコマンドを使用してSnowflakeテーブルにロードするためにフィルタリングされます。しかし、ロードのパフォーマンスは低いです。
データ読み込みパフォーマンスを向上させるには、どのような変更を加えることができますか？
A. 仮想倉庫のサイズを拡大します。
B. マルチクラスタウェアハウスを作成し、小さなファイルをマージして大きなファイルを作成します。
C. ファイルスキャンを回避するために、専用のストレージランディングバケットを作成します。
D. ファイル形式をCSVからJSONに変更します。
正解: **(**正解を表示します**)**
Snowflakeのドキュメントによると、データファイルの準備とステージングに関するいくつかのベストプラクティスとガイドラインに従うことで、データロードのパフォーマンスを向上させることができます。推奨事項の1つは、圧縮後のデータファイルのサイズが約100〜250MB (またはそれ以上)になるようにすることです。これにより、ロード時の並列操作の数が最適化されます。このサイズ範囲を達成するには、小さいファイルは集約し、大きいファイルは分割する必要があります。もう1つの推奨事項は、ロードにマルチクラスタウェアハウスを使用することです。これにより、ロード需要に応じてコンピューティングリソースをスケールアップまたはスケールアウトできます。シングルクラスタウェアハウスでは、ロードの同時実行性とスループットを効率的に処理できない場合があります。したがって、マルチクラスタウェアハウスを作成し、小さいファイルをマージして大きいファイルを作成することで、データロードのパフォーマンスを向上させることができます。参照:
データ読み込みに関する考慮事項
データファイルの準備
データロードの計画

質問: **81**
Data Vaultを使用してモデリングを行う際に、Snowflakeのどの機能を活用すべきでしょうか？
A. Snowflakeは、データモデルのData Vaultテーブルへの複数テーブル挿入をサポートしています。
B. 優れたデータアクセス性能を得るためには、データを事前にパーティション分割する必要があります。
C. 仮想倉庫の規模拡大により、新規ソースロードの並列処理がサポートされます
D. Snowflakeのハッシュキー機能により、ハッシュキー結合は整数結合よりも高速に実行できます。
正解: **(正解を表示します)**
これら2つの機能は、Snowflake上のData Vaultを使用したモデリングに関連しています。Data Vaultは、データをハブ、リンク、サテライトに整理するデータモデリング手法です。Data Vaultは、データ統合と分析において高い拡張性、柔軟性、パフォーマンスを実現するように設計されています。Snowflakeは、Data Vaultを含むさまざまなデータモデリング手法をサポートするクラウドデータプラットフォームです。Snowflakeには、Data Vaultモデリングを強化できる以下のような機能があります。
Snowflake は、データ モデルの Data Vault テーブルへの複数テーブル挿入をサポートしています。複数テーブル挿入 (MTI) は、単一の DML ステートメントで単一のクエリから複数のテーブルにデータを挿入できる機能です。MTI は、特にリアルタイムまたはほぼリアルタイムのデータ統合において、Data Vault テーブルへのデータロードのパフォーマンスと効率を向上させることができます。また、MTI は、ロード コードの複雑さとメンテナンス、およびデータの重複とレイテンシを削減することもできます 12。
仮想ウェアハウスをスケールアップすると、新しいソースロードの並列処理がサポートされます。仮想ウェアハウスは、データ処理のためにオンデマンドでコンピューティング リソースをプロビジョニングできる機能です。仮想ウェアハウスは、ウェアハウスのサイズを変更することでスケールアップまたはスケールダウンできます。ウェアハウスのサイズは、ウェアハウス内のサーバー数を決定します。仮想ウェアハウスをスケールアップすると、特に大規模または複雑なデータセットの場合、新しいソースロードを Data Vault テーブルに処理する際のパフォーマンスと並行性が向上します。仮想ウェアハウスをスケールアップすると、Snowflake アーキテクチャの並列性と分散性を活用することもでき、データのロードとクエリを最適化できます 34。
参照：
Snowflakeドキュメント：複数テーブルへの挿入
Snowflakeブログ :Snowflake上のデータボールトアーキテクチャを最適化するためのヒント Snowflakeドキュメント：仮想ウェアハウス
Snowflakeブログ :Snowflakeでリアルタイムデータボールトを構築する

質問: **82**
水使用量を測定するIoTデバイス用のテーブルが作成されます。このテーブルはすぐに大きくなり、20億行を超えるデータ量になります。

```
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)
)
```

このテーブルに対する一般的なクエリパターンは以下のとおりです。
1. DeviceId、IOT_timestamp、および CustomerId は、select ステートメントのフィルタ述語で頻繁に使用されます。
2. City 列と DeviceManufacturer 列はよく取得されます

3. UniqueId にはよくカウントがあります
クラスタリングキーにはどのフィールドを使用すべきですか？
A. IOT_タイムスタンプ
B. ユニークルド
C. 都市およびデバイスメーカー
D. DeviceId と CustomerId
正解: **(**正解を表示します**)**

質問: **83**
ORGADMINロールで実行できる組織関連のタスクはどれですか？ 3つ選択してください。）
A. 組織名の変更
B. アカウントの作成
C. 組織アカウントの一覧を表示する
D. アカウント名の変更
E. アカウントの削除
F. データベースのレプリケーションを有効にする
正解: **(**正解を表示します**)**
SnowPro Advanced: Architectのドキュメントと学習リソースによると、ORGADMINロールで実行できる組織関連のタスクは次のとおりです。
* 組織内でのアカウントの作成。ORGADMIN ロールを持つユーザーは、CREATE ACCOUNT コマンドを使用して、現在のアカウント 1 と同じ組織に属する新しいアカウントを作成できます。
* 組織アカウントの一覧を表示します。ORGADMIN ロールを持つユーザーは、SHOW ORGANIZATION ACCOUNTS コマンドを使用して、組織内のすべてのアカウントの名前とプロパティを表示できます。2. または、Web インターフェイスの [アカウントの管理] ページを使用して、組織名とアカウント名を表示することもできます。3.
* データベースのレプリケーションを有効にする。ORGADMIN ロールを持つユーザーは、SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER 関数を使用して、組織内のアカウントのデータベース レプリケーションを有効にできます。これにより、ユーザーは異なるリージョンやクラウド プラットフォームのアカウント間でデータベースをレプリケートして、データの可用性と耐久性を確保できます。4
他のオプションは、ORGADMIN ロールで実行できる組織関連のタスクではないため、誤りです。オプション A は、組織名の変更は ORGADMIN ロールで実行できるタスクではないため、誤りです。組織名を変更するには、ユーザーは Snowflake サポートに連絡する必要があります 3。オプション D は、アカウント名の変更は ORGADMIN ロールで実行できるタスクではないため、誤りです。アカウント名を変更するには、ユーザーは Snowflake サポートに連絡する必要があります 5。オプション E は、アカウントの削除は ORGADMIN ロールで実行できるタスクではないため、誤りです。アカウントを削除するには、ユーザーは Snowflake サポートに連絡する必要があります。参照: CREATE ACCOUNT | Snowflake ドキュメント、SHOW ORGANIZATION ACCOUNTS | Snowflake ドキュメント、Getting Started with Organizations | Snowflake ドキュメント、SYSTEM$GLOBAL_ACCOUNT_SET_PARAMETER | Snowflake ドキュメント、ALTER ACCOUNT | Snowflake ドキュメント、[DROP ACCOUNT | Snowflake ドキュメント]

質問: **84**
大手製造会社のデータエンジニアリングチームは、以下のような多様なユースケースとデータ利用者の要件をサポートするために、多くのソースから得られるデータをエンジニアリングする必要があります。
1) レポート作成と可視化を必要とする財務およびベンダー管理チームのメンバー
2) 機械学習モデル開発のために生データへのアクセスが必要なデータサイエンスチームのメンバー

3) データ収益化のために、設計され保護されたデータを必要とする営業チームメンバー。これらの要件を満たすSnowflakeのデータモデリング手法はどれですか？ 2つ選択してください。）
A. 会社のデータレイクにデータを統合し、外部テーブルを使用します。
B. データパイプラインに入力される生データを取り込んで永続化するための生データベースを作成します。
C. 使用パターンに合わせてデータを整理した、プロファイル固有のデータベースセットを作成します。
D. すべての消費者の要件をサポートするために、単一のデータベースに単一のスター型スキーマを作成します。
E. データ ボールトを唯一のデータ パイプライン エンドポイントとして作成し、すべてのコンシューマーがボールトに直接アクセスできるようにします。
正解: **(**正解を表示します**)**
企業内のさまざまなチームやユースケースの多様なニーズに対応するためには、柔軟で多面的なデータモデリングのアプローチが必要となる。
オプションB：生データを格納するための生データベースを作成することで、データサイエンスチームが機械学習モデル開発に必要な未処理データにアクセスできるようになります。これは、最新のデータアーキテクチャにおいて、生データを様々なユースケースに合わせて変換または処理する前に取り込むためのステージングエリアまたは生データゾーンを設けるというベストプラクティスに合致しています。
オプションC :プロファイル固有のデータベースを持つということは、社内の各ユーザープロファイルまたはチームの特定の要件を満たすように設計された、ターゲットを絞ったデータベースを作成することを意味します。財務チームとベンダー管理チームの場合、データはレポート作成と視覚化のために構造化され、最適化されます。営業チームの場合、データベースにはデータ収益化に適した、設計され保護されたデータを含めることができます。
この戦略は、データを利用パターンに合わせるだけでなく、データアクセスとセキュリティポリシーを効果的に管理するのにも役立ちます。

質問: **85**
Snowflakeで使用されているロールベースアクセス制御（RBAC）の特徴は何ですか？
A. 権限はデータベースレベルで付与でき、すべての下位オブジェクトに継承されます。
B. ユーザーは、将来の権限付与をサポートし、スキーマ所有者のみが他のロールに権限を付与できるようにするための管理アクセススキーマを作成できます。
C. ユーザーは、現在および将来の権限付与をサポートする管理アクセス スキーマを作成し、オブジェクトの所有者のみが他のロールに権限を付与できるようにすることができます。
D. ユーザーは、securityadmin とともに「スーパーユーザー」アクセスを使用することで、認証チェックをバイパスして、すべてのデータベース、スキーマ、および基となるオブジェクトにアクセスできます。
正解: **(**正解を表示します**)**

質問: **86**
アーキテクトは、アプリケーションサーバーに追加のソフトウェアをインストールすることなく、Snowflakeとの間でデータの読み書きを行う必要があるアプリケーションを統合しようとしています。
この要件を満たすにはどうすればよいでしょうか？
A. SnowSQLを使用してください。
B. Snowpipe REST API を使用します。
C. Snowflake SQL REST API を使用します。
D. Snowflake ODBC ドライバーを使用します。

正解: **(**正解を表示します**)**
Snowflake SQL REST API は、Snowflake データベース内のデータにアクセスして更新するために使用できる REST API です。この API を使用して、標準クエリとほとんどの DDL および DML ステートメントを実行できます。この API を使用すると、アプリケーション サーバーに追加のソフトウェアをインストールすることなく、Snowflake にデータを読み書きできるカスタム アプリケーションや統合を開発できます。オプション A は正しくありません。SnowSQL はコマンドライン クライアントであり、アプリケーション サーバーにインストールと構成が必要です。オプション B は正しくありません。Snowpipe REST API は、クラウド ストレージから Snowflake テーブルにデータをロードするために使用され、Snowflake にデータを読み書きするためには使用されません。オプション D は正しくありません。Snowflake ODBC ドライバーは、アプリケーションが ODBC プロトコルを使用して Snowflake に接続できるようにするソフトウェア コンポーネントであり、アプリケーション サーバーにインストールと構成が必要です。参照: 回答は、Snowflake の Web サイトにある Snowflake SQL REST API の公式ドキュメントで確認できます。関連リンクを以下に示します。
Snowflake SQL REST API | Snowflakeドキュメント
SQL API の概要 | Snowflake ドキュメント
SQLステートメント実行リクエストの送信 | Snowflakeドキュメント

質問: **87**
ファイルは、独自のシステムから10秒ごとに外部ステージに届きます。ファイルのサイズは500KBから3MBまでです。データは到着後すぐにダッシュボードからアクセスできる必要があります。
Snowflakeアーキテクトは、最小限のコーディングでこの要件を満たすにはどうすればよいでしょうか？ 2つ選択してください。）
A. Snowpipeを自動取り込み機能付きで使用する。
B. タスクで COPY コマンドを使用します。
C. 外部テーブルのマテリアライズドビューを使用します。
D. COPY INTOコマンドを使用します。
E. タスクとストリームを組み合わせて使用します。
正解: **(**正解を表示します**)**
これら2つの方法は、外部ステージからデータを読み込み、最小限のコーディングでダッシュボードからアクセスできるようにするという要件を満たすための最良の方法です。
Snowpipeの自動取り込み機能は、外部ステージからSnowflakeテーブルへの継続的かつ自動的なデータロードを可能にする機能です。Snowpipeはクラウドストレージサービスからのイベント通知を利用して、ステージ内の新規ファイルまたは変更されたファイルを検出し、COPY INTOコマンドをトリガーしてデータをテーブルにロードします。Snowpipeは効率的でスケーラブルなサーバーレス設計のため、ユーザーによるインフラストラクチャの構築やメンテナンスは不要です。また、Snowpipeはサポートされているフォーマット1であれば、あらゆるサイズのファイルからのデータロードをサポートします。
外部テーブルのマテリアライズドビューは、外部テーブルから事前に計算された結果セットを作成し、Snowflakeに保存できる機能です。マテリアライズドビューを使用すると、特に複雑なクエリやダッシュボードの場合、外部テーブルからのデータクエリのパフォーマンスと効率を向上させることができます。また、マテリアライズドビューは、外部テーブルデータに対する集計、結合、フィルタもサポートします。外部テーブルのマテリアライズドビューは、AUTO_REFRESHパラメータがtrueに設定されている場合、外部ステージの基となるデータが変更されると自動的に更新されます。
参照：
Snowpipeの概要 | Snowflakeドキュメント
外部テーブルのマテリアライズドビュー | Snowflakeドキュメント

質問: **88**
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・分析するための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門と部署が存在します。要件は、拡張性、セキュリティ、およびコスト効率です。
どのようなデザインを採用すべきでしょうか？
A. データ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. データ分離とセキュリティを確保するため、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、構造化された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および分析します。
D. コアビジネスデータには中央集権型のSnowflakeデータベースを使用し、部門別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。
正解: **(**正解を表示します**)**
さまざまな事業部門や部署向けに大量の構造化データおよび半構造化データを保存・分析するための最適な設計は、コア業務データにはSnowflakeの中央データベースを使用し、部門別またはプロジェクト別のデータには個別のデータベースを使用することです。この設計では、Snowflakeの以下のような機能を活用することで、拡張性、セキュリティ、コスト効率が向上します。
データベースのクローン作成：データベースをクローン作成すると、元のデータベースと同じデータファイルを共有するものの、独立して変更可能なゼロコピーのクローンが作成されます。これにより、ストレージコストが削減され、さまざまな用途に対応した高速かつ一貫性のある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ことで、データベース内のデータの一部に対して、他のSnowflakeアカウントや顧客に安全かつ管理されたアクセス権限を付与できます。これにより、異なる事業部門間や外部パートナー間でのデータ連携や収益化が可能になります。
ウェアハウスのスケーリング：ウェアハウスのスケーリングにより、ウェアハウスのサイズと同時実行数を調整して、さまざまなワークロードのパフォーマンスとコスト要件に合わせることができます。これにより、最適なリソース利用と、さまざまなデータ分析ニーズに対する柔軟性が実現します。参照：Snowflakeドキュメント：データベースのクローニング、Snowflakeドキュメント：データベースの共有、[Snowflakeドキュメント：ウェアハウスのスケーリング]

質問: **89**
Snowflakeにおけるマテリアライズドビューの使用に関する特徴を説明しているのは、次のうちどれですか？（2つ選択してください。）
A. ORDER BY句を含めることができます。
B. ネストされたサブクエリを含めることはできません。
C. CURRENT_TIME() などのコンテキスト関数を含めることができます。
D. MINとMAXの集計をサポートできます。
E. 内部結合はサポートできますが、外部結合はサポートできません。
正解: **B,D (**コメントを発表する**)**
説明
Snowflakeのドキュメントによると、マテリアライズドビューには、それを定義するクエリ仕様に関していくつかの制限があります。その1つは、FROM句内のサブクエリやSELECTリスト内のスカラーサブクエリなど、ネストされたサブクエリを含めることができないことです。もう1つの制限は、ORDER BY句、コンテキスト関数（CURRENT_TIME()など）、外部結合を含めることができないことです。ただし、マテリアライズドビューは、MINおよびMAX集計関数、ならびにSUM、COUNT、AVGなどの他の集計関数をサポートできます。

参考文献：
* マテリアライズド ビューの作成に関する制限事項 | 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

質問: **90**
開発環境におけるデータライフサイクル管理ツールとしてデータベースクローニングを使用する際に考慮すべき事項は何ですか？ 2つ選択してください)。
A. クローンは、ソースオブジェクト内のすべての子オブジェクトに付与されたすべての権限を継承しますが、データベースは除外します。
B. クローンは、データベースを含む、ソースオブジェクト内のすべての子オブジェクトに付与されたすべての権限を継承します。
C. ソース内の外部ステージを参照するパイプはクローンされません。
D. ソース内のパイプはクローンされません。
E. ソース内の内部ステージを参照するパイプはクローンされません。
正解: **(**正解を表示します**)**

質問: **91**
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した場合、開発およびテストのシナリオでデータのコピーが必要となり、ゼロコピー クローニングが適さないのはどのようなシナリオでしょうか？ 2 つ選択してください)。
A. 開発者は、ライブデータの変換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. リリースプロセスでは、本番環境と同等の規模と複雑さを持つデータを用いて、変更内容の事前テストを実施する必要があります。セキュリティ上の理由から、事前テストは本番環境のアカウントでも実行されます。
C. 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、開発者は特定の列がマスクされた本番環境のようなデータを見る必要があります。
D. 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コピーを、初期開発および単体テスト用に独自に作成します。
E. データは本番環境の 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/テスト用 Snowflake アカウントに開発者に提供する必要があります。
正解: **B,E (**コメントを発表する**)**

質問: **92**
開発環境におけるデータライフサイクル管理ツールとしてデータベースクローニングを使用する際に考慮すべき事項は何ですか？ 2つ選択してください)。
A. クローンは、ソースオブジェクト内のすべての子オブジェクトに付与されたすべての権限を継承しますが、データベースは除外します。
B. ソース内のパイプはクローンされません。
C. クローンは、データベースを含む、ソースオブジェクト内のすべての子オブジェクトに付与されたすべての権限を継承します。
D. ソース内の内部ステージを参照するパイプはクローンされません。

E. ソース内の外部ステージを参照するパイプはクローンされません。
正解: **(**正解を表示します**)**

質問: **93**
What step will im the performance of queries executed against an external table?
A. Partition the external table.
B. Shorten the names of the source files.
C. Convert the source files' character encoding to UTF-8.
D. Use an internal stage instead of an external stage to store the source files.
正解: **(**正解を表示します**)**
外部テーブルのパーティショニングは、スキャンするデータ量を減らすことで、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ニングでは、テーブルを論理的にデータのサブセット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を作成します。パーティション列は、ファイルメタデータ（ファイル名、パス、サイズ、変更時刻など）またはファイルコンテンツ（列の値やJSON属性など）から取得できます。外部テーブルをパーティショニングすると、クエリオプティマイザはクエリ述語に一致しないファイルを削除できるため、不要なデータスキャンと処理を回避できます。2 その他のオプションは、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ための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縮します。ファイル名はクエリ処理には使用されないため、このオプションはクエリのパフォーマンスに影響を与えません。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。
* 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UTF-8、UTF-16、UTF-32、ISO-8859-1、Windows-1252など、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的にUTF-8に変換します。4
* ソースファイルの保存には、外部ステージではなく内部ステージを使用します。このオプションは適用できません。外部テーブルは、Amazon S3、Google Cloud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です。
* ストレージ、または Azure Blob Storage。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。5
参考資料:
* 1 :SnowPro Advanced :Architect｜学習ガイド
* 2: Snowflake ドキュメント | 外部テーブルのパーティショニング
* 3: Snowflake ドキュメント | 外部テーブルの作成
* 4: Snowflake 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ているファイル形式と圧縮
* 5 :Snowflakeドキュメント｜ステージの概要
* : SnowPro Advanced: Architect | 学習ガイド
* : 外部テーブルのパーティショニング
* : 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
* : 各段階の概要

質問: **94**
COPY INTOを使用してステージからデータをロードする場合、ON_ERROR句にはどのようなオプションを指定できますか？
A. 続ける
B. SKIP_FILE

C. ABORT_STATEMENT
D. 失敗
正解**: (**正解を表示します**)**
* ON_ERROR句は、COPY INTOコマンドのオプションパラメータであり、ファイル内でエラーが発生した場合のコマンドの動作を指定します。ON_ERROR句には、次のいずれかの値を指定できます。1:
* CONTINUE: この値を指定すると、コマンドはファイルの読み込みを続行し、データファイルごとに最大 1 つのエラーが発生した場合にエラー メッセージを返します。ROWS_PARSED 列と ROWS_LOADED 列の値の差は、検出されたエラーを含む行数を表します。データ ファイル内のすべてのエラーを表示するには、VALIDATION_MODE パラメーターを使用するか、VALIDATE 関数 1 をクエリしてください。
* SKIP_FILE: この値を指定すると、コマンドはファイル内のいずれかのレコードでデータエラーが発生した場合に、そのファイルをスキップします。コマンドはステージ内の次のファイルに進み、読み込みを続行します。スキップされたファイルは読み込まれず、ファイル1に関するエラーメッセージは返されません。
* ABORT_STATEMENT: この値は、最初のエラーが発生した時点でデータの読み込みを停止するようにコマンドに指示します。コマンドはファイルのエラーメッセージを返し、読み込み操作を中止します。これは、ON_ERROR 句1 のデフォルト値です。
したがって、選択肢A、B、Cが正解です。
参照: : COPY INTO <table>

質問**: 95**
Snowflakeで使用されているロールベースアクセス制御（RBAC）の特徴は何ですか？
A. 権限はデータベースレベルで付与でき、すべての下位オブジェクトに継承されます。
B. ユーザーは、securityadmin とともに「スーパーユーザー」アクセスを使用することで、認証チェックをバイパスして、すべてのデータベース、スキーマ、および基となるオブジェクトにアクセスできます。
C. ユーザーは、将来の権限付与をサポートし、スキーマ所有者のみが他のロールに権限を付与できるようにするための管理アクセススキーマを作成できます。
D. ユーザーは、現在および将来の権限付与をサポートする管理アクセス スキーマを作成し、オブジェクトの所有者のみが他のロールに権限を付与できるようにすることができます。
正解**: C (**コメントを発表する**)**
ロールベースアクセス制御（RBAC）は、Snowflakeのアクセス制御フレームワークであり、オブジェクトの所有者がロールに権限を付与し、ロールをユーザーに割り当てることで、オブジェクトに対する操作を制限または許可することができます。Snowflakeで使用されるRBACの特徴は次のとおりです。
権限はデータベースレベルで付与でき、すべての基となるオブジェクトに継承されます。つまり、CREATE SCHEMAやUSAGEなどのデータベースに対する特定の権限を持つロールは、明示的に取り消されない限り、そのデータベース内の任意のスキーマ、テーブル、ビュー、またはその他のオブジェクトに対しても同じ操作を実行できます。これにより、アクセス制御管理が簡素化され、必要な権限付与の数を減らすことができます。
ユーザーは、将来の権限付与をサポートし、スキーマ所有者のみが他のロールに権限を付与できるようにするために、管理アクセス スキーマを作成できます。つまり、ユーザーは「管理アクセス」オプションを使用してスキーマを作成でき、これにより、スキーマ内のオブジェクト所有権と権限付与のデフォルトの動作が変更されます。管理アクセス スキーマでは、オブジェクト所有者はオブジェクトに対する権限を他のロールに付与できなくなり、スキーマ所有者または「権限付与の管理」権限を持つロールのみが権限を付与できます。これにより、スキーマとそのオブジェクトのセキュリティとガバナンスが強化されます。
その他のオプションは、Snowflakeで使用されているRBACの特徴ではありません。

ユーザーは、securityadmin と 「スーパーユーザー」アクセスを組み合わせて使用することで、認証チェックを回避し、すべてのデータベース、スキーマ、および基となるオブジェクトにアクセスできます。これは誤りです。Snowflake には 「スーパーユーザー」アクセスというものは存在しません。securityadmin ロールは、ユーザーとロールを管理できる事前定義済みのロールですが、デフォルトではどのデータベースオブジェクトに対しても権限を持っていません。オブジェクトにアクセスするには、securityadmin ロールに、オブジェクトの所有者または grant オプションを持つ別のロールから、適切な権限を明示的に付与する必要があります。
ユーザーは、現在および将来の権限付与をサポートし、オブジェクト所有者のみが他のロールに権限を付与できるようにするために、管理アクセス スキーマを作成できます。これは、管理アクセス スキーマの定義と矛盾するため、正しくありません。管理アクセス スキーマでは、オブジェクト所有者は自分のオブジェクトに対する権限を他のロールに付与することはできず、スキーマ所有者または MANAGE GRANTS 権限を持つロールのみが付与できます。
アクセス制御の概要
Snowflakeのロールベースアクセス制御のための機能的アプローチ
Snowflakeのロールベースアクセス制御を簡素化
SnowflakeのRBACセキュリティは、ロール構成よりもロール継承を優先する。
Snowflakeのロールベースアクセス制御の概要

質問: **96**
Snowflakeアカウントには、以下のパラメータがあります。
* MIN_DATA_RETENTION_TIME_IN_DAYS はアカウントレベルで 5 に設定されています。
* DATA_RETENTION_TIME_IN_DAYS は、データベース DB1 では 4 に設定され、DB1 の Schema1 では 6 に設定されています。
* データベース DB2 の DATA_RETENTION_TIME_IN_DAYS が 5 に設定され、DB2 の Schema2 が 8 に設定された場合、結果はどうなりますか?
A. DB1とSchema1は5日間保持され、DB2とSchema2は5日間保持されました。
B. DB1とSchema1は4日間保持され、DB2とSchema2は5日間保持されました。
C. DB1は4日間、Schema1は6日間保持され、DB2は5日間、Schema2は8日間保持されました。
D. DB1は5日間、Schema1は6日間保持され、DB2は5日間、Schema2は8日間保持されました。
正解: **(正解を表示します)**
Snowflake SnowPro Architect試験範囲および公開されているすべての資料に基づいた、包括的かつ詳細な150～250語の説明：
Snowflake Time Travelの保持期間は、パラメータ階層とエディション依存の制限によって管理されますが、このシナリオにおける重要な動作は、最小保持期間のガードレールです。
MIN_DATA_RETENTION_TIME_IN_DAYS は下限値を設定します。オブジェクトは、この最小値より小さい DATA_RETENTION_TIME_IN_DAYS 値を効果的に使用することはできません。アカウント レベルの最小値が 5 日の場合、データベース レベルの設定が 4 日だと許容される最小値を下回るため、DB1 の実効保持期間は 4 日にはならず、少なくとも 5 日である必要があります。一方、Schema1 は 6 日に設定されており、これは最小値を超えているため、スキーマ レベルで適用されるはずです (スキーマは、許容範囲内で親データベースの設定を上書きできます)。DB2 の場合、データベース設定は 5 日で、これは最小値と一致し、有効です。Schema2 の 8 日も有効で、該当する場合はそのスキーマの下にあるオブジェクトに適用されます。アーキテクチャと運用の観点から、これは試験に関連する 2 つのポイントを強調しています。(1) 保持は階層パラメーター (アカウント # データベース # スキーマ # テーブル) によって制御され、(2) 最小保持設定は環境全体でガバナンス制約を強制し、リカバリ要件に違反する可能性のある過度に積極的な削減を防止します。

質問: **97**

ある小売企業は、3000以上の店舗をすべて同じPOS（販売時点情報管理）システムで運用しています。同社は、カテゴリーマネージャーにほぼリアルタイムの売上データを提供したいと考えています。店舗は様々なタイムゾーンにまたがって営業しており、毎分発生する取引件数は店舗によって大きく異なり、売上高も店舗ごとに変動しています。
売上実績は、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されたフィールドを使用して、統一された形式で提供されます。計算には、例外処理、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。集計の元データは1億行を超えています。
POSシステムは1分ごとに、すべての販売取引ファイルをクラウドストレージに送信します。ファイル名には、店舗番号とタイムスタンプが含まれており、ファイルに含まれる取引セットを識別するために使用されます。ファイルサイズは通常10MB未満です。
カテゴリーマネージャーにほぼリアルタイムの結果を提供するにはどうすればよいでしょうか？ 2つ選択してください）
A. マイクロインジェクションを避けるため、すべてのファイルはSnowflakeに取り込む前に連結する必要があります。
B. Snowpipeを作成し、AUTO_INGEST = trueに設定する必要があります。ストリームメタデータを使用してストア番号とタイムスタンプを通知し、単一のターゲットテーブルへのINSERT処理を行う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のストリームを作成し、リアルタイム分析のニーズに合った頻度で実行され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、リアルタイム分析のニーズに合った頻度でデータを処理するSnowSQLコマンドを発行する必要があります。
E. ほぼリアルタイムの要件を満たすには、1秒ごとに実行されるようにスケジュールされたタスクを含むコピーイントコマンドを使用する必要があります。
正解: **(**正解を表示します**)**
カテゴリーマネージャーにほぼリアルタイムの販売結果を提供するために、アーキテクトは以下の手順を使用できます。
* POSが販売取引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。
* 外部ステージから Snowflake のターゲット テーブルにファイルをロードする Snowpipe を作成します。Snowpipe は AUTO_INGEST = true に設定する必要があります。これは、外部ステージに到着した新しいファイルを自動的に検出して取り込むことを意味します。Snowpipe はコピーも使用する必要があります。
* 重複した取り込みを避けるため、読み込み後に外部ステージからファイルを削除するオプション3
* Snowpipe によって行われた INSERT をキャプチャするストリームをターゲット テーブル上に作成します。このストリームには、ファイル名、パス、サイズ、最終更新日時に関する情報を提供するメタデータ列を含める必要があります。また、ストリームにはリアルタイム分析のニーズに合致する保持期間を設定する必要があります。4
ほぼリアルタイムのデータを処理するために、ストリームに対してクエリを実行するタスクを作成します。このクエリは、ストリームのメタデータを使用してファイル名とパスから店舗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。また、クエリの結果は、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出力する必要があります。このタスクは、1分ごと、または5分ごとなど、リアルタイム分析のニーズに合わせて実行されるようにスケジュールする必要があります。
他の選択肢は、ほぼリアルタイムの結果を提供するには最適でも実現可能でもない。
* マイクロインジェストを避けるため、すべてのファイルはSnowflakeに取り込む前に連結する必要があります。ただし、このオプションはデータパイプラインに余分な遅延と複雑さをもたらすため、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、ファイルのマージ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への新規ファイルの取り込みが遅延し、データ損失や破損のリスクが高まります。さらに、Snowpipeは連結された各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インジェクションは回避されません。

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、リアルタイム分析のニーズに合った頻度でデータを処理するSnowSQLコマンドを発行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に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須ではありません。外部スケジューラを使用すると、データパイプラインにオーバーヘッドと依存関係が増え、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、ほぼリアルタイムでの取り込みが保証されません。
* ほぼリアルタイムの要件を満たすには、1秒ごとに実行され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1秒ごとに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現不可能です。タスクの最小間隔は1分ですが、タスクはスケジューリングの遅延や同時実行制限の影響を受けるため、それさえも保証されません。さらに、タスクを含む copy into コマンドを使用すると、自動ファイル検出、負荷分散、マイクロパーティション最適化などの Snowpipe の利点を活用できません。参考資料:
* 1 :SnowPro Advanced :Architect｜学習ガイド
* 2: Snowflake ドキュメント | ステージの作成
* 3: Snowflake ドキュメント | Snowpipe を使用したデータの読み込み
* 4: Snowflake ドキュメント | ELT におけるストリームとタスクの使用
* : Snowflake ドキュメント | タスクの作成
* : Snowflake ドキュメント | データロードのベストプラクティス
* : Snowflake ドキュメント | Snowpipe REST API の使用方法
* : Snowflake ドキュメント | タスクのスケジュール設定
* : SnowPro Advanced: Architect | 学習ガイド
* : ステージの作成
* : Snowpipeを使用してデータをロードします
* : ELT にストリームとタスクを使用する
* : [タスクの作成]
* : [データ読み込み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]

質問: **98**
大手製造会社のデータエンジニアリングチームは、以下のような多様なユースケースとデータ利用者の要件をサポートするために、多くのソースから得られるデータをエンジニアリングする必要があります。
1) レポート作成と可視化を必要とする財務およびベンダー管理チームのメンバー
2) 機械学習モデル開発のために生データへのアクセスが必要なデータサイエンスチームのメンバー
3) データ収益化のために、設計され保護されたデータを必要とする営業チームメンバー。これらの要件を満たすSnowflakeのデータモデリング手法はどれですか？ 2つ選択してください。）
A. 会社のデータレイクにデータを統合し、外部テーブルを使用します。
B. データパイプラインに入力される生データを取り込んで永続化するための生データベースを作成します。
C. 使用パターンに合わせてデータを整理した、プロファイル固有のデータベースセットを作成します。
D. すべての消費者の要件をサポートするために、単一のデータベースに単一のスター型スキーマを作成します。
E. データ ボールトを唯一のデータ パイプライン エンドポイントとして作成し、すべてのコンシューマーがボールトに直接アクセスできるようにします。
正解: **(**正解を表示します**)**

説明
Snowflakeでは、データレイク環境におけるデータモデリングに、以下の2つのアプローチを推奨しています。生データベースを作成することで、データエンジニアリングチームは、さまざまなソースから変換やクレンジングを行わずにデータを取り込み、元のデータ品質とフォーマットを維持できます。これにより、データサイエンスチームは機械学習モデル開発のために生データにアクセスできるようになります。プロファイル固有のデータベースセットを作成することで、データエンジニアリングチームは、さまざまなユースケースやデータ利用者の要件に応じて、異なる変換や最適化を適用できます。たとえば、財務 -ベンダー管理チームはレポート作成や可視化をサポートするディメンションデータベースにアクセスできる一方、営業チームはデータ収益化をサポートするセキュアデータベースにアクセスできます。
参考文献：
* Snowflakeデータレイクアーキテクチャ | Snowflakeドキュメント
* Snowflakeデータレイクのベストプラクティス | Snowflakeドキュメント

質問: **99**
A company is using Snowflake in Azure in the Netherlands. The company analyst team also has data in JSON format that is stored in an Amazon S3 bucket in the AWS Singapore region that the team wants to analyze.
The Architect has been given the following requirements:
1. Provide access to frequently changing data
2. Keep egress costs to a minimum
3. Maintain low latency
How can these requirements be met with the LEAST amount of operational overhead?
A. Use AWS Transfer Family to replicate data between the S3 bucket in AWS Singapore and an Azure Netherlands Blob storage, then use an external table against the Blob storage.
B. Use an external table against the S3 bucket in AWS Singapore and copy the data into transient tables.
C. Copy the data between providers from S3 to Azure Blob storage to collocate, then use Snowpipe for data ingestion.
D. Use a materialized view on top of an external table against the S3 bucket in AWS Singapore.
正解: **(**正解を表示します**)**

質問: **100**
ある大手製造会社は、事業部門全体で12個の個別のSnowflakeアカウントを運用している。同社は、サプライチェーンの最適化を支援し、複数のベンダーとの購買交渉力を高めるために、データ共有のレベルを向上させたいと考えている。
同社のSnowflakeアーキテクトは、各事業部門が共有する情報を決定できると同時に、構成と管理にかかる労力を最小限に抑えるソリューションを設計する必要がある。欧州を拠点とする一部の部門を除き、ほとんどの事業部門は同じクラウド環境でSnowflakeアカウントを使用している。
Snowflakeが推奨するベストプラクティスによると、これらの要件はどのように満たすべきでしょうか？
A. 欧州のアカウントをグローバルリージョンに移行し、接続されたグラフアーキテクチャで共有を管理します。データ交換をデプロイします。
B. 欧州アカウントのデータ共有と組み合わせたプライベートデータ交換を導入する。
C. すべてのセキュアビューでinvoker_share()が使用されていることを確認して、Snowflake Marketplaceにデプロイします。
D. プライベートデータ交換を導入し、レプリケーションを使用して交換内でヨーロッパのデータ共有を可能にします。
正解: **(**正解を表示します**)**
Snowflake が推奨するベストプラクティスによると、大規模製造企業の要件は、欧州アカウントのデータ共有と組み合わせてプライベートデータエクスチェンジを導入することで満たされるはずです。プライベートデータエクスチェンジは、Snowflake Data Cloud プラットフォームの

機能で、組織間でデータを安全かつ統制された方法で共有することを可能にします。これにより、Snowflake の顧客は独自のデータハブを作成し、組織の他の部門や外部パートナーを招待してデータセットにアクセスし、提供することができます。プライベートデータエクスチェンジは、エクスチェンジで共有されるデータに対して、一元管理、きめ細かなアクセス制御、およびデータ使用状況のメトリクスを提供します 1。データ共有は、データをコピーしたり移動したりすることなく、Snowflake アカウント間でデータを安全かつ直接的に共有する方法です。データ共有を使用すると、データプロバイダーは、アカウント内の選択したオブジェクトに対する権限を、他のアカウントの 1 つ以上のデータコンシューマーに付与できます 2。プライベートデータエクスチェンジとデータ共有を組み合わせて使用することで、企業は次のメリットを実現できます。
* 各事業部門は、共有するデータを決定し、プライベートデータ交換プラットフォームに公開できます。公開されたデータは、交換プラットフォームの他のメンバーが検索してアクセスできます。これにより、複数のデータ共有関係や構成を管理する手間と複雑さが軽減されます。
* 企業は、同じクラウド環境にある既存のSnowflakeアカウントを活用してプライベートデータエクスチェンジを作成し、メンバーを招待することができます。これにより、移行とセットアップのコストを最小限に抑え、既存のSnowflakeの機能とセキュリティを活用できます。
* 同社はデータ共有機能を利用して、異なる地域やクラウドプラットフォームにある欧州のアカウントとデータを共有できます。これにより、同社はデータ主権とプライバシーに関する地域および規制上の要件を遵守しつつ、組織全体でのデータ連携を実現できます。
同社はSnowflake Data Cloudプラットフォームを利用して、共有データに対するデータ分析や変換を実行できるほか、他のデータソースやアプリケーションとの統合も可能です。これにより、サプライチェーンを最適化し、複数のベンダーとの購買交渉力を高めることができます。
他のオプションは、要件を満たしていないか、ベストプラクティスに従っていないため、誤りです。オプションAは、欧州アカウントをグローバルリージョンに移行すると、データ主権とプライバシー規制に違反する可能性があり、データエクスチェンジを導入しても、企業が必要とするレベルの制御と管理が実現できない可能性があるため、誤りです。オプションCは、Snowflake Marketplaceに導入すると、企業のデータが望ましくない消費者に公開される可能性があり、セキュアビューでinvoker_share()を使用しても、望ましいレベルのセキュリティとガバナンスが実現できない可能性があるため、誤りです。オプションDは、エクスチェンジで欧州のデータ共有を可能にするためにレプリケーションを使用すると、追加のコストと複雑さが発生する可能性があり、代わりにデータ共有を使用できる場合は不要になる可能性があるため、誤りです。参照：プライベートデータエクスチェンジ｜Snowflakeドキュメント、セキュアデータ共有の概要｜Snowflakeドキュメント

質問**: 101**
ある建築家は、Snowflakeの本番環境と品質保証環境を分離するために、2つの別々のSnowflakeアカウントを使用することを選択しました。
QAアカウントは、データおよびデータベースオブジェクトに対する変更を、本番アカウントに反映させる前に実行および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およびデータは、少なくとも毎晩、本番アカウントのデータベースオブジェクト（権限およびデータを含む）と完全に一致するコピーである必要があります。
毎晩、本番環境アカウントのデータとデータベースオブジェクトをQAアカウントに投入するために使用する最も単純な方法はどれですか？
A. 1) Create a share in the Production account for each database
2) Share access to the QA account as a Consumer
3) The QA account creates a database directly from each share
4) Create clones of those databases on a nightly basis
5) Run tests directly on those cloned databases
B. 1) In the Production account, create an external function that connects into the QA account and returns all the data for one specific table
2) Run the external function as part of a stored procedure that loops through each table in the Production account and populates each table in the QA account
C. 1）本番アカウントの各データベースでレプリケーションを有効にする
2) Create replica databases in the QA account
3) Create clones of the replica databases on a nightly basis

4) Run tests directly on those cloned databases
D. 1) Create a stage in the Production account
2) Create a stage in the QA account that points to the same external object-storage location
3) Create a task that runs nightly to unload each table in the Production account into the stage
4) Use Snowpipe to populate the QA account
正解**: (**正解を表示します**)**

質問**: 102**
Snowflake内に新しいユーザーuser_01が作成されます。以下の2つのコマンドが実行されます。
コマンド1-> ユーザーuser_01への権限を表示する;
コマンド2 ~> show grants on user user 01;
これらのコマンドからどのような推論が導き出せるだろうか？
A. コマンド1は、user_01を所有するユーザーを定義します。
コマンド2は、user_01に付与されたすべての権限を定義します。
B. コマンド1はuser_01に付与されるすべての権限を定義します。コマンド2はuser_01を所有するユーザーを定義します。
C. コマンド1は、user_01を所有するロールを定義します。
コマンド2は、user_01に付与されたすべての権限を定義します。
D. コマンド1はuser_01に付与されるすべての権限を定義します。コマンド2はuser_01が所有するロールを定義します。
正解**: D (**コメントを発表する**)**
Snowflake の SHOW GRANTS コマンドを使用すると、ロール、ユーザー、および共有に対して明示的に付与されたすべてのアクセス制御権限を一覧表示できます。コマンドの構文と出力は、コマンドで指定されたオブジェクトの種類と付与対象の種類によって異なります。この質問では、2 つのコマンドは次の意味を持ちます。
* コマンド 1: show grants to user user_01; このコマンドは、ユーザー user_01 に付与されているすべてのロールを一覧表示します。
出力には、各権限付与について、ロール名、権限付与対象者名、および権限付与元ロール名が含まれます。このコマンドは、user_01が現在のユーザー1である場合、show grants to user current_userコマンドと同等です。
* コマンド 2: show grants on user user_01; このコマンドは、ユーザー オブジェクト user_01 に付与されているすべての権限を一覧表示します。出力には、各権限について、権限名、権限付与者名、および権限付与元ロール名が含まれます。このコマンドは、ユーザー オブジェクト user_01 を所有するロールを示します。所有者ロールには、ユーザー オブジェクト 2 を変更または削除する権限があります。
したがって、正しい推論は、コマンド1はuser_01に付与されるすべての権限を定義し、コマンド2はuser_01を所有するロールを定義する、というものである。
参考文献：
* 助成金について
* Snowflakeにおけるアクセス制御の理解

質問**: 103**
ストレージ統合を作成する目的は何ですか？ 3つ選択してください。）
A. クラウドプロバイダーのキー管理サービスで管理されているマスター暗号化キーを使用して、Snowflakeデータへのアクセスを制御します。
B. Snowflakeアカウントをホストするクラウドプロバイダーに関係なく、外部クラウドプロバイダー用に生成されたIDおよびアクセス管理（IAM）エンティティを保存します。
C. 1つのSnowflakeオブジェクトを使用して複数の外部ステージをサポートします。

D. ステージを作成する際、またはデータのロードやアンロードを行う際に、認証情報を提供しないようにしてください。
E. パブリックインターネットを経由せずに、VPC 間で直接かつ安全な接続を可能にするプライベート VPC エンドポイントを作成します。
F. 複数のクラウドプロバイダーの認証情報を単一のSnowflakeオブジェクトで管理します。
正解: **(**正解を表示します**)**
説明
* ストレージ統合とは、Amazon S3、Google Cloud Storage、Microsoft Azure Blob Storageなどの外部クラウドプロバイダー用に生成されたIDおよびアクセス管理（IAM）エンティティを格納するSnowflakeオブジェクトです。この統合により、Snowflakeは外部ステージ1で参照される外部ストレージの場所からデータを読み取ったり、そこにデータを書き込んだりすることができます。
ストレージ統合を作成する目的の1つは、1つのSnowflakeオブジェクトを使用して複数の外部ステージをサポートすることです。統合では、ユーザーが統合を使用する外部ステージを作成する際に指定できる場所を制限するバケット（およびオプションのパス）をリストできます。多くの外部ステージオブジェクトは、異なるバケットとパスを参照し、認証に同じストレージ統合を使用できることに注意してください1。
したがって、選択肢Cが正解です。
* ストレージ統合を作成するもう 1 つの目的は、ステージの作成時やデータのロードまたはアンロード時に認証情報を提供する必要がないようにすることです。統合は、名前付きの第一級 Snowflake オブジェクトであり、秘密鍵やアクセス トークンなどの明示的なクラウド プロバイダー認証情報を渡す必要がありません。統合オブジェクトには IAM ユーザー ID が格納され、組織の管理者がクラウド プロバイダー アカウント 1 で IAM ユーザーの権限を付与します。したがって、オプション D が正解です。
* ストレージ統合を作成する3つ目の目的は、Snowflakeアカウントをホストしているクラウドプロバイダーに関係なく、外部クラウドプロバイダー用に生成されたIAMエンティティを保存することです。たとえば、SnowflakeアカウントがAzureやGoogle Cloud Platformでホストされている場合でも、Amazon S3用のストレージ統合を作成できます。これにより、Snowflake1を使用して異なるクラウドプラットフォーム間でデータにアクセスできるようになります。
したがって、選択肢Bが正解です。
* オプション A は誤りです。ストレージ統合を作成しても、マスター暗号化キーを使用して Snowflake データへのアクセスを制御することはできません。Snowflake は階層型キー モデルを使用してすべてのデータを暗号化し、マスター暗号化キーは Snowflake またはクラウド プロバイダーのキー管理サービスを使用して顧客が管理します。これはストレージ統合機能とは無関係です。
* オプションEは誤りです。ストレージ統合を作成しても、プライベートVPCエンドポイントは作成されません。
プライベートVPCエンドポイントは、パブリックインターネットを経由せずにVPC間で直接かつ安全な接続を可能にするネットワーク構成オプションです。これはストレージ統合機能3とは独立しています。
* オプション F は誤りです。ストレージ統合を作成しても、複数のクラウド プロバイダーの認証情報を単一の Snowflake オブジェクトで管理することはできません。ストレージ統合は 1 つのクラウド プロバイダーに固有のものであり、アクセスしたいクラウド プロバイダーごとに個別の統合を作成する必要があります。4
参考資料: 暗号化と復号化 : Snowflake のプライベートリンク : ストレージ統合の作成 : オプション 1: Amazon S3 にアクセスするための Snowflake ストレージ統合の設定

質問**: 104**
ある企業は、さまざまなIoT操作用のJSONレコードを提供するソースシステムを持っています。JSONは、バリアント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、パフォーマンスが問題になりつつあります。バリアントフィールド内のcreate_dateキーでフィルタリングするために、汎用的なアクセスパターンが使用されています。
パフォーマンスを向上させるにはどうすればよいでしょうか？

A. 対象テーブルを変更して、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します。これにより、データ型がタイムスタンプのcreate_dateフィールドが追加されます。このフィールドがフィルタで使用されると、パーティションプルーニングが実行されます。
B. JSONレコードから取得した追加フィールドを対象テーブルに追加します。これには、データ型がvarcharの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で使用されると、パーティションプルーニングが実行されます。
C. 使用するデータウェアハウスのサイズを検証します。レコード数が数億件に近づく場合、この量のデータを処理するために必要な最小サイズはXL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取り入れます。ユーザーまたはプロセスが特定の日付範囲を照会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るようにしてください。
正解: **(**正解を表示します**)**
正解はAです。スキャンおよび処理されるデータ量を削減することで、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リングし、フィルタ条件に一致しないマイクロパーティションを削除できます。これにより、JSONデータを解析してレコードごとにvariant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を向上させないため、誤りです。varchar型のcreate_dateフィールドを追加しても、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、フィルタ条件に一致しないマイクロパーティションを削除したりすることはできません。そのため、JSONデータを解析し、すべてのレコードのvariant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、誤りです。Snowflakeは、使用するデータウェアハウスのサイズを検証することで、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。しかし、これによってスキャンおよび処理されるデータ量が削減されるわけではなく、これがJSONデータに対するクエリの主なボトルネックとなっています。
オプションDは、データロードとクエリ処理に不要な複雑さとオーバーヘッドを追加するため、誤りです。日付範囲でパーティション分割された複数のテーブルを使用することで、Snowflakeは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量を削減できます。ただし、これには複数のテーブルの作成と維持、日付に基づいて適切なテーブルへのデータのロード、および複数の日付範囲にまたがるクエリのためのテーブル結合が必要です。参照:
Snowflakeドキュメント :Snowpipeを使用したデータロード :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、COPY INTOコマンドの構文と使用方法についても説明します。このコマンドは、ON_ERROR、PURGE、SKIP_FILEなど、ロード動作を制御するためのさまざまなオプションとパラメータをサポートしています。
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扱う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の設定方法と変更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内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、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、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。

質問**: 105**

Snowflakeのタスクでは、夏時間による現地時間の変更はどのように処理されますか？ 2つ選択してください。）
A. UTCベースのスケジュールでスケジュールされたタスクは、時間の変更による問題は発生しません。
B. タスクスケジュールは、時間変更に対応するために、指定されたタイムゾーンまたは現地のタイムゾーンに従うように設計できます。
C. 夏時間への変更中は、タスクは一時停止状態になります。
D. 数分ごとの頻繁なタスク実行スケジュールは問題を引き起こさないかもしれませんが、タスク履歴に影響を与えます。
E. タスク スケジュールは指定された時間のみに従い、時間の損失や重複には対応できません。
正解: **(**正解を表示します**)**
SnowflakeのドキュメントとWeb検索結果によると、Snowflakeタスクにおける夏時間による現地時間の変更の処理方法について、以下の2つの記述は正しい。タスクとは、SnowflakeでSQLステートメントまたはストアドプロシージャをスケジュールして実行できるようにする機能である。タスクは、タスクの実行頻度とタイムゾーンを指定するcron式を使用してスケジュールできる。
UTCベースのスケジュールでスケジュールされたタスクは、時刻変更による影響を受けません。UTCは夏時間を適用しない世界標準の時刻です。そのため、タイムゾーンとしてUTCを使用するタスクは、現地の時刻変更に関係なく、年間を通して同じ時刻に実行されます。
タスクスケジュールは、時間変更に対応するため、指定された時間帯または現地の時間帯に合わせて設計できます。
Snowflakeは、タスクのcron式で有効なIANAタイムゾーン識別子を使用することをサポートしています。これにより、タスクは指定されたタイムゾーンの現地時間に従って実行され、夏時間調整も含まれる場合があります。たとえば、タイムゾーンとしてEurope/Londonを使用するタスクは、現地時間がGMTとBST12の間で切り替わると、1時間早くまたは遅く実行されます。
Snowflakeドキュメント :タスクのスケジュール設定
Snowflakeコミュニティ :Snowflakeでタスクをスケジュールする際に使用されるタイムゾーンは、夏時間に対応していますか？

質問: **106**
データロード変換は、ステージからテーブルへのデータコピーの一部として、ユーザーステージと名前付きステージ（内部および外部）からのデータの選択のみをサポートします。
A. 偽
B. 真
正解: **B (**コメントを発表する**)**

質問: **107**
ストレージ統合を作成する目的は何ですか？ 3つ選択してください。）
A. パブリックインターネットを経由せずに、VPC 間で直接かつ安全な接続を可能にするプライベート VPC エンドポイントを作成します。
B. ステージを作成する際、またはデータのロードやアンロードを行う際に、認証情報を提供しないようにしてください。
C. Snowflakeアカウントをホストするクラウドプロバイダーに関係なく、外部クラウドプロバイダー用に生成されたIDおよびアクセス管理（IAM）エンティティを保存します。
D. 1つのSnowflakeオブジェクトを使用して複数の外部ステージをサポートします。
E. クラウドプロバイダーのキー管理サービスで管理されているマスター暗号化キーを使用して、Snowflakeデータへのアクセスを制御します。

F. 複数のクラウドプロバイダーの認証情報を単一のSnowflakeオブジェクトで管理します。
正解: **(**正解を表示します**)**

質問: **108**
ある企業が、クラウドストレージに保存されているデータのためのサービス提供層を設計している。数テラバイトに及ぶデータはレポート作成に使用される予定だ。一部のデータには明確なユースケースはないものの、実験的な分析に役立つ可能性がある。
この実験データは頻繁に変更され、数日のうちに完全に消去されて新しいデータに置き換えられることもある。
同社は、アクセス制御を一元化し、エンドユーザーへの単一の接続ポイントを提供し、データガバナンスを維持したいと考えている。
コスト、管理作業、開発コストを最小限に抑えつつ、これらの要件を満たすソリューションとはどのようなものでしょうか？
A. レポート作成に使用するデータを、ネイティブテーブルを含む Snowflake スキーマにインポートします。次に、実験データに使用するクラウドストレージフォルダを指す外部テーブルを作成します。その後、異なるユーザーペルソナに合わせて、異なるデータセットへの権限を付与した 2 つの異なるロールを作成し、対応するユーザーにこれらのロールを付与します。
B. レポートに使用するクラウドストレージ内のすべてのデータを、ネイティブテーブルを持つ Snowflake スキーマにインポートします。次に、このスキーマへのアクセス権を持つロールを作成し、そのロールを通じてデータへのアクセスを管理します。
C. レポートに使用するクラウドストレージ内のすべてのデータを、ネイティブテーブルを持つ Snowflake スキーマにインポートします。次に、異なるユーザーペルソナに合わせて、異なるデータセットへのアクセス権限を持つ2つの異なる役割を作成し、これらの役割を対応するユーザーに付与します。
D. レポート作成に使用するデータを、ネイティブテーブルを含む Snowflake スキーマにインポートします。次に、実験データ用のクラウドストレージファイルを指す SELECT コマンドを含むビューを作成します。その後、異なるユーザーペルソナに合わせて 2 種類のロールを作成し、それぞれのロールを対応するユーザーに付与します。
正解: **A (**コメントを発表する**)**
最も費用対効果が高く、管理効率の良いソリューションは、ネイティブテーブルと外部テーブルを組み合わせることです。レポートデータ用のネイティブテーブルはパフォーマンスとガバナンスを確保し、外部テーブルは頻繁に変更される実験データに対する柔軟性を提供します。
データセットへの特定の権限を持つロールを作成することで、最小権限の原則に沿い、アクセス制御を一元化し、ユーザー管理を簡素化できます12。
参考文献
*Snowflakeのコスト最適化に関するドキュメント1。
*Snowflakeのコスト管理に関するドキュメント2。

質問: **109**
Snowflakeのアーキテクトが、複数アカウントに対応した設計戦略を策定している。
この戦略は、どのシナリオにおいて最も費用対効果が高いでしょうか？ 2つ選択してください。）
A. 会社は、AWS 上に存在する本番データベースを、Azure 上に存在する開発データベースにクローンしたいと考えています。
B. 会社は2つのデータベース間でデータを共有する必要がありますが、一方のデータベースはペイメントカード業界データセキュリティ基準（PCI DSS）に準拠している必要がありますが、もう一方はそうではありません。
C. 会社は、開発、テスト、および本番環境向けに、異なる役割ベースのアクセス制御機能をサポートする必要があります。
D. 会社のセキュリティポリシーでは、開発環境、テスト環境、本番環境で異なるActive Directoryインスタンスを使用することが義務付けられています。
E. 会社は、特定のユーザーに対して特定のネットワーク ポリシーを使用して、特定の IP アドレスを許可およびブロックする必要があります。

正解: (正解を表示します)
B: PCI DSS コンプライアンスに対応する際、アカウントを分けることは、機密データを扱う環境とそうでない環境を強力に分離できるため有益です。準拠リソースと非準拠リソースを分離することで、組織はコンプライアンスの範囲を限定でき、コスト効率の良い戦略となります。D. 異なる Active Directory インスタンスは、異なるアカウントに分けることで、より効果的かつ安全に管理できます。このアプローチにより、個別の ID およびアクセス管理ポリシーを設定でき、セキュリティ要件を強制し、環境間のアクセス ポリシー エラーのリスクを最小限に抑えることができます。

質問: **110**
外部テーブルスキーマに含めることができる列はどれですか？ 3つ選択してください。）
A. 価値
B. METADATASROW_ID
C. メタデータ更新
D. メタデータ A$ ファイル名
E. メタデータ ファイル行番号
F. メタデータ外部テーブルパーティション
正解: **A,D,E** (コメントを発表する)
外部テーブルスキーマは、外部ステージに格納されるデータの列とデータ型を定義します。すべての外部テーブルには、デフォルトで以下の列が含まれます。
* VALUE: 外部ファイル内の単一行を表す VARIANT 型の列。
* METADATA$FILENAME: 外部テーブルに含まれる各ステージング済みデータ ファイルの名前と、ステージ内でのパスを識別する擬似列。
* METADATA$FILE_ROW_NUMBER: ステージングされたデータ ファイル内の各レコードの行番号を示す擬似列。
VALUE列や擬似列を使用して、式として追加の仮想列を作成することもできます。ただし、以下の列は外部テーブルでは無効であり、スキーマに含めることはできません。
* METADATASROW_ID: この列は内部テーブルでのみ使用可能で、テーブル内の各行の一意の識別子を示します。
* METADATASISUPDATE: この列は内部テーブルでのみ使用可能で、行がマージ操作によって挿入されたか更新されたかを示します。
* METADATASEXTERNAL TABLE PARTITION: この列は有効な列名ではなく、Snowflake には存在しません。
外部テーブルの概要、CREATE EXTERNAL TABLE

質問: **111**
テーブル contacts_data には、ネストされた JSON データを含む VARIANT 型の列があります。
名の値 Snow」を正しく取得するクエリはどれですか？ 2つ選択してください）。
A. SELECT o[customer][name][first]::STRING FROM contacts_data;
B. SELECT o['customer']['name']['first']::STRING FROM contacts_data;
C. SELECT o:['customer']:['name']:['first']::STRING FROM contacts_data;
D. SELECT o:Customer.Name.First::STRING FROM contacts_data;
E. SELECT o:customer.name.first::STRING FROM contacts_data;
正解: (正解を表示します)
Snowflakeは、VARIANT列の要素にアクセスするための複数の構文をサポートしています。引用符付きキーを使用した角括弧表記はオブジェクトの走査に有効であり、オプションBが正解です。オプションEに示すように、コロン構文を使用したドット表記も有効であり、可読性を高めるためによく使用されます。

オプションAは、括弧内の引用符なしの識別子がJSONキーではなく列名として解釈されるため無効です。オプションCは、括弧とコロンを組み合わせた誤った構文が含まれています。オプションDは、JSONキー名は引用符で囲まれていない限り大文字と小文字が区別されるのに対し、指定された構造では小文字のキーが使用されているため無効です。
VARIANTトラバーサル構文を理解することは、半構造化データの処理と変換を行うSnowPro Architectにとって不可欠なスキルです。

質問: **112**
ある建築家がSnowflakeを使ってデータレイクを設計しています。同社は構造化データ、半構造化データ、非構造化データを保有しています。同社はSnowflakeシステム内のデータレイクにデータを保存したいと考えています。また、Snowflakeのデータ共有機能を利用して、社内各支社間でデータを共有することを計画しています。
Snowflake内で非構造化データを共有する際に考慮すべき点は何ですか？
A. 安全なビューを介してデータを共有するために、署名付きURLを使用して非構造化データをSnowflakeに保存する必要があります。URLには時間制限はありません。
B. 安全なビューを介してデータを共有するために、スコープ付き URL を使用して非構造化データを Snowflake に保存する必要があります。URL の有効期限は 24 時間です。
C. セキュアビューを介してデータを共有するために、非構造化データを Snowflake に保存するにはファイル URL を使用する必要があります。URL の有効期限は 7 日間です。
D. セキュアなビューを介してデータを共有するため、非構造化データを Snowflake に保存するにはファイル URL を使用する必要があります。URL の有効期限には「expiration_time」引数が定義されます。
正解: **B** (コメントを発表する)
Snowflake内で非構造化データを共有する場合、スコープ付きURLの使用をお勧めします。スコープ付きURLを使用すると、ステージ自体に権限を付与することなく、ステージングされたファイルへの一時的なアクセスが可能になり、セキュリティが強化されます。URLは、永続化されたクエリ結果の期間が終了すると期限切れになります。現在の有効期限は24時間です。この方法は、Snowflakeのデータ共有フレームワーク内で、セキュアなビューを介して非構造化データを共有するのに適しています。
参考資料 :この回答は、非構造化データの共有とスコープ付きURLの使用に関するSnowflakeの公式ドキュメントに基づいています1。

質問: **113**
アーキテクトは、Snowflake Kafkaコネクタを使用してイベントデータをSnowflakeにストリーミングするパイプラインを設計しています。
アーキテクトの最優先事項は、最もコスト効率の良い方法でデータをストリーミングするようにコネクタを構成することです。
Snowflake Kafkaコネクタに関連するコストを最適化するために推奨される方法は次のうちどれですか？
A. コネクタ構成で Buffer.flush.time を高く設定してください。
B. コネクタ構成で Buffer.size.bytes の値を大きくします。
C. コネクタ構成で Buffer.size.bytes を小さくします。
D. コネクタ構成で Buffer.count.records の値を小さくします。
正解: **(**正解を表示します**)**
説明
buffer.flush.time プロパティでサポートされる最小値は 1 (秒) です。平均データフローレートが高い場合は、レイテンシを改善するためにデフォルト値を小さくすることをお勧めします。レイテンシよりもコストが重要な場合は、バッファのフラッシュ時間を長くすることができます。メモリ不足例外を避けるため、Kafka メモリバッファがいっぱいになる前にフラッシュするように注意してください。https://docs.snowflake.com/en/user-guide/data-load-snowpipe-streaming-kafka

質問: **114**
テーブルには5つの列があり、数百万件のレコードが含まれています。列のカーディナリティ分布は以下のとおりです。

Column	Number of Distinct Values
C1	10,790
C2	108
C3	302,605
C4	1.117,736
C5	2.205,400

列C4とC5は、主にSELECTクエリのGROUP BY句とORDER BY句で使用されます。一方、列C1、C2、C3は、SELECTクエリのフィルタ条件と結合条件で頻繁に使用されます。
アーキテクトは、クエリのパフォーマンスを向上させるために、このテーブルのクラスタリングキーを設計する必要があります。
Snowflakeの推奨事項に基づくと、複数列のクラスタリングキーを定義する際に、クラスタリングキーの列はどのように順序付けるべきでしょうか？
A. C3、C4、C5
B. C1、C3、C2
C. C2、C1、C3
D. C5、C4、C2
正解: **C** (コメントを発表する)

質問: **115**
Snowflakeにおける標準的な仮想倉庫ポリシーはどのように機能しますか？
A. 追加のクラスターを起動するのではなく、実行中のクラスターを常にフルロード状態にしておくことでクレジットを節約します。
B. システムが、少なくとも 6 分間クラスタをビジー状態にするクエリ負荷があると推定した場合にのみ開始します。
C. システムが、少なくとも 2 分間クラスタをビジー状態にするクエリ負荷があると推定した場合にのみ開始します。
D. クレジットを節約するのではなく、追加のクラスタを起動することでキューイングを防止または最小限に抑えます。
正解: (正解を表示します)
Snowflakeのマルチクラスタウェアハウスで利用できるスケーリングポリシーは2種類あり、そのうちの1つが標準仮想ウェアハウスポリシーです。もう1つはエコノミクスポリシーです。標準ポリシーは、キュー内のクエリ数に関係なく、現在のクラスタが完全にロードされたらすぐに別のクラスタを起動することで、キューイングを防止または最小限に抑えることを目的としています。このポリシーはクエリのパフォーマンスと同時実行性を向上させることができますが、実行中のクラスタが完全にロードされた状態を維持してから別のクラスタを起動することでクレジットを節約しようとするエコノミクスポリシーよりも多くのクレジットを消費する可能性があります。スケーリングポリシーはウェアハウスの作成時または変更時に設定でき、いつでも変更できます。
参考文献：
* Snowflakeドキュメント :マルチクラスタウェアハウス
* Snowflakeドキュメント :マルチクラスタウェアハウスのスケーリングポリシー

質問: **116**

Snowflakeで使用されているロールベースアクセス制御（RBAC）の特徴は何ですか？
A. 権限はデータベースレベルで付与でき、すべての下位オブジェクトに継承されます。
B. ユーザーは、securityadmin とともに「スーパーユーザー」アクセスを使用することで、認証チェックをバイパスして、すべてのデータベース、スキーマ、および基となるオブジェクトにアクセスできます。
C. ユーザーは、将来の権限付与をサポートし、スキーマ所有者のみが他のロールに権限を付与できるようにするための管理アクセススキーマを作成できます。
D. ユーザーは、現在および将来の権限付与をサポートする管理アクセス スキーマを作成し、オブジェクトの所有者のみが他のロールに権限を付与できるようにすることができます。
正解: **(**正解を表示します**)**
ロールベースアクセス制御（RBAC）は、Snowflakeのアクセス制御フレームワークであり、オブジェクトの所有者がロールに権限を付与し、ロールをユーザーに割り当てることで、オブジェクトに対する操作を制限または許可することができます。Snowflakeで使用されるRBACの特徴は次のとおりです。
権限はデータベースレベルで付与でき、すべての基となるオブジェクトに継承されます。つまり、CREATE SCHEMAやUSAGEなど、データベース上で特定の権限を持つロールは、明示的に取り消されない限り、そのデータベース内の任意のスキーマ、テーブル、ビュー、またはその他のオブジェクトに対しても同じ操作を実行できます。これにより、アクセス制御管理が簡素化され、必要な権限付与の数を減らすことができます。
* ユーザーは、将来の権限付与をサポートし、スキーマ所有者のみが他のロールに権限を付与できるようにするために、管理アクセス スキーマを作成できます。つまり、ユーザーは管理アクセス オプションを使用してスキーマを作成でき、これにより、スキーマ内のオブジェクトの所有権と権限付与のデフォルトの動作が変更されます。管理アクセス スキーマでは、オブジェクト所有者はオブジェクトに対する権限を他のロールに付与できなくなり、スキーマ所有者または管理権限を持つロールのみが権限を付与できます。これにより、スキーマとそのオブジェクトのセキュリティとガバナンスが強化されます。
その他のオプションは、Snowflakeで使用されているRBACの特徴ではありません。
* ユーザーは、securityadmin とともに「スーパーユーザー」アクセスを使用して認証チェックをバイパスし、すべてのデータベース、スキーマ、および基となるオブジェクトにアクセスできます。これは正しくありません。
Snowflakeにおける「スーパーユーザー」アクセス。securityadminロールは、ユーザーとロールを管理できる事前定義済みのロールですが、デフォルトではデータベースオブジェクトに対する権限は一切ありません。オブジェクトにアクセスするには、securityadminロールにオブジェクトの所有者またはgrantオプションを持つ別のロールから、適切な権限を明示的に付与する必要があります。
* ユーザーは、現在および将来の権限付与をサポートし、オブジェクト所有者のみが他のロールに権限を付与できるようにするために、管理アクセス スキーマを作成できます。これは、管理アクセス スキーマの定義と矛盾するため、正しくありません。管理アクセス スキーマでは、オブジェクト所有者は自分のオブジェクトに対する権限を他のロールに付与することはできず、スキーマ所有者または MANAGE GRANTS 権限を持つロールのみが付与できます。
参考文献：
* アクセス制御の概要
* Snowflakeのロールベースアクセス制御のための機能的アプローチ
* Snowflakeのロールベースアクセス制御を簡素化
* Snowflake RBAC セキュリティは、ロール構成よりもロール継承を優先します
* Snowflakeのロールベースアクセス制御の概要

質問**: 117**

Snowflake Kafkaコネクタの認証は、以下に依存しています。
A. キーペア認証とユーザー名/パスワード認証の両方
B. ユーザー名／パスワード
C. 鍵ペア認証
正解: **(**正解を表示します**)**

質問: **118**
以下のコマンドのうち、倉庫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. SELECT MAX(FLAKE_ID) FROM SNOWFLAKE;
C. SELECT COUNT(*) FROM SNOWFLAKE;
D. SELECT COUNT(FLAKE_ID) FROM SNOWFLAKE GROUP BY FLAKE_ID;
正解: **(**正解を表示します**)**
説明
* 倉庫クレジットは、Snowflake内の各仮想倉庫が処理に要する時間に対する支払いに使用されます。
仮想ウェアハウスとは、クエリの実行、データのロード、その他のDML操作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、およびそのサイズに基づいて課金されます[1]。
* 質問に記載されているコマンドのうち、以下のコマンドは倉庫クレジットを使用します。
* SELECT MAX(FLAKE_ID) FROM SNOWFLAKE: このコマンドは、仮想ウェアハウスを必要とするクエリであるため、ウェアハウスクレジットを使用します。このクエリは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、オプションBが正解です。
* SELECT COUNT(*) FROM SNOWFLAKE: このコマンドは倉庫クレジットも使用します
* これは、実行に仮想ウェアハウスを必要とするクエリであるためです。このクエリはSNOWFLAKEテーブルをスキャンし、テーブル3の行数を返します。したがって、オプションCが正解です。
* SELECT COUNT(FLAKE_ID) FROM SNOWFLAKE GROUP BY FLAKE_ID: このコマンドは、仮想ウェアハウスを必要とするクエリであるため、ウェアハウスクレジットも使用します。このクエリは SNOWFLAKE テーブルをスキャンし、FLAKE_ID 列の一意の値ごとに行数を返します。したがって、オプション D が正解です。
* 倉庫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%': このコマンドはメタデータ操作であり、仮想ウェアハウスを必要としないため、ウェアハウスクレジットは使用されません。このコマンドは、現在のデータベースとスキーマ5内でパターン「SNOWFL%」に一致するテーブル名を返します。したがって、オプションAは誤りです。
参考資料: : 計算コストの理解 : MAX 関数 : COUNT 関数 : GROUP BY 句 : SHOW TABLES

質問: **119**
データはJSON形式でインポートされ、VARIANT型の列に保存されています。クエリのパフォーマンスは以前は良好でしたが、最近になってパフォーマンスの低下が報告されています。
何が原因なのでしょうか？
A. 最近のデータインポートにJSONのnull値が含まれていました。
B. 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。
C. 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。

D. JSON内のキーの順序が変更されました。
正解: **(**正解を表示します**)**

質問: **120**
セキュアビューでは、ビューの基となるテーブル内のデータへのアクセスを必要とする内部最適化を利用できません。
A. 偽
B. 真
正解: **(**正解を表示します**)**

質問: **121**
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すればよいでしょうか？
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を示します。
D. FILERONSはファイル形式の場所を示します。FILE_ROW_NUMBERはステージです。
正解: **(**正解を表示します**)**
説明
* ステージとは、Snowflake 内でデータロードおよびアンロード用のファイルを保存できる名前付きの場所です。ステージは、ファイルの保存場所に応じて、内部ステージまたは外部ステージになります。
* 質問にあるクエリは、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、FILE_ROW_NUMBER （ステージ内のファイルの行番号）を含む、さまざまな列を持つテーブルを返します。
したがって、このクエリは、FILEROWS という名前のステージ内のファイルを一覧表示し、そのステージ内の各ファイルの行番号を表示するものと解釈できます。
参考文献：
* :ステージ
* : リスト関数

質問: **122**
Snowpipe REST APIを使用して、データロード履歴のログを保持するにはどうすればよいですか？
A. insertReport を 20 分ごとに呼び出し、最新の 10,000 件のエントリを取得します。
B. 15分間の時間範囲で、10分ごとにloadHistoryScanを呼び出します。
C. 10分間の時間範囲で、8分ごとにinsertReportを呼び出します。
D. 最大時間範囲で、1分ごとにloadHistoryScanを呼び出します。
正解: **(**正解を表示します**)**

質問: **123**
ある企業は、AWS us-east-1リージョンにACCOUNTAという名前のSnowflakeアカウントを保有しています。この企業は、MARKET_DBという名前のSnowflakeデータベースにマーケティングデータを保存しています。同社のビジネスパートナーの1社は、Azure East US 2リージョンにPARTNERBという名前のアカウントを保有しています。マーケティング目的で、同社はMARKET_DBデータベースをパートナーアカウントと共有することに同意しました。
アカウントPARTNERBがMARKET_DBデータベースからデータを取得するために、以下の手順のうちどれを必ず実行する必要がありますか？
A. Azure East US 2 リージョンに新しいアカウント (AZABC123 という名前) を作成します。アカウント ACCOUNTA からデータベース MARKET_DB の共有を作成し、この共有から AWS us-east-1 リージョンにローカルで新しいデータベースを作成し、この新しいデータベースを AZABC123 アカウントにレプリケートします。次に、PARTNERB アカウントとのデータ共有を設定します。
B. アカウントACCOUNTAからデータベースMARKET_DBの共有を作成し、AWS us-east-1リージョンにこの共有から新しいデータベースをローカルに作成します。次に、このデータベースをプロバイダーとして設定し、PARTNERBアカウントと共有します。
C. Azure East US 2 リージョンに新しいアカウント (AZABC123 という名前) を作成します。アカウント ACCOUNTA からデータベース MARKET_DB を AZABC123 にレプリケートし、このアカウントから PARTNERB アカウントへのデータ共有を設定します。
D. データベース MARKET_DB の共有を作成し、AWS us-east-1 リージョンでこの共有から新しいデータベースをローカルに作成します。次に、このデータベースをパートナーのアカウント PARTNERB にレプリケートします。
正解: **C (**コメントを発表する**)**
* Snowflakeは、アカウントレプリケーションと共有レプリケーション機能を使用して、リージョン間およびクラウドプラットフォーム間でのデータ共有をサポートします。アカウントレプリケーションを使用すると、ソースアカウントから同じ組織内の1つ以上のターゲットアカウントにオブジェクトを複製できます。共有レプリケーションを使用すると、ソースアカウントから同じ組織内の1つ以上のターゲットアカウントに共有を複製できます1。
* AWS us-east-1 リージョンの ACCOUNTA アカウントにある MARKET_DB データベースのデータを、Azure East US 2 リージョンの PARTNERB アカウントと共有するには、以下の手順を実行する必要があります。
* Azure East US 2 リージョンに新しいアカウント (AZABC123 という名前) を作成します。このアカウントは、ソース アカウントとターゲット アカウント間のブリッジとして機能します。新しいアカウントは、組織 2 を使用して ACCOUNTA アカウントにリンクする必要があります。
* アカウントレプリケーション機能を使用して、ACCOUNTA アカウントから MARKET_DB データベースを AZABC123 アカウントに複製します。これにより、AZABC123 アカウントに、ACCOUNTA アカウントのプライマリ データベースのレプリカであるセカンダリ データベースが作成されます。
* AZABC123 アカウントから、共有レプリケーション機能を使用して PARTNERB アカウントへのデータ共有を設定します。これにより、AZABC123 アカウントにセカンダリ データベースの共有が作成され、PARTNERB アカウントにアクセス権が付与されます。PARTNERB アカウントは、共有からデータベースを作成し、データをクエリできます。

したがって、選択肢Cが正解です。
参考資料: : リージョンとクラウド プラットフォーム間での共有の複製 : 組織とアカウントの操作 : 複数のアカウント間でのデータベースの複製 : 複数のアカウント間での共有の複製

質問: **124**
アーキテクトは、次のSQLクエリを実行します。

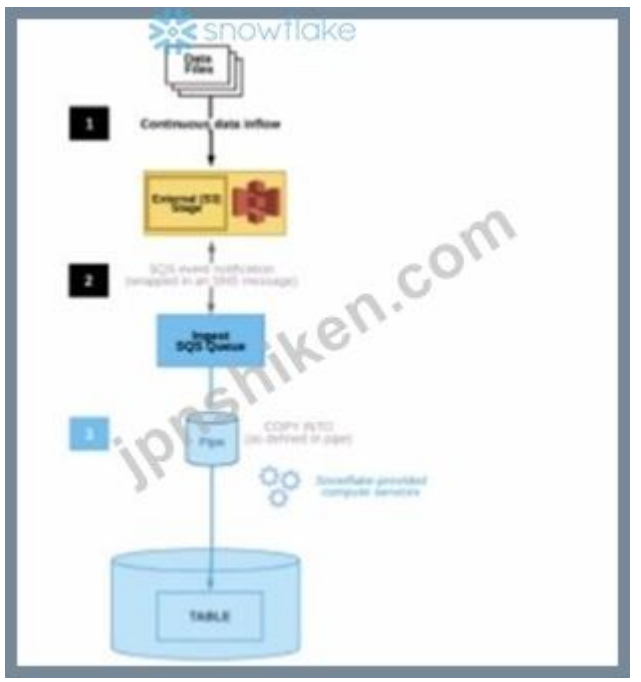
```
SELECT
    METADATA$FILENAME,
    METADATA$FILE_ROW_NUMBER
FROM @FILEROWS/Food_Reviews.csv
        (file_format=CSV_N)
```

このクエリはどのように解釈すればよいでしょうか？
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を示します。FILE_ROW_NUMBERはステージです。
正解: **(**正解を表示します**)**
ステージとは、Snowflake内でデータロードおよびアンロード用のファイルを保存できる名前付きの場所です。ステージは、ファイルの保存場所に応じて、内部ステージまたは外部ステージに分類されます。
質問にあるクエリは、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、FILE_ROW_NUMBER（ステージ内のファイルの行番号）を含む、さまざまな列を持つテーブルを返します。
したがって、このクエリは、FILEROWSという名前のステージ内のファイルを一覧表示し、そのステージ内の各ファイルの行番号を表示するものと解釈できます。
参照：
1：段階
2 :リスト関数

質問: **125**
Snowflakeでは、以下のオブジェクトをクローンできます。
A. 常設テーブル
B. 過渡応答表
C. 一時テーブル
D. 外部テーブル
E. 内部段階
正解: **(**正解を表示します**)**
説明

Snowflakeは、データベース、スキーマ、テーブル、ステージ、ファイル形式、シーケンス、ストリーム、タスク、ロールなど、さまざまなオブジェクトのクローン作成をサポートしています。クローン作成では、データやメタデータをコピーせずに、システム内の既存のオブジェクトのコピーを作成します。クローン作成は、ゼロコピークローンとも呼ばれます。
* 質問に挙げられているオブジェクトのうち、Snowflakeでクローンできるのは以下のものです。
* 永続テーブル: 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テーブルの一種です。永続テーブルは、CREATE TABLE ... を使用してクローンできます。
CLONEコマンド2。したがって、オプションAが正解です。
* 一時テーブル: 一時テーブルは、フェイルセーフ期間を持たないテーブルの一種で、タイムトラベル保持期間は0日または1日のいずれかになります。一時テーブルは、CREATE TABLE ... CLONEコマンド2を使用してクローンすることもできます。したがって、オプションBが正解です。
* 外部テーブル: 外部テーブルは、Amazon S3、Google Cloud Storage、Microsoft Azure Blob Storage などの外部の場所に保存されているデータ ファイルを参照するタイプのテーブルです。外部テーブルは、CREATE EXTERNAL TABLE ... CLONE コマンドを使用してクローンできます。
したがって、選択肢Dが正解です。
* 質問に記載されている以下のオブジェクトは、Snowflake でクローンできません。
* 一時テーブル：一時テーブルは、セッションが終了したとき、または現在のユーザーがログアウトしたときに自動的に削除されるタイプのテーブルです。一時テーブルはクローン作成をサポートしていません。したがって、選択肢Cは誤りです。
* 内部ステージ：内部ステージは、Snowflakeによって管理され、Snowflakeの内部クラウドストレージにファイルを保存するタイプのステージです。内部ステージはクローン作成をサポートしていません。したがって、選択肢Eは誤りです。
参照: : クローン作成に関する考慮事項 : CREATE TABLE ... CLONE : CREATE EXTERNAL TABLE ...
クローン：一時テーブル：内部ステージ

質問**: 126**
この図は、Amazon Simple Notification Service (SNS) を使用した Snowpipe の自動取り込みのプロセスフローを、以下の手順で示しています。
ステップ1 :データファイルがステージにロードされます。
ステップ2 :SNSによって発行されたAmazon S3イベント通知により、Amazon Simple Queue Service (SQS) を介して、ファイルがロード準備完了であることがSnowpipeに通知されます。Snowpipeはファイルをキューにコピーします。
ステップ3 :Snowflakeが提供する仮想ウェアハウスは、指定されたパイプで定義されたパラメータに基づいて、キューに入れられたファイルからターゲットテーブルにデータをロードします。



ステップ2でAWS管理者が誤ってSNSトピックへのSQSサブスクリプションを削除した場合、Amazon S3からイベントメッセージを受信するためにそのトピックを参照するパイプはどうなりますか？

A. Snowflake が同じ SNS トピックへのサブスクリプションを自動的に復元し、パイプ定義で同じ SNS トピック名を指定することでパイプを再作成するため、パイプは引き続きメッセージを受信し続けます。

B. パイプはメッセージを受信できなくなります。SNSトピックの購読が削除されてから24時間経過するまで、ユーザーは待機する必要があります。24時間後にはパイプは既存のSNSトピックへの購読を再利用するため、パイプの再作成は不要です。

C. Snowflakeは新しいSNSトピックを作成することでサブスクリプションを自動的に復元するため、パイプは引き続きメッセージを受信し続けます。その後、Snowflakeはパイプ定義で新しいSNSトピック名を指定することでパイプを再作成します。

D. パイプはメッセージを受信できなくなります。システムをすぐに復旧するには、ユーザーは手動で別の名前の新しいSNSトピックを作成し、パイプ定義で新しいSNSトピック名を指定してパイプを再作成する必要があります。

正解: **D** (コメントを発表する)

AWS 管理者がステップ 2 で SNS トピックへの SQS サブスクリプションを誤って削除した場合、Amazon S3 からイベント メッセージを受信するためにトピックを参照するパイプはメッセージを受信できなくなります。これは、SQS サブスクリプションが SNS トピックと Snowpipe 通知チャネル間のリンクとなっているためです。サブスクリプションがないと、SNS トピックは Snowpipe キューに通知を送信できず、パイプは新しいファイルをロードするようにトリガーされません。システムをすぐに復元するには、ユーザーは別の名前で新しい SNS トピックを手動で作成し、パイプ定義で新しい SNS トピック名を指定してパイプを再作成する必要があります。これにより、新しい通知チャネルとパイプ用の新しい SQS サブスクリプションが作成されます。または、既存の SNS トピックへの SQS サブスクリプションを再作成し、パイプ定義で同じ SNS トピック名を使用するようにパイプを変更することもできます。これにより、通知チャネルとパイプの機能も復元されます。参照:
Amazon S3向けSnowpipeの自動化
Amazon SNSのSnowpipeエラー通知を有効にする
Snowpipeの自動取り込みとAWS S3ステージの設定手順

質問: **127**

検索最適化に必要な権限は、以下のどれですか？

A. 所有権特権がテーブルに置かれている

B. テーブルを含むスキーマに対して検索最適化権限を追加します

C. テーブルに検索最適化権限を追加します。
正解: **A,B (**コメントを発表する**)**

質問: **128**
ある建築家は、Snowflakeの本番環境と品質保証環境を分離するために、2つの別々のSnowflakeアカウントを使用することを選択しました。
QAアカウントは、データおよびデータベースオブジェクトに対する変更を、本番アカウントに反映させる前に実行および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およびデータは、少なくとも毎晩、本番アカウントのデータベースオブジェクト（権限およびデータを含む）と完全に一致するコピーである必要があります。
毎晩、本番環境アカウントのデータとデータベースオブジェクトをQAアカウントに投入するために使用する最も単純な方法はどれですか？
A. 1）各データベースごとに本番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権を共有する
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) 毎晩、それらのデータベースのクローンを作成する
5) クローンしたデータベース上で直接テストを実行する
B. 1）制作アカウントでステージを作成する
2) QAアカウントに、同じ外部オブジェクトストレージの場所を指すステージを作成します。
3) 本番環境アカウントの各テーブルをステージング環境にアンロードするタスクを毎晩実行するように作成します。
4) Snowpipeを使用してQAアカウントにデータを入力する
C. 1）本番アカウントの各データベースでレプリケーションを有効にする
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) クローンしたデータベース上で直接テストを実行する
D. 1）本番環境アカウントで、QA環境アカウントに接続し、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) 外部関数をストアドプロシージャの一部として実行し、本番アカウントの各テーブルをループ処理して、QAアカウントの各テーブルにデータを入力します。
正解: **(**正解を表示します**)**
この方法は、Snowflakeに組み込まれているレプリケーション機能を使用して、本番環境アカウントからQA環境アカウントにデータとデータベースオブジェクトをコピーするため、最もシンプルな方法です。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ことで、QA環境アカウントは元のデータに影響を与えることなく、クローンされた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あるため、データが変更されない限り、追加のストレージ容量を消費しません。このアプローチでは、データ転送プロセスに複雑さやオーバーヘッドをもたらす可能性のある外部ステージ、タスク、Snowpipe、または外部関数は一切必要ありません。
参照：
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データベースを複製する
クローン作成に関する考慮事項

質問: **129**

あなたは組織内のスノーフレーク ·アーキテクトです。ビジネスチームが、Tableauで可視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導入するためにあなたのところに来ました。毎日新しいデータが入ってくるため、古いデータは不要になります。
この場合、コストを最適化するためにどのようなタイプのテーブルを使用しますか？
A. 一時的
B. 一時的
C. 永久
正解: **(**正解を表示します**)**
* 一時テーブルは、Snowflake のテーブルの一種で、フェイルセーフ期間がなく、タイムトラベル保持期間を 0 日または 1 日に設定できます。一時テーブルは、簡単に複製または再現できる一時的または中間的なデータに適しています 1。
* 一時テーブルは、Snowflake のテーブルの一種で、セッションが終了したり、現在のユーザーがログアウトしたりすると自動的に削除されます。一時テーブルはストレージコストを消費しませんが、他のユーザーやセッションからは表示されません。
* パーマネントテーブルは、Snowflake のテーブルの一種で、フェイルセーフ期間と最大 90 日のタイムトラベル保持期間を持ちます。パーマネントテーブルは、偶発的または悪意のある削除から保護する必要のある永続的で耐久性のあるデータに適しています 3。
このユースケースでは、Tableauで視覚化できる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、古いデータは不要になります。そのため、コストを最適化するために最適なテーブルタイプは、一時テーブルです。一時テーブルはフェイルセーフコストを発生せず、タイムトラベル保持期間を0日または1日と短く設定できるためです。このようにして、Tableauでデータをロードしてクエリを実行した後、不要なストレージコストを発生させることなく、データを削除または上書きできます。
参考文献 :: 一時テーブル : テンポラリテーブル : タイムトラベルの理解と使用

質問: **130**
Snowflakeにおける標準的な仮想倉庫ポリシーはどのように機能しますか？
A. システムが、少なくとも 2 分間クラスタをビジー状態にするクエリ負荷があると推定した場合にのみ開始します。
B. 追加のクラスターを起動するのではなく、実行中のクラスターを常にフルロード状態にしておくことでクレジットを節約します。
C. システムが、少なくとも 6 分間クラスタをビジー状態にするクエリ負荷があると推定した場合にのみ開始します。
D. クレジットを節約するのではなく、追加のクラスタを起動することでキューイングを防止または最小限に抑えます。
正解: **(**正解を表示します**)**

質問: **131**
アーキテクトはデータベースとそのすべてのオブジェクト（タスクを含む）を複製します。複製後、タスクは実行を停止します。
なぜこのようなことが起こるのでしょうか？
A. タスクは複製できません。
B. タスクが参照するオブジェクトは完全修飾されていません。
C. クローンされたタスクはデフォルトで一時停止されており、手動で再開する必要があります。
D. アーキテクトには、クローンされたデータベース上のタスクを変更するのに十分な権限がありません。
正解: **C (**コメントを発表する**)**
データベースがクローンされると、タスクを含むすべてのオブジェクトもクローンされます。ただし、クローンされたタスクはデフォルトで一時停止され、ALTER TASK コマンドを使用して手動で再開する必要があります。これは、クローンされたタスクが予期せず実行されたり、元のタスクに干渉したりするのを防ぐためです。したがって、クローン後にタスクの実行が停止する理由は、デフォルトで一時停止されているためです (オプション C)。オプション A、B、および D は正しくありません。タスクはクローンでき、タスクが参照するオブジェクトもクローンさ

れ、完全修飾する必要はありません。また、アーキテクトはクローンされたデータベース上のタスクを変更する必要はなく、再開するだけで済みます。参照: 回答は、Snowflake の Web サイトにあるクローンとタスクに関する公式ドキュメントで確認できます。関連リンクを以下に示します。
オブジェクトのクローン作成 | Snowflake ドキュメント
タスク | Snowflake ドキュメント
タスクの変更 | Snowflake ドキュメント

質問: **132**
ORGADMINロールを持つアーキテクトが、Snowflakeアカウントをエンタープライズエディションからビジネスクリティカルエディションに変更したいと考えています。
これはどのように実現すべきでしょうか？
A. ALTER ACCOUNT コマンドを実行して EDITION タグを作成し、そのタグを Business Critical に設定します。
B. アカウントの ACCOUNTADMIN ロールを使用してエディションを変更します。
C. 同じリージョン内の新しいアカウントにフェイルオーバーし、新しいアカウントの作成時にエディションを指定します。
D. Snowflakeサポートに連絡して、アカウントのエディションを変更するよう依頼してください。
正解: **(**正解を表示します**)**
Snowflakeアカウントのエディションを変更するには、組織管理者（ORGADMIN）はSQLコマンドやSnowflakeインターフェースを介してアカウント設定を直接変更することはできません。適切な手順は、Snowflakeサポートに連絡してアカウントのエディション変更を依頼することです。これにより、変更が正しく管理され、Snowflakeの運用プロトコルに準拠することが保証されます。
参考資料: このプロセスは Snowflake のドキュメントに概説されており、アカウントのエディションの変更は Snowflake サポートを通じて行う必要があると規定されています。1

質問: **133**
外部テーブルに行アクセス ポリシーを適用する際の特徴として、次のうちどれが挙げられますか？
（3選択してください。）
A. 外部テーブルは行アクセス ポリシーを使用して作成でき、そのポリシーは VALUE 列に適用できます。
B. 既存の外部テーブルのVALUE列に行アクセス ポリシーを適用できます。
C. 行アクセス ポリシーを外部テーブルの仮想列に直接追加することはできません。
D. 外部テーブルは、行アクセス ポリシーのマッピング テーブルとしてサポートされています。
E. データベースをクローンする際に、行アクセス ポリシーと外部テーブルの両方がクローンされます。
F. 外部テーブルの上に作成されたビューには、行アクセス ポリシーを適用できません。
正解: **(**正解を表示します**)**
説明
SnowflakeのドキュメントとWeb検索結果によると、以下の3つの記述は正しいです。行アクセスポリシーは、ユーザー定義の条件に基づいて行をフィルタリングできる機能です。行アクセスポリシーは、外部テーブル（ステージ内で外部ファイルからデータを読み取るテーブル）に適用できます。ただし、外部テーブルで行アクセスポリシーを使用する際には、いくつかの制限事項と考慮事項があります。
* CREATE EXTERNAL TABLE ステートメントの WITH ROW ACCESS POLICY 句を使用することで、行アクセス ポリシーを設定した外部テーブルを作成できます。このポリシーは、外部ファイルからの生データを VARIANT データ型で格納する列である VALUE 列に適用できます。

* 行アクセス ポリシーは、ALTER TABLE ステートメントに SET ROW ACCESS POLICY 句を使用することで、既存の外部テーブルの VALUE 列にも適用できます。
* 行アクセス ポリシーは、外部テーブルの仮想列に直接追加することはできません。仮想列とは、VALUE 列から式を使用して派生した列です。行アクセス ポリシーを仮想列に適用するには、VALUE 列にポリシーを適用し、ポリシー定義内で式を繰り返す必要があります。
* 外部テーブルは、行アクセス ポリシーのマッピング テーブルとしてサポートされていません。マッピング テーブルとは、何らかの基準に基づいてユーザーまたはロールのアクセス権限を決定するために使用されるテーブルです。Snowflake では、パフォーマンスの問題やエラーが発生する可能性があるため、外部テーブルをマッピング テーブルとして使用することはサポートされていません。
* Snowflakeはデータベースをクローンする際に、行アクセス ポリシーはクローンしますが、外部テーブルはクローンしません。そのため、クローンされたデータベースのポリシーは、クローンされたデータベースには存在しないテーブルを参照します。この問題を回避するには、クローンされたデータベースで外部テーブルを手動でクローンするか、再作成する必要があります4。
* 行アクセス ポリシーは、外部テーブルの上に作成されたビューに適用できます。ポリシーは、ビュー自体または基となる外部テーブルに適用できます。ただし、ポリシーをビューに適用する場合は、ビューはセキュア ビューである必要があります。セキュア ビューとは、基となるデータとビュー定義を不正なユーザーから隠蔽するビューです5。
参考文献：
* 外部テーブルの作成 | Snowflake ドキュメント
* 外部テーブルの変更 | Snowflake ドキュメント
* 行アクセス ポリシーの理解 | Snowflake ドキュメント
* Snowflakeデータガバナンス：行アクセスポリシーの概要
* セキュアビュー | Snowflakeドキュメント

質問: **134**
データ共有は、同一地域内のプロバイダーアカウントと消費者アカウント間でのみサポートされています。
A. 真
B. 偽
正解: **(**正解を表示します**)**

質問: **135**
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した場合、開発およびテストのシナリオでデータのコピーが必要となり、ゼロコピー クローニングが適さないのはどのようなシナリオでしょうか？ 2 つ選択してください）。
A. 開発者は、ライブデータの変換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. 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、開発者は特定の列がマスクされた本番環境のようなデータを見る必要があります。
C. データは本番環境の 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/テスト用 Snowflake アカウントに開発者に提供する必要があります。
D. 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コピーを、初期開発および単体テスト用に独自に作成します。
E. リリースプロセスでは、本番環境と同等の規模と複雑さを持つデータを用いて、変更内容の事前テストを実施する必要があります。セキュリティ上の理由から、事前テストは本番環境のアカウントでも実行されます。
正解: **B,C (**コメントを発表する**)**
https://docs.snowflake.com/en/user-guide/tag-based-masking-policies#considerations

質問: **136**

どのクエリを使用すれば、特定のワークロードのパフォーマンスを向上させるためにマルチクラスタウェアハウスの恩恵を受ける特定の日と仮想ウェアハウスを特定できますか？

A.

```
SELECT TO_DATE(START_TIME) AS DATE,
WAREHOUSE_NAME,
BYTES_SCANNED,
BYTES_SPILLED
FROM "SNOWFLAKE"."ACCOUNT_USAGE"."QUERY_HISTORY"
HAVING BYTES_SPILLED>BYTES_SCANNED;
```

B.

```
SELECT TO_DATE(START_TIME) AS DATE,
WAREHOUSE_NAME,
SUM(AVG_RUNNING) AS SUM_RUNNING,
SUM(AVG_QUEUED_LOAD) AS SUM_QUEUED
FROM "SNOWFLAKE"."ACCOUNT_USAGE"."WAREHOUSE_LOAD_HISTORY"
GROUP BY 1,2
HAVING SUM(AVG_QUEUED_LOAD) > 0;
```

C.

```
SELECT TO_DATE(START_TIME) AS DATE,
WAREHOUSE_NAME,
BYTES_SCANNED,
BYTES_SPILLED
FROM "SNOWFLAKE"."ACCOUNT_USAGE"."WAREHOUSE_LOAD_HISTORY"
HAVING BYTES_SPILLED>BYTES_SCANNED;
```

D.

```
SELECT TO_DATE(START_TIME) AS DATE,
WAREHOUSE_NAME,
BYTES_SPILLED_TO_LOCAL_STORAGE,
SUM(AVG_QUEUED_LOAD) AS SUM_QUEUED
FROM "SNOWFLAKE"."ACCOUNT_USAGE"."QUERY_HISTORY"
HAVING BYTES_SPILLED_TO_LOCAL_STORAGE >0;
```

正解: **(**正解を表示します**)**

正解はオプションBです。このクエリは、異なる日付と仮想ウェアハウスにおけるキューイング時間（AVG_QUEUED_LOAD）を調べることで、マルチクラスタウェアハウスの必要性を評価するように設計されています。AVG_QUEUED_LOADがゼロより大きい場合、クエリがリソースを待機していることを示しており、マルチクラスタウェアハウスを使用してワークロードをより効率的に処理することでパフォーマンスが向上する可能性があることを示唆しています。日付とウェアハウス名でグループ化し、平均キューイング負荷の合計がゼロより大きいという条件でフィルタリングすることで、ワークロードが利用可能なコンピューティングリソースを超えた特定の日付とウェアハウスを特定します。この情報は、パフォーマンス向上のためにウェアハウスをマルチクラスタ構成にスケールアウトすることを検討する際に役立ちます。

質問: **137**
ある大手製造会社は、傘下の事業部門全体で12個の個別のSnowflakeアカウントを運用している。
同社は、サプライチェーンの最適化を支援し、複数のベンダーとの購買交渉力を高めるために、データ共有のレベルを向上させたいと考えている。
同社のSnowflakeアーキテクトは、各事業部門が共有する情報を決定できると同時に、構成と管理にかかる労力を最小限に抑えるソリューションを設計する必要がある。欧州を拠点とする一部の部門を除き、ほとんどの事業部門は同じクラウド環境でSnowflakeアカウントを使用している。
Snowflakeが推奨するベストプラクティスによると、これらの要件はどのように満たすべきでしょうか？
A. 欧州のアカウントをグローバルリージョンに移行し、接続されたグラフアーキテクチャで共有を管理します。データ交換をデプロイします。
B. 欧州アカウントのデータ共有と組み合わせたプライベートデータ交換を導入する。
C. すべてのセキュアビューでinvoker_share()が使用されていることを確認して、Snowflake Marketplaceにデプロイします。
D. プライベートデータ交換を導入し、レプリケーションを使用して交換内でヨーロッパのデータ共有を可能にします。
正解: **D** (コメントを発表する)
Snowflake が推奨するベストプラクティスによると、大規模製造企業の要件は、欧州アカウントのデータ共有と組み合わせてプライベートデータエクスチェンジを導入することで満たされるはずです。プライベートデータエクスチェンジは、Snowflake Data Cloud プラットフォームの機能で、組織間でデータを安全かつ統制された方法で共有することを可能にします。これにより、Snowflake の顧客は独自のデータハブを作成し、組織の他の部門や外部パートナーを招待してデータセットにアクセスし、提供することができます。プライベートデータエクスチェンジは、エクスチェンジで共有されるデータに対して、一元管理、きめ細かなアクセス制御、およびデータ使用状況のメトリクスを提供します 1。
データ共有は、データをコピーしたり移動したりすることなく、Snowflake アカウント間でデータを安全かつ直接的に共有する方法です。データ共有を使用すると、データプロバイダーは、アカウント内の選択したオブジェクトに対する権限を、他のアカウントの 1 つ以上のデータコンシューマーに付与できます 2。プライベートデータエクスチェンジとデータ共有を組み合わせて使用することで、企業は次のメリットを実現できます。
* 各事業部門は、共有するデータを決定し、プライベートデータ交換プラットフォームに公開できます。公開されたデータは、交換プラットフォームの他のメンバーが検索してアクセスできます。これにより、複数のデータ共有関係や構成を管理する手間と複雑さが軽減されます。
* 企業は、同じクラウド環境にある既存のSnowflakeアカウントを活用してプライベートデータエクスチェンジを作成し、メンバーを招待することができます。これにより、移行とセットアップのコストを最小限に抑え、既存のSnowflakeの機能とセキュリティを活用できます。
* 同社はデータ共有機能を利用して、異なる地域やクラウドプラットフォームにある欧州のアカウントとデータを共有できます。これにより、同社はデータ主権とプライバシーに関する地域および規制上の要件を遵守しつつ、組織全体でのデータ連携を実現できます。
同社はSnowflake Data Cloudプラットフォームを利用して、共有データに対するデータ分析や変換を実行できるほか、他のデータソースやアプリケーションとの統合も可能です。これにより、サプライチェーンを最適化し、複数のベンダーとの購買交渉力を高めることができます。

質問: **138**
水使用量を測定するIoTデバイス用のテーブルが作成されます。このテーブルはすぐに大きくなり、20億行を超えるデータ量になります。

```
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)
)
```

このテーブルに対する一般的なクエリパターンは以下のとおりです。
1. DeviceId、IOT_timestamp、および CustomerId は、select ステートメントのフィルタ述語で頻繁に使用されます。
2. City 列と DeviceManufacturer 列はよく取得されます
3. UniqueId にはよくカウントがあります
クラスタリングキーにはどのフィールドを使用すべきですか？
A. IOT_タイムスタンプ
B. 都市およびデバイスメーカー
C. DeviceId と CustomerId
D. ユニークルド
正解: **(**正解を表示します**)**
クラスタリングキーとは、Snowflakeのストレージ単位であるマイクロパーティション内にデータをまとめて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。クラスタリングは、スキャンする必要のあるデータ量を削減するため、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を向上させることができます。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。この例では、DeviceId、IOT_timestamp、CustomerIdの各列がSELECT文のフィルタ述語で頻繁に使用されているため、クラスタリングキーの候補として適しています。CityとDeviceManufacturerの各列は取得されることは多いものの、フィルタリングの対象にはならないため、クラスタリングキーとしてはそれほど重要ではありません。
UniqueId列はカウントに使用されますが、クラスタリングキーとしては適していません。なぜなら、UniqueId列はカーディナリティが高く、分布が均一である可能性が高く、データの共存に役立たないからです。
したがって、クラスタリングキーとしてDeviceIdとCustomerIdを使用するのが最善の選択肢です。これらはマイクロパーティションの削減とクエリの高速化に役立ちます。参考資料 :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、Snowflakeクラスタリング完全ガイド

質問: **139**
受信共有データに対して許可されている操作は以下のどれですか？
A. テーブルの作成／削除／変更
B. 挿入
C. SELECT WITH JOIN
D. 図表変更
E. GROUP BY句で選択
F. GO
正解: **C,E (**コメントを発表する**)**

質問: **140**
CSVファイル形式で`copy into <table>`コマンドを使用する場合、`match_by_column_name`パラメータはどのように動作しますか？
A. このコマンドを実行すると、ファイルに一致しない列があることを示す警告が表示されます。
B. CSVファイルにはヘッダーが存在することが想定されており、そのヘッダーは大文字小文字を区別するテーブルの列名と照合されます。
C. このパラメータは無視されます。
D. このコマンドはエラーを返します。
正解: **(正解を表示します)**

質問: **141**
ある小売企業は、3000以上の店舗をすべて同じPOS（販売時点情報管理）システムで運用しています。同社は、カテゴリーマネージャーにほぼリアルタイムの売上データを提供したいと考えています。店舗は様々なタイムゾーンにまたがって営業しており、毎分発生する取引件数は店舗によって大きく異なり、売上高も店舗ごとに変動しています。
Sales results are provided in a uniform fashion using data engineered fields that will be calculated in a complex data pipeline. Calculations include exceptions, aggregations, and scoring using external functions interfaced to scoring algorithms. The source data for aggregations has over 100M rows.
Every minute, the POS sends all sales transactions files to a cloud storage location with a naming convention that includes store numbers and timestamps to identify the set of transactions contained in the files. The files are typically less than 10MB in size.
How can the near real-time results be provided to the category managers? (Select TWO).
A. All files should be concatenated before ingestion into Snowflake to avoid micro-ingestion.
B. A Snowpipe should be created and configured with AUTO_INGEST = true. A stream should be created to process INSERTS into a single target table using the stream metadata to inform the store number and timestamps.
C. A stream should be created to accumulate the near real-time data and a task should be created that runs at a frequency that matches the real-time analytics needs.
D. An external scheduler should examine the contents of the cloud storage location and issue SnowSQL commands to process the data at a frequency that matches the real-time analytics needs.
E. The copy into command with a task scheduled to run every second should be used to achieve the near- real time requirement.
正解: **(正解を表示します)**
To provide near real-time sales results to category managers, the Architect can use the following steps:
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 フォーマットと暗号化設定を使用する必要があります。2 外部ステージから Snowflake のターゲット テーブルにファイルをロードする 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、外部ステージに到着した新しいファイルを自動的に検出して取り込むことを意味します。Snowpipe は、重複した取り込みを避けるために、ロード後に外部ステージからファイルを削除するためのコピー オプションも使用する必要があります。3 Snowpipe によって行われた INSERTS をキャプチャするターゲット テーブル上の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、最終更新時刻に関する情報を提供するメタデータ列を含める必要があります。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持期間も設定する必要があります。4 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処理、集計、スコアリングの計算を実行する必要があります。また、クエリは結果を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出力する必要があります。タスクは、1分ごと、または5分ごとなど、リアルタイム分析のニーズに合わせて実行されるようにスケジュールする必要があります。
他の選択肢は、ほぼリアルタイムの結果を提供するには最適でも実現可能でもない。

マイクロインジェストを避けるため、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。ただし、この方法はデータパイプラインに余分な遅延と複雑さをもたらすため、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、ファイルのマージ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への新規ファイルの取り込みが遅延し、データ損失や破損のリスクが高まります。さらに、Snowpipeは連結された各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インジェクションは回避されません。
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、リアルタイム分析のニーズに合った頻度でデータを処理するSnowSQLコマンドを発行する必要があります。ただし、Snowpipeは外部トリガーやスケジューラを必要とせずに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須ではありません。外部スケジューラを使用すると、データパイプラインにオーバーヘッドと依存関係が加わり、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データ取り込みが保証されません。
ほぼリアルタイムの要件を満たすには、1秒ごとに実行されるようにスケジュールされたタスクを含むコピーイントコマンドを使用する必要があります。Snowflakeではタスクを1秒ごとに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現不可能です。タスクの最小間隔は1分ですが、タスクはスケジューリングの遅延や同時実行制限の影響を受けるため、それさえも保証されません。さらに、タスクを含むコピーイントコマンドを使用すると、自動ファイル検出、負荷分散、マイクロパーティション最適化などのSnowpipeの利点を活用できません。参考文献:
1 :SnowPro Advanced :アーキテクト｜学習ガイド
2 :Snowflakeドキュメント｜ステージの作成
3 :Snowflakeドキュメント｜Snowpipeを使用したデータ読み込み
4 :Snowflakeドキュメント｜ストリームとタスクを使用したELT
Snowflakeドキュメント | タスクの作成
Snowflakeドキュメント | データロードのベストプラクティス
Snowflakeドキュメント | Snowpipe REST APIの使用方法
Snowflake ドキュメント | タスクのスケジュール設定
SnowPro 上級者向け :アーキテクト | 学習ガイド
ステージの作成
Snowpipeを使用したデータ読み込み
ELTにおけるストリームとタスクの活用
【タスクの作成】
【データ読み込みに関する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】

質問: **142**
データはJSON形式でインポートされ、VARIANT型の列に保存されています。クエリのパフォーマンスは以前は良好でしたが、最近になってパフォーマンスの低下が報告されています。
何が原因なのでしょうか？
A. 最近のデータインポートにJSONのnull値が含まれていました。
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では、JSON値の文字列の長さにばらつきがありました。
正解: **(**正解を表示します**)**

データはJSON形式でVARIANT型の列にインポートされ、保存されています。クエリのパフォーマンスは以前は良好でしたが、最近になってパフォーマンスの低下が報告されています。これは以下の要因が原因であ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、最も一般的な要素については列のような構造で、残りの要素については 残り物」のような列で、半構造化データを内部的に保存します。JSON内のキーの順序は、Snowflakeが共通要素をどのように判断し、クエリのパフォーマンスをどのように最適化するかに影響します。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て内部ストレージを再編成する必要があり、その結果、クエリのパフォーマンスが低下する可能性があります。
* 最近のデータインポートにおいて、JSON値の文字列長にばらつきが見られました。日付やタイムスタンプなどの非ネイティブ値は、VARIANT型の列にロードされる際に文字列として保存されます。
これらの値に対する操作は、対応するデータ型のリレーショナル列に格納する場合よりも遅くなり、より多くのスペースを消費する可能性があります。
* 最近のデータインポートに含まれる JSON 値の場合、Snowflake はより多くの領域を割り当て、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があります。
その他の選択肢は、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のnull値が含まれていました。Snowflakeは、半構造化データにおいて、SQL NULLとJSON nullの2種類のnull値をサポートしています。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、適切に処理できます。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はありません。
* 最近インポートされたデータには、通常よりもフィールド数が少なくなっています。Snowflakeは、スキーマやフィールドが異なる半構造化データを処理できます。最近のデータインポートでフィールド数が通常より少なくても、Snowflakeは既存のフィールドに基づいてデータ取り込みとクエリ実行を最適化できるため、クエリのパフォーマンスに大きな影響はありません。
参考文献：
* VARIANTに格納される半構造化データに関する考慮事項
* Snowflake Architect トレーニング
* バリアント列内の一意要素に対するSnowflakeクエリのパフォーマンス
* Snowflakeバリアントのパフォーマンス

質問: **143**
Snowpipe REST APIを使用して、データロード履歴のログを保持するにはどうすればよいですか？
A. insertReport を 20 分ごとに呼び出し、最新の 10,000 件のエントリを取得します。
B. 最大時間範囲で、1分ごとにloadHistoryScanを呼び出します。
C. 10分間の時間範囲で、8分ごとにinsertReportを呼び出します。
D. 15分間の範囲で、10分ごとにloadHistoryScanを呼び出します。
正解: **(**正解を表示します**)**
Snowpipe REST API は、データロード履歴を取得するためのエンドポイントとして insertReport と loadHistoryScan の 2 つを提供します。insertReport エンドポイントは insertFiles エンドポイントに送信されたファイルのステータスを返し、loadHistoryScan エンドポイントは Snowpipe によって実際にテーブルにロードされたファイルの履歴を返します。データロード履歴のログを保持するには、データ取り込みプロセスに関するより正確で完全な情報を提供する loadHistoryScan エンドポイントを使用することをお勧めします。loadHistoryScan エンドポイントは、開始時刻と終了時刻をパラメータとして受け取り、その時間範囲内でロードされたファイルを返します。指定できる最大時間範囲は 15 分で、返されるファイルの最大数は 10,000 です。したがって、データロード履歴のログを保持するには、15 分間隔で 10 分ごとに

loadHistoryScan エンドポイントを呼び出し、結果をログファイルまたはテーブルに保存するのが最善の方法です。この方法であれば、ログにはSnowpipeによってロードされたすべてのファイルが記録され、時間範囲の欠落や重複が回避されます。他のオプションは以下の理由で誤りです。
* 20分ごとにinsertReportを呼び出して直近10,000件のエントリを取得しても、Snowpipeの非同期処理の性質上、一部のファイルが欠落したり重複したりする可能性があるため、データロード履歴の完全なログは得られません。さらに、insertReportは送信されたファイルのステータスのみを返し、ロードされたファイルのステータスは返しません。
* 最大時間範囲で毎分 loadHistoryScan を呼び出すと、API 呼び出しが多すぎることになります
* また、同じファイルが複数回返されるため、不要なオーバーヘッドが発生します。さらに、最大時間範囲は1分ではなく15分です。
* 10分間の時間範囲で8分ごとにinsertReportを呼び出すと、オプションAと同じ問題が発生し、時間範囲にギャップや重複も発生します。
参考文献：
* Snowpipe REST API
* オプション 1: Snowpipe REST API を使用したデータの読み込み
* パイプ使用履歴

質問**: 144**

図のアーキテクチャに基づいて、DB1のデータをTBL2にコピーするにはどうすればよいでしょうか？ 2つ選択してください)。

A.
```
use database DB1;
use schema SH1;
copy into DB1.SH2.TBL2
  from @STAGE1
 file_format = (format_name = FF_PIPE_1);
```

B.
```
use database DB1;
use schema SH2;
copy into TBL2
  from @STAGE1
 file_format = (format_name = FF_PIPE_1);
```

C.
```
use database DB1;
use schema SH2;
copy into DB1.SH2.TBL2
  from @STAGE1
 file_format = (format_name = DB1.SH1.FF_PIPE_1);
```

D.
```
use database DB1;
use schema SH2;
copy into TBL2
  from @DB1.SH1.STAGE1
 file_format = (format_name = FF_PIPE_1);
```

E.
```
use database DB1;
use schema SH2;
copy into DB1.SH2.TBL2
  from @DB1.SH1.STAGE1
 file_format = (format_name = DB1.SH1.FF_PIPE_1);
```

正解: (正解を表示します)

* 図のアーキテクチャは、DB1とDB2という2つのデータベースと、SH1とSH2という2つのスキーマを持つSnowflakeデータプラットフォームを示しています。DB1にはテーブルTBL1とステージSTAGE1が含まれています。DB2にはテーブルTBL2が含まれています。また、図には、ファイル形式FF PIPE 1を使用してSTAGE1からTBL2にデータをコピーするSQL言語で記述されたコードスニペットも示されています。

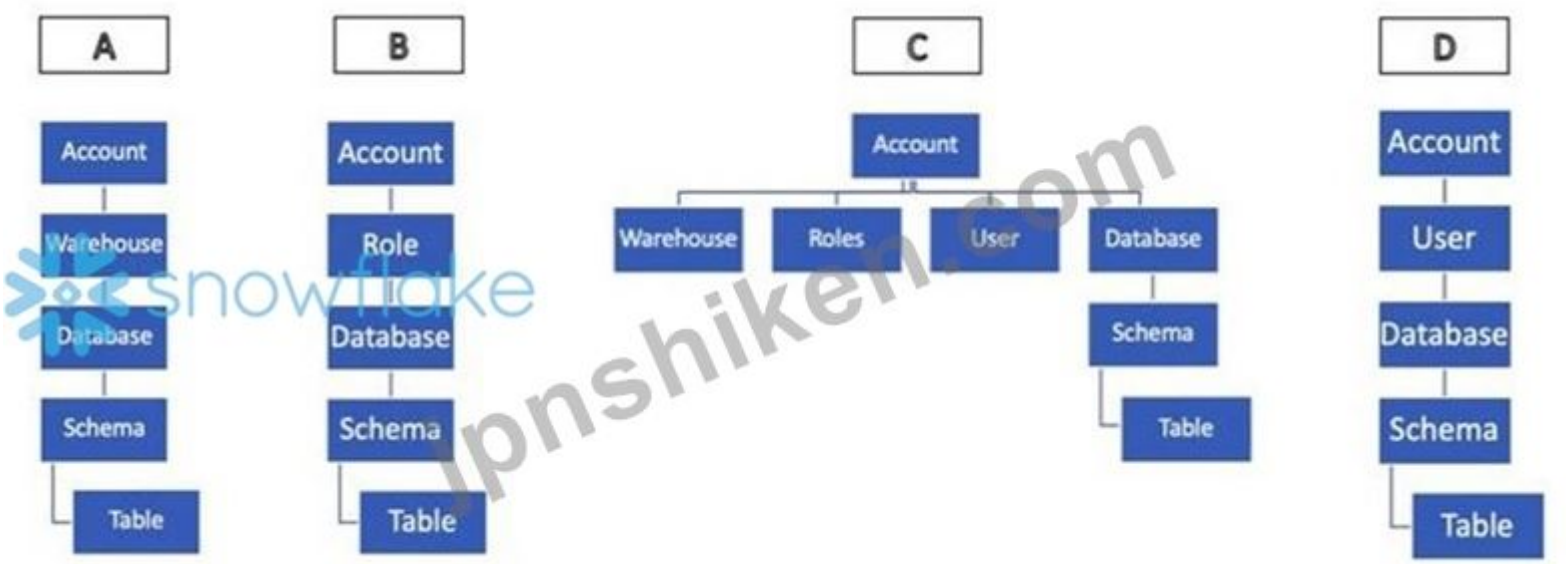
* DB1からTBL2にデータをコピーするには、提示された選択肢の中から2つの方法があります。

* オプション B: STAGE1 を参照する名前付き外部ステージを使用します。このオプションでは、DB1.SH1 の STAGE1 と同じ場所を指す外部ステージ オブジェクトを DB2.SH2 に作成する必要があります。外部ステージは、URL パラメーターで STAGE11 の場所を指定して CREATE STAGE コマンドを使用して作成できます。例:
SQLAIが生成したコードです。内容をよく確認の上、ご使用ください。詳細はFAQをご覧ください。
データベースDB2を使用する。
スキーマSH2を使用する。
ステージEXT_STAGE1を作成します
url = @DB1.SH1.STAGE1;
* 次に、COPY INTOコマンドを使用して、外部ステージ名をFROMパラメータで、ファイルフォーマット名をFILE FORMATパラメータで指定して、外部ステージからTBL2にデータをコピーできます。例：
SQLAIが生成したコードです。内容をよく確認の上、ご使用ください。詳細はFAQをご覧ください。
TBL2にコピーする
@EXT_STAGE1 より
ファイル形式 = (形式名 = DB1.SH1.FF PIPE 1);
* オプション E: クロス データベース クエリを使用して、TBL1 からデータを選択し、TBL2 に挿入します。このオプションでは、INSERT INTO コマンドと SELECT 句を使用して、DB1.SH1 の TBL1 からデータをクエリし、DB2.SH2 の TBL2 に挿入する必要があります。クエリでは、データベース名とスキーマ名を含む、テーブルの完全修飾名を使用する必要があります。例:
SQLAIが生成したコードです。内容をよく確認の上、ご使用ください。詳細はFAQをご覧ください。
データベースDB2を使用する。
スキーマSH2を使用する。
TBL2に挿入する
DB1.SH1.TBL1 から * を選択します。
* 他の選択肢は無効です。理由は以下のとおりです。
* オプション A: COPY INTO コマンドに無効な構文が使用されています。FROM パラメーターにはテーブル名を指定することはできません。ステージ名またはファイルロケーションのみを指定できます。
* オプション C: COPY INTO コマンドに無効な構文が使用されています。FILE FORMAT パラメーターではステージ名を指定できず、ファイル形式名またはオプション 2 のみを指定できます。
* オプション D: CREATE STAGE コマンドに無効な構文が使用されています。URL パラメータではテーブル名を指定できず、ファイルロケーションのみを指定できます。
参考文献：
* 1: ステージの作成 | Snowflake ドキュメント
* 2: テーブルへのコピー | Snowflake ドキュメント
* 3: クロスデータベースクエリ | Snowflakeドキュメント

質問**: 145**
Snowflakeアーキテクトが、Snowflakeデータクラウド上の組織向けにマルチテナントアプリケーション戦略を設計しており、テナントごとにアカウントを割り当てる戦略の採用を検討している。
このアプローチでは、どの要件が満たされますか？ 2つ選択してください。）
A. テナントのデータ形状はテナントごとに固有の場合があります。

B. テナントあたりのオブジェクト数を減らす必要があります。
C. セキュリティおよびロールベースアクセス制御（RBAC）ポリシーは、簡単に設定できる必要があります。
D. 保管コストを最適化する必要があります。
E. 計算コストを最適化する必要があります。
正解: **(**正解を表示します**)**

質問: **146**
下記から正しい階層を選択してください



A. オプションD
B. オプションB
C. オプションC
D. オプションA
正解: **(**正解を表示します**)**

質問: **147**
Snowflake Business Criticalエディションのデータプロバイダーアカウントが、Enterpriseエディションのデータコンシューマーアカウントとデータを共有することは可能ですか？
A. ビジネスクリティカルアカウントは、エンタープライズコンシューマーへのデータ共有プロバイダーになることはできません。コンシューマーアカウントもすべてビジネスクリティカルである必要があります。
B. プロバイダー アカウントのユーザーが共有の作成または変更権限を持つ場合、エンタープライズ アカウントをコンシューマーとして追加すると、共有をインポートできます。
C. プロバイダー アカウントの共有所有ロールを持つユーザーが、エンタープライズ コンシューマー アカウントを追加する際に share_restrictions を False に設定すると、共有をインポートできます。
D. プロバイダー アカウントのユーザーが共有所有ロールを持ち、かつ共有制限のオーバーライド権限 share_restrictions が False に設定されている場合、エンタープライズ コンシューマー アカウントを追加することで共有をインポートできます。
正解: **(**正解を表示します**)**

Snowflake Business Critical (BC) エディションのアカウントがデータを共有する場合、BC によって保証されるより高いレベルのコンプライアンスとセキュリティを維持するために設計されたデータ共有制限に従う必要があります。
デフォルトでは、BCアカウントは他のBC（またはそれ以上の）アカウントとのみデータを共有し、一貫したセキュリティとコンプライアンス（HIPAA、HITRUST、FedRAMPなど）を維持します。
ただし、適切な権限を持つユーザーが明示的に制限を無効にした場合は、例外が認められる場合があります。
重要な概念：共有制限
* Snowflakeは、BCアカウントに対してデフォルトでデータ共有制限を適用します。
* OVERRIDE SHARE RESTRICTIONS グローバル権限を持つロールは、対象アカウントを追加する際に share_restrictions = FALSE を設定することでこれを回避できます。
これは正しい。理由は以下のとおりである。
* ユーザーは、OVERRIDE SHARE RESTRICTIONS権限を持つロールを持っている必要があります。
* そのユーザーは、エンタープライズ版の消費者アカウントを追加する際に、share_restrictions = FALSE に設定できます。
公式文書抜粋：
「データプロバイダーがビジネスクリティカル（またはそれ以上）のアカウントである場合、Snowflake はデフォルトで、他のビジネスクリティカル（またはそれ以上）のアカウントとのみデータを共有できる制限を適用します。グローバル権限 OVERRIDE SHARE RESTRICTIONS を持つロールを持つプロバイダーアカウントのユーザーは、コンシューマーアカウントを追加する際に SHARE_RESTRICTIONS = FALSE を明示的に設定することで、この制限を上書きできます。」出典: Snowflake ドキュメント - CREATE SHARE 他のオプションが正しくない理由:
A：間違いです。これは絶対的なものではありません。ビジネスクリティカルアカウントは、制限が明示的に解除されていれば、エンタープライズアカウントとデータを共有できます。
B：不正解です。共有を作成または変更する権限を持っているだけでは不十分です。共有制限を上書きする権限を持ち、制限を明示的に設定する必要があります。
C: 誤りです。share_restrictions = FALSE の設定が必要ですが、オーバーライドする権限もロールに付与されている必要があります。権限がない場合、操作は失敗します。
参考文献：
CREATE SHARE - SHARE_RESTRICTIONS パラメータ
Snowflake Editionsの比較
株式およびデータ共有制限の管理

質問: **148**
セキュリティ、ガバナンス、データ保護機能のうち、最低限SnowflakeのBusiness Criticalエディションを必要とするものはどれですか？（2つ選択してください。）
A. 暗号化データの定期的な再鍵生成
B. AWS、Azure、またはGoogle CloudからSnowflakeへのプライベート接続
C. 延長時間旅行（最大90日間）
D. Tri-Secret Secureによる顧客管理型暗号化キー
E. フェデレーション認証とSSO
正解: **(**正解を表示します**)**

質問: **149**

ある会社には、データ破損を含む「Data」という名前のテーブルがあります。会社は、クローン技術とタイムトラベルを用いて、5分前のデータ状態に復元したいと考えています。
これを実現するには、どのコマンドを使えばよいでしょうか？
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 Data AT(OFFSET => -60*5);
D. CREATE TABLE Recover Data CLONE Data AT(TIME => -60*5);
正解: **(**正解を表示します**)**
これは、クローニングとタイムトラベルを使用して、5分前の状態のテーブル Data のクローンを作成するための正しいコマンドです。クローニングとは、データやメタデータを複製することなく、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。タイムトラベルとは、定義された期間内の任意の時点の履歴データ（変更または削除されたデータ）にアクセスできる機能です。過去の特定の時点のテーブルのクローンを作成するには、次の構文を使用します。
CREATE TABLE <clone_name> CLONE <source_table> AT (OFFSET => <offset_in_seconds>); OFFSET パラメータは、現在時刻からの秒単位の時間差を指定します。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、指定された時点のソース テーブルに存在していたデータが含まれます 12。
参照：
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：過去の特定の時点におけるオブジェクトのクローン作成

質問: 150
テーブルには5つの列があり、数百万件のレコードが含まれています。列のカーディナリティ分布は以下のとおりです。

Column	Number of Distinct Values
C1	10,790
C2	108
C3	302,605
C4	1.117,736
C5	2.205,400

列C4とC5は、主にSELECTクエリのGROUP BY句とORDER BY句で使用されます。
一方、C1、C2、C3列は、SELECTクエリのフィルタ条件や結合条件で頻繁に使用されます。
アーキテクトは、クエリのパフォーマンスを向上させるために、このテーブルのクラスタリングキーを設計する必要があります。
Snowflakeの推奨事項に基づくと、複数列のクラスタリングキーを定義する際に、クラスタリングキーの列はどのように順序付けるべきでしょうか？
A. C5、C4、C2
B. C3、C4、C5
C. C1、C3、C2

D. C2、C1、C3
正解: **(**正解を表示します**)**
説明
Snowflakeのドキュメントによると、テーブル1のクラスタリングを選択する際の考慮事項は以下のとおりです。
* クラスタリングが最適なのは、以下のいずれかの条件を満たす場合です。
費用に関係なく、可能な限り最速の応答時間を求めている。
* クエリのパフォーマンスが向上したことで、テーブルのクラスタリングとメンテナンスに必要なクレジットが相殺されます。
* クラスタリングは、クラスタリングキーが以下のタイプのクエリ述語で使用される場合に最も効果的です。
* フィルタ述語（例WHERE句）
* 述語を結合する（例ON節）
* グループ化述語（例GROUP BY句）
* ソート述語（例ORDER BY句）
* クラスタリングキーが上記のクエリ述語のいずれにも使用されていない場合、またはクラスタリングキーが、キーに関数または式を適用する必要がある述語 (DATE_TRUNC、TO_CHAR など) で使用されている場合、クラスタリングの効果は低下します。
* Snowflakeでは、ほとんどのテーブルにおいて、キーごとに最大3つまたは4つの列 または式）を推奨しています。
3〜4列以上追加すると、メリットよりもコストの増加の方が大きくなる傾向があります。
これらの考察に基づくと、クラスタリングキー列として最適な選択肢はCです。C1、C3、C2。理由は以下のとおりです。
* これらの列は、クラスタリングに最も効果的な述語である SELECT クエリのフィルタ条件と結合条件で頻繁に使用されます。
* これらの列はカーディナリティが高く、つまり多くの異なる値を持つため、クラスタリングの偏りを軽減し、圧縮率を向上させるのに役立ちます。
* これらの列は互いに相関関係にある可能性が高く、そのため、類似の行を同じマイクロパーティションにまとめて配置し、スキャン効率を向上させるのに役立ちます。
* これらの列には関数や式を適用する必要がないため、クラスタリングに影響を与えることなく述語で直接使用できます。
参考文献: 1: テーブルのクラスタリングを選択する際の考慮事項 | Snowflake ドキュメント

質問**: 151**
在庫テーブルがあります。このテーブルに2つのビューを作成しました。ビューは以下のようになります。
ビュー NON_SECURE_INVENTORY を作成します
書誌番号、タイトル、著者、ISBNを選択
在庫から
WHERE BIBNUMBER IN(511784,511805,511988,512044,512052,512063);
CREATE SECURE VIEW SECURE_INVENTORY AS
書誌番号、タイトル、著者、ISBNを選択
在庫から
WHERE BIBNUMBER IN(511784,511805,511988,512044,512052,512063);
以下のクエリを実行しました
ALTER SESSION SET USE_CACHED_RESULT=FALSE; -- クエリ キャッシュから取得しないようにするため
SELECT * FROM NON_SECURE_INVENTORY WHERE BIBNUMBER =511784; SELECT * FROM SECURE_INVENTORY WHERE BIBNUMBER =511784;
最初のクエリのクエリプロファイルは以下のようになります。

しかし、2番目のクエリプロファイルは以下のようになります。

どちらのビューも同じ基となるビューの同じ列を使用しています。それなのに、なぜクエリプロファイルにこのような違いが生じるのでしょうか。

A. ビューへのアクセス権を持たないロールによってクエリが実行されました

B. クエリプロファイルが破損しています

C. セキュアビューでは、基となるテーブルやビューの内部構造の詳細が公開されません。

正解: **(**正解を表示します**)**

質問: **152**
ロールが共有フォルダを所有していないが、共有フォルダ内のオブジェクトを所有している場合、オブジェクトへのアクセスをブロックするにはどうすればよいでしょうか？
A. 株主と連絡を取る必要があります
B. 共有所有者からオブジェクトに対するUSAGEまたはSELECT権限をCASCADEで取り消します。
C. 共有所有者からオブジェクトに対するUSAGEまたはSELECT権限を取り消す
正解: **(**正解を表示します**)**

質問: **153**
CSVファイル形式で`copy into <table>`コマンドを使用する場合、`match_by_column_name`パラメータはどのように動作しますか？
A. CSVファイルにはヘッダーが存在することが想定されており、そのヘッダーは大文字小文字を区別するテーブルの列名と照合されます。
B. このパラメータは無視されます。
C. コマンドはエラーを返します。
D. このコマンドを実行すると、ファイルに一致しない列があることを示す警告が表示されます。
正解: **(**正解を表示します**)**
https://community.snowflake.com/s/article/COPY-INTO-table-Command-MATCH-BY-COLUMN-NAME- Copy-Option-Returns-Error-When-Loading-CSV-Data

質問: **154**
Snowflake SQL REST APIを使用する際に許可される操作はどれですか？ 2つ選択してください。）
A. ROLLBACKコマンドの使用
B. PUTコマンドの使用方法
C. GETコマンドの使用
D. テーブルを返すストアドプロシージャへのCALLコマンドの使用
E. 1回の呼び出しで複数のSQLステートメントを送信する
正解: **(**正解を表示します**)**

質問: **155**
ある企業は、さまざまなIoT操作用のJSONレコードを提供するソースシステムを持っています。JSONは、バリアント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、パフォーマンスが問題になりつつあります。バリアントフィールド内のcreate_dateキーでフィルタリングするために、汎用的なアクセスパターンが使用されています。
パフォーマンスを向上させるにはどうすればよいでしょうか？
A. 対象テーブルを変更して、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します。これにより、データ型がタイムスタンプのcreate_dateフィールドが追加されます。このフィールドがフィルタで使用されると、パーティションプルーニングが実行されます。

B. JSONレコードから取得した追加フィールドを対象テーブルに追加します。これには、データ型がvarcharの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で使用されると、パーティションプルーニングが実行されます。
C. 使用するデータウェアハウスのサイズを検証します。レコード数が数億件に近づく場合、この量のデータを処理するために必要な最小サイズはXL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取り入れます。ユーザーまたはプロセスが特定の日付範囲を照会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るようにしてください。
正解: **A** (コメントを発表する)
正解はAです。スキャンおよび処理されるデータ量を削減することで、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リングし、フィルタ条件に一致しないマイクロパーティションを削除できます。これにより、JSONデータを解析してレコードごとにvariant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を向上させないため、誤りです。varchar データ型の create_date フィールドを追加しても、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できず、
フィルタ条件に一致しないマイクロパーティションを削除します。ただし、これにはJSONデータの解析と、各レコードのバリアントフィールドへのアクセスが必要です。
* 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、誤りです。Snowflakeは、使用するデータウェアハウスのサイズを検証することで、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。しかし、これによってスキャンおよび処理されるデータ量が削減されるわけではなく、これがJSONデータに対するクエリの主なボトルネックとなっています。
* オプション D は、データ読み込みおよびクエリ処理に不要な複雑さとオーバーヘッドを追加するため、誤りです。日付範囲でパーティション分割された複数のテーブルを使用することで、Snowflake は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量を削減できます。ただし、これには複数のテーブルの作成と維持、日付に基づいて適切なテーブルへのデータの読み込み、および複数の日付範囲にまたがるクエリのためのテーブル結合が必要です。参照:
* 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、COPY INTO コマンドの構文と使用方法についても説明します。このコマンドは、ON_ERROR、PURGE、SKIP_FILE など、ロード動作を制御するためのさまざまなオプションとパラメーターをサポートしています。
* 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扱う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、セッションのタイムゾーンとシステムの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 内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、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、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。

質問: **156**

Snowflakeのアーキテクトが、複数アカウントに対応した設計戦略を策定している。
この戦略は、どのシナリオにおいて最も費用対効果が高いでしょうか？ 2つ選択してください。）
A. 会社は、AWS 上に存在する本番データベースを、Azure 上に存在する開発データベースにクローンしたいと考えています。
B. 会社は2つのデータベース間でデータを共有する必要がありますが、一方のデータベースはペイメントカード業界データセキュリティ基準（PCI DSS）に準拠している必要がありますが、もう一方はそうではありません。
C. 会社は、開発、テスト、および本番環境向けに、異なる役割ベースのアクセス制御機能をサポートする必要があります。
D. 会社のセキュリティポリシーでは、開発環境、テスト環境、本番環境で異なるActive Directoryインスタンスを使用することが義務付けられています。
E. 会社は、特定のユーザーに対して特定のネットワーク ポリシーを使用して、特定の IP アドレスを許可およびブロックする必要があります。
正解: **(**正解を表示します**)**
マルチアカウント設計戦略とは、クラウドプロバイダー、リージョン、環境、ビジネスユニットなど、さまざまな基準に基づいてSnowflakeアカウントを論理的なグループに整理する方法です。マルチアカウント設計戦略は、コスト最適化、パフォーマンス分離、セキュリティコンプライアンス、データ共有1など、さまざまな目標の達成に役立ちます。この質問では、マルチアカウント設計戦略が最もコスト効率の良いシナリオは次のとおりです。
* 同社は、AWS 上に存在する本番データベースを、Azure 上に存在する開発データベースにクローンしたいと考えています。このシナリオでは、Snowflake のクロスクラウドレプリケーション機能を活用できるため、マルチアカウント設計戦略が有効です。この機能により、異なるクラウドプラットフォームやリージョン間でデータベースを複製することが可能になります。この機能は、データ転送コストとレイテンシを削減し、高可用性と災害復旧を実現するのに役立ちます。2
* 会社のセキュリティ ポリシーでは、開発、テスト、本番環境でそれぞれ異なる Active Directory インスタンスを使用することが義務付けられています。このシナリオでは、複数アカウント設計戦略を採用することで、各環境で異なるフェデレーション認証方式を使用し、それらを異なる Active Directory インスタンスと統合できるため、メリットがあります。これにより、Snowflake アカウントへのアクセスに関するセキュリティとガバナンスが向上し、ユーザー管理とプロビジョニングも簡素化されます。3
他のシナリオでは、複数アカウント設計戦略において最も費用対効果の高い選択肢とはならない。理由は以下のとおりである。
* 企業は、2 つのデータベース間でデータを共有する必要があります。一方のデータベースはペイメント カード業界データ セキュリティ基準 (PCI DSS) に準拠する必要がありますが、もう一方は準拠する必要はありません。このようなシナリオは、セキュア ビューとセキュア UDF を使用して機密データをマスクまたはフィルタリングし、データにアクセスするユーザーに適切な役割と権限を適用することで、単一の Snowflake アカウント内で処理できます。これにより、複数のアカウントを管理する追加コストをかけずに PCI DSS 準拠を実現できます。4
* 開発環境、テスト環境、本番環境それぞれにおいて、異なるロールベースのアクセス制御機能をサポートする必要があります。このシナリオは、Snowflakeのネイティブなロールベースアクセス制御（RBAC）機能（ロール、権限付与、特権など）を使用して、各環境ごとに異なるアクセスレベルとアクセス許可を定義することで、単一のSnowflakeアカウント内でも対応できます。これにより、データとオブジェクトのセキュリティと整合性を確保し、ユーザー間の職務と責任の分離を徹底することができます。
* 企業は、特定のユーザーに対して、特定のIPアドレスを許可またはブロックするための特定のネットワークポリシーを使用する必要があります。
このシナリオは、ネットワークポリシーを使用することで、単一の Snowflake アカウント内でも処理できます。
* Snowflakeの機能の一つで、ネットワークポリシーを作成・適用することで、SnowflakeアカウントにアクセスできるIPアドレスを制限できます。これにより、不正アクセスを防止し、悪意のある攻撃からデータを保護できます。
参考文献：
* スノーフレークトポロジーの設計
* クロスクラウドレプリケーション

* フェデレーション認証とSSOの設定
* セキュアビューとセキュアUDFを使用してPCI DSSに準拠する
* [Snowflakeにおけるアクセス制御の理解]
* [ネットワークポリシー]