

## Esri.EGMP2201.v2025-02-24.q24

試験コード:	EGMP2201
試験名称:	Enterprise Geodata Management Professional 2201
認定資格:	Esri
無料問題数:	24
バージョン:	v2025-02-24
アクセス数:	316
ページビュー数:	240
<a href="https://www.jpnpdf.com/Esri.EGMP2201.v2025-02-24.q24-mondaishu.html">https://www.jpnpdf.com/Esri.EGMP2201.v2025-02-24.q24-mondaishu.html</a>	

### 最新問題: 1

井戸フィーチャクラスには、井戸ごとに1つの行があります。well\_inspections テーブルには、井戸が検査された時間ごとに1つの行があります。すべての検査日は、マップ上の各井戸の周りに集められたラベルとして表示する必要があります。

この要件を満たすにはどのような種類の関連付けを使用する必要がありますか？

- A. 参加する
- B. 関連付ける
- C. 関係クラス

**Answer: B (メッセージを残す)**

シナリオの概要:

井戸フィーチャクラスには、井戸ごとに1つの行があります。

well\_inspections テーブルには、井戸の検査ごとに1つの行があります。

well\_inspections テーブルの検査日付は、マップ上の各井戸の周りに集められたラベルとして表示する必要があります。

目標は、データが動的に表示されるため（検査日が井戸の周りに集まっている）、これら2つのデータセットを永続的に結合することなく、それらの間の接続を確立することです。

ジオデータベース内の関連:

リレートは、テーブルが共通のキーフィールドによってリンクされているものの、個別のままであるテーブル関連付けの一種です。

リレートを使用すると、データを複製したり永続的に関連付けたりすることなく、動的なクエリで関連レコードを取得できます。

リレートを使用すると、特定の井戸のすべての検査日を動的に照会し、それらをマップ上にラベルとして表示し、井戸と検査データセットの両方の整合性を維持できます。

(ArcGIS ドキュメント: 関連)

代替オプション:

オプションA: 参加する

結合は、共有キーに基づいて2つのテーブルを1つの仮想テーブルにマージします。ただし、このアプローチは静的であり、更新のたびにテーブルを再結合する必要があるため、動的にクラスター化されたラベルを表示するには適していません。

オプション C: 関係クラス

リレーションシップクラスは、2つのデータセット間のルールを適用する、より永続的な関連付けです。データ間の関係を維持するのに最適ですが、マップ上で検査日を動的にラベル付けするには必要ありません。

したがって、このシナリオでは、リレートが最も効率的で適切なオプションです。

## 最新問題: 2

GIS アナリストがデータベース ビューを作成します。データベース ビューがマップに読み込まれると、パフォーマンスが最適になりません。

アナリストはどのワークフローを使用すべきでしょうか？

- A. シェープファイルにエクスポート
- B. 別のフィーチャクラスに結合する
- C. ジオデータベースに登録する

**Answer: C (メッセージを残す)**

データベース ビューがマップ内でパフォーマンスが低い場合は、ジオデータベースに登録するとパフォーマンスが大幅に向上します。

1. ジオデータベースに登録する理由

- \* データベース ビューをジオデータベースに登録すると、データベース ビューがジオデータベース システムに統合され、次のことが可能になります。
- \* 空間インデックスの適切な使用。
- \* ジオデータベースによるクエリの最適化。
- \* ジオデータベース固有のパフォーマンス強化を活用して、大規模なデータセットの処理を改善しました。

2. 他の選択肢はないのでしょうか？

- \* シェイプファイルへのエクスポート:
- \* シェープファイルは、インデックス作成とパフォーマンスの最適化が欠けているフラット ファイル形式です。シェープファイルの使用は、複雑なクエリや大規模なデータセットには適していません。
- \* 別のフィーチャクラスへの結合:
- \* データを結合しても、ビューのパフォーマンス問題の根本的な原因は解決されず、結合によって複雑さが増すとパフォーマンスがさらに低下する可能性があります。

ジオデータベースにデータベース ビューに登録する手順:

- \* ArcGIS Pro を開き、データベースに接続します。
- \* データベース ビューを指定して、[ジオデータベースに登録] ツールを使用してください。
- \* ビューに空間データが含まれている場合は、空間インデックスを含めるように登録を構成します。
- \* 登録したビューをマップに保存してパフォーマンスをテストします。

追加メモ:

- \* データベース ビューに複雑な SQL ロジックが含まれている場合は、可能であればクエリを簡素化してパフォーマンスをさらに向上させます。
- \* ビュー内の基になるテーブルに適切なインデックスが付けられていることを確認します。

Esri ドキュメントおよび学習リソースからの参照:

- \* ジオデータベースへのビューの登録
- \* ビューによるマップパフォーマンスの向上

結論:

データベース ビューをジオデータベースに登録すると、空間インデックス作成とジオデータベースの最適化が可能になり、パフォーマンスの問題に対処し、効率的なレンダリングとクエリが保証されます。

最新問題: 3

組織では、モバイル アプリと Web アプリでフィーチャクラスを編集できるようにする必要があります。データベース管理者は、一般的なクエリのパフォーマンスを向上させるために、フィーチャクラスにインデックスを追加する必要があります。

これらのフィーチャクラスはどこに保存すればよいのでしょうか?

- A. エンタープライズ ジオデータベース
- B. ファイルジオデータベース
- C. モバイルジオデータベース

**Answer: A (メッセージを残す)**

シナリオを理解する:

\* 組織では、フィーチャクラスがモバイル アプリと Web アプリで編集可能であることが必要です。

\* 一般的なクエリでは、インデックスによるパフォーマンスの最適化も必要です。

フィーチャクラスのストレージ オプション:

\* エンタープライズ ジオデータベース:

\* マルチユーザー環境をサポートしているため、モバイルや Web アプリの編集に適しています。

\* データベース レベルでのインデックス作成が可能になり、大規模なデータセットのクエリ パフォーマンスが向上します。

\* モバイル ワークフローに強力なバージョン管理、レプリケーション、同期機能を提供します。

\* ファイルジオデータベース:

\* シングルユーザー アクセスをサポートしており、エンタープライズ グレードのインデックス作成機能とマルチユーザー編集機能はありません。

\* Web およびモバイル アプリの編集の要件を満たしていません。

\* モバイル ジオデータベース:

\* モバイル アプリ向けに最適化されていますが、このシナリオに必要なエンタープライズレベルのインデックス作成とマルチユーザー ワークフローはサポートされていません。フィーチャクラスを保存および最適化する手順:

- \* フィーチャクラスをエンタープライズ ジオデータベースに保存します。
- \* 頻繁にクエリされる列にインデックスを作成し、モバイル アプリと Web アプリのクエリのパフォーマンスを最適化します。
- \* ArcGIS Pro などのツールを使用して、モバイルおよび Web アプリ編集用のフィーチャサービスを公開します。

参考文献:

- \* Esri ドキュメント: エンタープライズ ジオデータベースの概要。
- \* パフォーマンス最適化のためのインデックス作成: エンタープライズ環境でのクエリ パフォーマンスを向上させるためのガイドライン。

正解が A である理由: エンタープライズ ジオデータベースは、モバイル アプリや Web アプリでのマルチユーザー編集をサポートし、高度なインデックス作成機能を提供する唯一のストレージオプションです。ファイル ジオデータベースとモバイル ジオデータベースには、このユース ケースに必要な機能が欠けています。

#### 最新問題: 4

GIS 管理者は、ジオデータベース全体のパフォーマンスが低下しているという報告を受けています。ユーザーからは、10,000 件のレコードを追加すると、編集や描画にかかる時間が影響を受けるという報告があります。従来のバージョン管理が使用されています。

次のプロセスは毎週完了します。

- \* インデックスと統計の再構築
- \* ジオデータベースの圧縮
- \* 孤立した接続を削除する

どのようなアクションを取るべきでしょうか?

- A. デフォルトバージョンを使用するように変更
- B. Python 経由でレコードを更新する
- C. バージョンを調整して投稿する

**Answer: C (メッセージを残す)**

シナリオの概要:

- \* ユーザーは、特に 10,000 件のレコードを追加する際の編集と描画において、ジオデータベース全体のパフォーマンスの低下を経験します。
- \* 組織は毎週メンテナンスタスクを実行します。
- \* インデックスと統計の再構築
- \* ジオデータベースの圧縮
- \* 孤立した接続の削除

バージョンを調整して投稿する理由

\* 従来のバージョン管理でパフォーマンスが低下する原因としては、多くの場合、調整されていないバージョンが多すぎることで、状態ツリーが肥大化していることが挙げられます。

\* バージョンのリコンサイルとポストにより、状態の数が削減され、ジオデータベースの圧縮によって冗長な状態が完全に圧縮され、パフォーマンスが向上します。(ArcGIS ドキュメント: リコンサイルとポスト) 代替オプション:

\* オプションA: デフォルトバージョンを使用するように変更する

\* これにより、バージョン管理ワークフローがバイパスされ、パフォーマンス低下の根本的な原因は解決されません。

\* オプションB: Python経由でレコードを更新する

\* Python を使用してレコードを更新しても、未調整バージョンや状態ツリーの非効率性によって発生する問題は解決されません。

したがって、正しいアクションはバージョンを調整してポストし、ジオデータベースの状態ツリーが最適化され、パフォーマンスが回復されるようにすることです。

#### 最新問題: 5

組織は、Web サービスを使用して GIS データを編集する必要があります。データは組織のサーバーにローカルに保存する必要があります。パフォーマンスを向上させるために、特定のビジネス フィールドをデータベースでインデックス化する必要があります。

データにはどのストレージを使用すればよいですか?

A. エンタープライズ ジオデータベース

B. ファイルジオデータベース

C. ホストされたリレーショナルデータベース

**Answer:** ([解答を表示する](#))

すべてのエンタープライズ ジオデータ参照を含む包括的で詳細なステップバイステップの説明:

次の理由により、このシナリオにはエンタープライズ ジオデータベースが最も適切な選択肢です。

1. 組織のサーバーにデータをローカルに保存する必要がある

\* エンタープライズ ジオデータベースを使用すると、組織は PostgreSQL、SQL Server、Oracle などの独自のデータベース管理システム (DBMS) に GIS データをローカルに保存できます。

\* これにより、データ ストレージに対する制御を維持し、データが組織のインフラストラクチャ内に確実に保存されるという要件が満たされます。

2. Web サービス経由で GIS データを編集する

\* エンタープライズ ジオデータベースは ArcGIS Server とシームレスに統合され、Web サービス経由でのデータ編集が可能になります。

\* 組織はフィーチャ サービスを公開して、承認されたユーザーが GIS データをリアルタイムで、または切断された環境 (同期経由) で編集できるようにすることができます。

\* これらのサービスは、バージョン管理や競合解決などの高度な編集ワークフローをサポートします。

### 3. パフォーマンスのための特定のビジネス分野のインデックス作成

\* エンタープライズ ジオデータベースは、クエリと編集のパフォーマンスを向上させる強力なインデックス作成オプションを提供します。

\* あなたはできる：

\* 頻繁にクエリされるフィールドに属性インデックスを作成します。

\* 空間インデックスを使用して、空間クエリの速度を向上します。

\* このレベルのカスタマイズは、特定のビジネス ワークフローのパフォーマンス要求を満たすのに役立ちます。

### 4. 他のストレージオプションに対する利点

\* ファイルジオデータベース：

\* 小規模なデータセットやローカルストレージには適していますが、マルチユーザー編集、Web サービスとの統合、ビジネス フィールドの高度なインデックス作成はサポートされていません。

\* ホストされたリレーショナルデータベース：

\* このオプションは、ArcGIS Online または ArcGIS Enterprise マネージド サービスの一部であり、データをクラウドに保存するため、ローカルストレージの要件に反します。

\* また、エンタープライズ ジオデータベースと同じレベルの制御機能やインデックス作成機能も提供されません。

Esri ドキュメントおよび学習リソースからの参照：

\* エンタープライズ ジオデータベース - ArcGIS Pro ドキュメント

\* ジオデータベースのインデックスの設定

\* 編集用公開機能サービス

結論：

エンタープライズ ジオデータベースは、指定されたすべての要件 (ローカルストレージ、Web サービス編集、パフォーマンスのためのインデックス付きフィールド) を満たすだけでなく、追加のスケラビリティ、セキュリティ、およびマルチユーザー編集機能も提供します。

### 最新問題: 6

GIS 管理者は、2 つの異なるオフィスにある 2 つの GIS アナリスト チームのコラボレーションを促進する必要があります。

各オフィスには、独自のエンタープライズ ジオデータベース内のデータのコピーが必要です。両方のオフィスのアナリストが同じフィーチャクラスを編集します。変更は夜間に同期されます。

GIS 管理者は、両方のチームが連携できるように情報インフラストラクチャを設定する必要があります。

要件を満たすために管理者は何を使用すべきでしょうか？

A. ジオデータベースのレプリケーション

B. データベースのレプリケーション

C. 分散コラボレーション

**Answer: A (メッセージを残す)**

異なるオフィスに所在する 2 つの GIS アナリスト チーム間のコラボレーションを容易にするために、各チームが独自のエンタープライズ ジオデータベース内のデータのコピーを必要とし、同じフィーチャクラスを編集して夜間に変更を同期する必要がある場合、ジオデータベース レプリケーションが適切なソリューションです。

ジオデータベース レプリケーションの理解:

ジオデータベース レプリケーションは、2 つ以上のジオデータベース間でデータのコピーを作成できる ArcGIS のデータ分散方法です。これにより、複数のユーザーが異なる場所にある同じデータセットを操作できるようになり、変更を同期して一貫性を確保できます。

アーキスフォー

ジオデータベース レプリケーションの種類:

ジオデータベース レプリケーションには 3 つの種類があります。

一方向レプリケーション: 変更は親レプリカから子レプリカへの単一方向に送信されます。

双方向レプリケーション: 変更は親レプリカと子レプリカ間で双方向に同期されます。

これは、複数の編集者が異なる場所にある同じデータセットを更新する必要がある場合に適しています。

チェックアウト/チェックイン レプリケーション: データは編集のために子レプリカにチェックアウトされ、その後親レプリカにチェックインされます。

このシナリオでは、双方向レプリケーションが理想的です。これにより、両方のチームが同じフィーチャクラスを編集し、変更を夜間に同期して、両方のジオデータベースの一貫性を保つことができます。

アーキスフォー

代替オプション:

データベース レプリケーション: これは、DBMS レベルでデータベース全体を複製することを指します。データを同期することはできますが、ArcGIS によって管理されるジオデータベース固有の動作、ルール、および関係は考慮されません。

したがって、ジオデータベース固有の機能の同期を必要とするシナリオには適さない可能性があります。

分散コラボレーション: これは、ArcGIS Enterprise のフレームワークであり、マップ、レイヤー、アプリなどのコンテンツを複数の ArcGIS Enterprise 展開間または ArcGIS Enterprise と ArcGIS Online 間で共有できます。ただし、このシナリオで必要なデータの編集と同期を細かく制御することはできません。

ジオデータベース リソース

したがって、両チームがそれぞれのエンタープライズ ジオデータベースで同じフィーチャクラスを編集し、夜間に変更を同期できるという要件を満たすには、ジオデータベース レプリケーションが最適なソリューションです。

最新問題: 7

GIS アナリストは、ArcGIS Pro で複数の画像を含む画像の集中フォルダーを視覚化する必要があります。管理者は、エンタープライズ ジオデータベース内の組織全体でこの画像にアクセスできるように準備する必要があります。

どのストレージメカニズムを使用すればよいですか？

A. ラスターデータセット

B. ラスターカタログ

C. モザイクデータセット

**Answer: C (メッセージを残す)**

Amosaic データセットは、特にエンタープライズ ジオデータベースを通じて組織全体で画像にアクセスする場合に、画像の集中フォルダーを視覚化および管理するための最も適切なストレージメカニズムです。

1. モザイク データセットとは何ですか？

\* モザイク データセットは、衛星画像や航空写真などのラスター データの大規模なコレクションを管理および提供するために最適化されたジオデータベース構造です。

\* 動的なモザイク、レンダリング、クエリ機能を提供します。

2. モザイク データセットを使用する理由

\* 一元管理: モザイク データセットは、データを重複させることなくフォルダーに保存されている複数の画像を参照できるため、画像のソースを一元的に管理できます。

\* スケーラビリティ: 大規模なラスター データのコレクションをサポートし、効率的なクエリ、レンダリング、分析を可能にします。

\* オンザフライ処理: ソースデータを変更せずに画像処理とレンダリングルールを適用するオプションを提供します。

\* エンタープライズ アクセス: エンタープライズ ジオデータベースに保存されたモザイク データセットは、組織全体でアクセスできるようにイメージ サービスとして公開できます。

3. 他の選択肢はないのでしょうか？

\* ラスターデータセット:

\* このオプションは、ジオデータベースに単一の画像またはラスター レイヤーを保存するためのものであり、複数の画像を管理するには適していません。

\* ラスターカタログ:

\* ラスター カタログはラスターを整理しますが、動的なモザイク、レンダリング、クエリ最適化の機能がありません。効率性が低く、現在ではモザイク データセットに取って代わられて時代遅れと見なされています。

モザイク データセットを作成して使用する手順:

\* ArcGIS Pro を開き、エンタープライズ ジオデータベースに接続します。

\* 「モザイク データセットの作成」ツールを使用して新しいモザイク データセットを作成します。

\* 「モザイク データセットにラスターを追加」ツールを使用して、画像の集中フォルダーを指定してモザイク データセットにラスターを追加します。

\* モザイク データセットを ArcGIS Enterprise のイメージ サービスとして公開し、組織全体でアクセスできるようにします。

Esri ドキュメントおよび学習リソースからの参照:

\* モザイクデータセットの概要

\* モザイクデータセットによる画像の管理

結論 :

Amosaic データセットは、エンタープライズ ジオデータベース内の集中フォルダーから複数の画像を管理、視覚化、共有するための最適な機能を提供します。

#### 最新問題: 8

GIS 管理者は、フィーチャクラスの値を更新するための SQL コマンドを作成します。テスト環境では、コマンドはフィーチャクラス テーブルに対して実行されます。すべての値が更新されるわけではないようです。

この問題の原因となっている構成は何ですか？

A. パーティション化されたバージョン非対応のフィーチャクラス

B. 編集が行われた従来のバージョン管理されたデータ

C. フィーチャクラスでアーカイブが有効になっています

**Answer: B (メッセージを残す)**

この問題は、従来のバージョン管理されたデータストアがベース テーブルではなくデルタ テーブル (追加と削除) を編集するために発生します。ベース テーブルに直接適用される SQL 更新はデルタ テーブルをバイパスするため、更新が不完全または不整合になります。

1. 従来のバージョン管理がアップデートに与える影響

\* 従来のバージョン管理では、編集内容はデルタ テーブルに記録されます。

\* A\_<ObjectID> (追加): 新しく挿入された行を追跡します。

\* D\_<ObjectID> (削除): 削除された行を追跡します。

\* SQL コマンドをベース テーブルで直接実行すると、デルタ テーブルのデータには影響がないため、フィーチャクラスに不完全な更新が反映されます。

2. 他の選択肢はないのでしょうか？

\* パーティション化されたバージョン非対応フィーチャクラス:

\* パーティショニングはパフォーマンスを最適化するためにデータを整理しますが、テーブル全体を更新する SQL コマンドには影響しません。

\* フィーチャクラスでアーカイブが有効:

\* アーカイブは個別のアーカイブ テーブルで履歴の変更を追跡しますが、フィーチャクラスの SQL コマンドに直接影響することはありません。

問題を解決する手順:

\* 従来のバージョン管理されたデータの場合は、調整と後処理を使用して値を更新します。これにより、変更がデルタ テーブルとベース テーブル全体に正しく適用されるようになります。

\* または、直接 SQL コマンドを実行する代わりに、ArcGIS Pro や ArcPy などのツールを使用してプログラムでデータを更新します。

Esri ドキュメントおよび学習リソースからの参照:

- \* 従来のバージョン管理の概要
- \* デルタテーブルと従来のバージョン管理

結論:

この問題は、データが従来のバージョン管理されており、直接 SQL コマンドでは編集内容が保存されている差分テーブルが考慮されないために発生します。更新を正しく適用するには、調整およびポスト ワークフローまたは ArcGIS ツールを使用してください。

### 最新問題: 9

GIS データ管理者は、次の条件を満たすデータベース ビューを作成する要求を受け取ります。

- \* データはフィーチャクラスと非空間テーブルから結合されます
- \* ソース フィーチャクラスはバージョン管理されています
- \* ソースは子供バージョンから
- \* 動的に更新する必要がある

ビューはどのように作成すればよいでしょうか?

- A. フィーチャクラスとリポジトリテーブル
- B. フィーチャクラスと非空間テーブル
- C. バージョンビューと非空間テーブル

**Answer:** ([解答を表示する](#))

シナリオを理解する:

- \* リクエストには、バージョン化されたフィーチャクラスと非空間テーブルからのデータを動的に結合するビューの作成が含まれます。
- \* データは子供バージョンからのものであり、動的に更新される必要があります。

重要な考慮事項:

- \* バージョン対応フィーチャクラス: バージョン対応ジオデータベースの標準フィーチャクラスは、編集内容をデルタ テーブルに保存します (追加と削除)。特定のバージョンのデータにアクセスするには、これらの編集内容を含むバージョン対応ビューを使用する必要があります。
- \* 動的更新: バージョン管理されたビューで作成されたビューでは、データが最新バージョンの編集を反映していることが保証されます。

ビューを作成する手順:

- \* フィーチャクラスのバージョン対応ビューを識別します (バージョン管理中に自動的に作成されます)。
- \* 適切なキーでバージョン対応ビューと非空間テーブルを結合する SQL クエリを作成します。
- \* SQL クエリをデータベース ビューとして保存します。

参考文献:

- \* Esri ドキュメント: バージョン管理されたビュー。

\* バージョン管理されたデータを含むビューの作成: SQL でバージョン管理されたビューを他のテーブルと結合するためのガイドライン。

正解が C である理由: バージョン対応ビューを使用すると、指定された子バージョンからの編集がデータに反映されます。このビューを非空間テーブルと結合すると、動的更新の要件が満たされます。オプション A と B では、バージョン対応の子バージョンからのデータが動的に提供されません。

#### 最新問題: 10

ArcGIS Pro を使用する AGIS アナリストは、フィーチャ データセットに格納されているバージョン対応フィーチャクラスにデータを再ロードする必要があります。フィーチャクラスは、ジオデータベース トポロジに参加しています。

GIS アナリストはどのような手順を踏むべきでしょうか?

A. Truncate Table ツールを実行し、Append を使用してデータをロードします。

B. フィーチャクラス内のすべての行を削除し、オブジェクトのロードを使用してデータをロードします。

C. フィーチャクラスのすべての行を削除し、Append を使用してデータをロードします。

**Answer:** ([解答を表示する](#))

シナリオを理解する:

\* フィーチャクラスはバージョン管理されており、ジオデータベース トポロジに参加しています。

\* 目標は、バージョン管理とトポロジの整合性を維持しながらデータを再ロードすることです。

データの再ロードに関する重要な考慮事項:

\* テーブルの切り捨て: テーブルの切り捨てツールは、ジオデータベース内の個々の行の削除をログに記録せずに、フィーチャクラス内のすべての行を効率的に削除します。これは、パフォーマンスへの影響を最小限に抑えながらデータを消去するための推奨される方法です。

\* 追加ツール: テーブルを切り捨てた後、追加ツールはフィーチャクラスに新しいデータをロードし、トポロジとバージョン管理構造がそのまま維持されるようにします。

\* 行の削除の回避: 行を手動で削除すると、各削除がデルタ テーブルに記録され、特にバージョン管理されたデータセットの場合、潜在的なパフォーマンスのボトルネックと不要なトランザクション ログ記録につながります。

\* ジオデータベース トポロジの考慮事項: 空間整合性を確保するために、データを再ロードした後にトポロジ ルールを検証する必要があります。

データを再ロードする手順:

\* 既存のレコードを削除するには、「Truncate Table」ツールを使用します。

\* Append ツールを使用して、新しいデータをフィーチャクラスに読み込みます。

\* 再ロード後にエラーがないか確認するために、ジオデータベース内のトポロジを検証します。

## 参考文献:

\* Esri ドキュメント: テーブルの切り捨て。

\* バージョン管理されたフィーチャクラスへのデータのロード: バージョン管理されたトポロジ対応データセットのベスト プラクティス。

正解が A である理由: Truncate Table ツールを実行すると、効率的なデータ消去が保証され、Append ツールを使用すると、ジオデータベースのバージョン管理とトポロジ構造が維持されます。オプション B と C では、不要な行レベルの削除が行われますが、これは非効率的であり、バージョン管理されたワークフローを中断する可能性があります。

## 最新問題: 11

AGIS データ管理者は、次のワークフローを使用して、エンタープライズ ジオデータベース内に新しいフィーチャクラスを作成しています。

\* 5つのフィーチャクラスがフィーチャデータセットに追加されます

\* フィーチャデータセットは、move-edits-to-base オプションなしでバージョン管理対象として登録されます。

\* 次に、同じフィーチャデータセットに別のフィーチャクラスが追加されます

フィーチャ データセット内のフィーチャクラスを編集しようとする、ユーザーにはエラーメッセージが表示されます。

管理者は何をすべきでしょうか？

**A.** フィーチャデータセットのバージョン管理を解除し、再度バージョン管理に登録します。

**B.** フィーチャデータセットにバージョン管理としてもう 1 回登録します

**C.** 編集ワークフローを、move-edits-to-base オプションを使用してバージョン管理された編集に切り替える

**Answer:** ([解答を表示する](#))

シナリオの概要:

\* フィーチャ データセットに 5 つのフィーチャクラスが追加され、move-edits-to-base オプションを使用せずにバージョン対応として登録されます。

\* その後、同じフィーチャ データセットに別のフィーチャクラスが追加されます。

\* フィーチャ データセット内のフィーチャクラスを編集しようとする、エラーが発生します。

問題の原因: フィーチャ データセットがバージョン管理対象として登録されている場合、その中のすべてのフィーチャクラスはバージョン管理状態の一貫性を維持する必要があります。以前にバージョン管理されたフィーチャ データセットに新しいフィーチャクラスを追加すると、同期が中断され、編集中にエラーが発生する可能性があります。

解決:

\* フィーチャ データセットでバージョン管理の登録を解除: これにより、データセット内のすべてのフィーチャクラスからバージョン管理が削除され、バージョン管理の状態がリセットされます。

\* フィーチャ データセットを再度バージョン対応として登録します。これにより、新しく追加されたフィーチャクラスも含め、すべてのフィーチャクラスが同じバージョン対応状態で正しく登録されます。(ArcGIS ドキュメント: データセットをバージョン対応として登録する) 代替オプション:

\* オプション B: フィーチャ データセット全体を登録解除して再登録しない限り、バージョン管理の競合が続くため、フィーチャ データセットを再度登録しても問題は解決されません。

\* オプション C: 編集内容をベースに移動するオプションへの切り替えは不要であり、編集ワークフローが変更されるため、現在の設定やユーザーのニーズに合わない可能性があります。

したがって、管理者はフィーチャ データセットをバージョン管理から登録解除し、再登録してエラーを解決する必要があります。

### 最新問題: 12

組織では、RDBMS ユーザーの数を減らす必要があります。ArcGIS Enterprise と ArcGIS Pro が実装されています。編集者は編集内容を分離し、公開される前に編集内容がレビューされるようにする必要があります。

GIS 管理者はどの編集モデルを実装する必要がありますか?

- A. ブランチバージョン管理
- B. 従来のバージョン管理
- C. バージョン管理されていない編集

**Answer: B (メッセージを残す)**

シナリオを理解する:

\* 編集者は、変更が他のユーザーにすぐに表示されないように、編集内容を分離する必要があります。

\* 編集内容は公開される前に確認される必要があります、構造化された承認プロセスが必要になります。

\* 組織は RDBMS ユーザーの数を削減することを目指しており、アクセスと権限の集中管理を提案しています。

編集モデルの概要:

\* ブランチ バージョン管理: Web ベースのワークフロー用に設計されており、各エディターが直接 RDBMS にアクセスする必要はありません。ただし、ブランチ バージョン管理で行われた編集は本質的に共同作業であり、ユーザーごとのブランチ ワークフローによって明示的に制御されない限り分離されないため、複雑さが増します。

\* 従来のバージョン管理:

\* プライベートバージョンによる個別の編集をサポートします。

\* 編集者は独自のバージョンを作成し、変更を加えて、デフォルト バージョンと調整して投稿することでレビューのために送信できます。

\* RDBMS への直接アクセスが集中化されるため、個々の RDBMS ユーザーの必要性が軽減されます。

\* バージョン管理されていない編集: 分離された編集やバージョン管理されたワークフローをサポートしていないため、このシナリオには適していません。

従来のバージョン管理を実装する手順:

\* フィーチャクラスをバージョン管理された状態でエンタープライズ ジオデータベースに登録します。

\* 編集者が個別の編集を行うためのプライベート バージョンを作成できるようにします。

\* レビュー後に編集を調整して投稿するためのワークフローを実装します。

参考文献:

\* Esri ドキュメント: 従来のバージョン管理。

\* バージョン管理された編集のベスト プラクティス: 編集を分離して確認するためのガイドライン。

正解が B である理由: 従来のバージョン管理は、編集を分離し、投稿前にレビューを可能にし、集中バージョン管理によって RDBMS ユーザーの数を削減するなど、すべての要件を満たしています。

ブランチのバージョン管理は Web 中心であり、構造化されたレビュー プロセスが欠けています。一方、バージョン管理されていない編集では分離やバージョン管理はサポートされません。

### 最新問題: 13

GIS 管理者は、ジオデータベース ユーザーが、子バージョンのデータをマップに追加するとパフォーマンスが低下すると報告していることを知りました。

\* 組織は複雑な従来のバージョンツリーアーキテクチャを使用しています

\* Python スクリプトはバッチ調整/ポスト操作を完了し、ジオデータベースを圧縮し、データ所有者はインデックスを再構築して統計を更新します。

\* Python スクリプトは、ジオデータベース接続がほとんどまたはまったく行われないう状態で夜間に実行されます。どの Analyze Datasets パラメータをチェックする必要がありますか?

A. システムテーブルを含める

B. 選択したデータセットのアーカイブ テーブルを分析する

C. 選択したデータセットのベーステーブルを分析する

**Answer: (解答を表示する)**

複雑な従来のバージョン ツリーで子バージョンからデータを追加する際にパフォーマンスの問題が発生する場合、多くの場合、基本テーブルに問題があることを示しています。この場合、選択したデータセットの基本テーブルを分析するパラメーターが最も関連しています。

1. 従来のバージョン管理におけるベーステーブルの役割

\* 従来のバージョン管理では、ベース テーブルにフィーチャクラスまたはテーブルの元のデータが格納されます。子バージョンで行われた変更は、デルタ テーブル (追加と削除) で追跡されます。

\* ベース テーブルが最適化されていない場合 (古い統計、断片化されたインデックスなど)、データのクエリやレンダリング時にパフォーマンスが低下する可能性があります。

## 2. ベース テーブルを分析する理由

\* 選択したデータセットの基本テーブルを分析]パラメータは、基本テーブルのデータベース統計を評価および更新し、クエリの最適化を改善します。

\* このプロセスにより、データベース クエリ オプティマイザーはデータを取得するときに効率的な決定を下すことができます。

## 3. 他の選択肢はないのでしょうか?

\* システムテーブルを含める:

\* これは、管理タスクには重要ですが、ユーザー データセットのパフォーマンスの問題とは関係のないジオデータベース システム テーブルを分析します。

\* 選択したデータセットのアーカイブテーブルを分析する:

\* これは、アーカイブが有効になっているデータセットに固有のもので、このシナリオでは、アーカイブが使用されているとは言及されていません。

ベーステーブルを分析する手順:

\* OpenArcGIS Pro または、データセットの分析ツールで Python スクリプトを使用します。

\* パフォーマンスの問題があるデータセットを指定します。

\* [選択したデータセットの基本テーブルを分析] パラメータを選択します。

\* ツールを実行し、更新された統計を監視します。

Esri ドキュメントおよび学習リソースからの参照:

\* データセット分析ツール

\* クエリパフォーマンスの向上

結論:

従来のバージョン管理で子バージョンからデータを追加するときに統計を更新し、パフォーマンスを向上させるには、選択したデータセットの基本テーブルを分析]パラメータを使用する必要があります。

## 最新問題: 14

GIS データ管理者は、権限のないユーザーによるフィーチャクラス内の一部のフィールドへのアクセスを制限する必要があります。

データ管理者はどの方法を使用すべきでしょうか?

A. レイヤーファイル

B. クエリレイヤー

C. データベースビュー

**Answer: C (メッセージを残す)**

フィーチャクラス内の特定のフィールドへのアクセスを制限するには、GIS データ管理者はデータベース ビューを使用する必要があります。

### 1. データベース ビューとは何ですか？

\* データベース ビューは、SQL クエリによって作成される仮想テーブルです。1 つ以上のテーブルまたはフィーチャクラスのデータが表示されますが、アクセスできるのは指定されたフィールドと行のみです。

\* ビューを使用すると、管理者は基になるデータを変更せずに、ユーザーに表示されるフィールドとレコードを制御できます。

### 2. データベース ビューを使用する理由

\* 特定のユーザーに表示されるフィールド (または行) を制限することで、データ アクセスをきめ細かく制御できます。

\* マルチユーザー環境でのデータ セキュリティ ポリシーの適用に役立ちます。

\* 使用ケースに応じて、読み取り専用レイヤーとして共有したり、更新権限を制限して共有したりできます。

### 3. 他の選択肢はないのでしょうか？

\* レイヤーファイル:

\* レイヤー ファイル (\*.lyr) は、データのシンボル化と表示方法のみを定義します。データベース レベルでのフィールド アクセスは制限されません。権限のないユーザーは、直接データベース接続を通じて非表示のフィールドにアクセスできます。

\* クエリレイヤー:

\* クエリ レイヤーを使用すると、ArcGIS でデータを表示するときにカスタム SQL クエリを実行できますが、セキュリティ対策にはなりません。ユーザーはクエリを変更またはバイパスして、すべてのフィールドにアクセスできます。

データベース ビューを作成する手順:

\* 必要なフィールドのみを指定して、SQL を使用してビューを定義します。

ビュー制限付きビューの作成

フィールド1、フィールド2を選択

フィーチャクラスから

WHERE <条件>;

\* ベース テーブルへのアクセスを制限しながら、承認されたユーザーにビューへの権限を付与します。

\* 必要に応じて、ArcGIS でビューを読み取り専用レイヤーとして公開します。

Esri ドキュメントおよび学習リソースからの参照:

\* データベースビューの作成と使用

\* エンタープライズ ジオデータベースのデータ セキュリティ

結論:

データベース ビューは、フィーチャクラス内の特定のフィールドへのアクセスを制限し、データ共有の柔軟性を提供しながらデータのセキュリティを確保する最も効果的な方法です。

## 最新問題: 15

データベース管理者は、エンタープライズ ジオデータベースを新しいサーバーに移動する必要があります。新しいエンタープライズ ジオデータベースはそのまま保持する必要があります。

どのプロセスを使用すべきでしょうか？

A. ファイルジオデータベースにエクスポート

B. 双方向レプリケーション

C. RDBMS エクスポート/インポート

**Answer: C (メッセージを残す)**

エンタープライズ ジオデータベースをそのまま維持しながら新しいサーバーに移動するには、RDBMS エクスポート/インポート プロセスが適切な方法です。

1. RDBMS エクスポート/インポートを使用する理由

\* エンタープライズ ジオデータベースは、基盤となる RDBMS (SQL Server、PostgreSQL、Oracle など) と緊密に統合されています。データベース全体をエクスポートおよびインポートすると、次のことが保証されます。

\* すべてのジオデータベース構成 (テーブル、インデックス、メタデータなど) は保持されません。

\* 移行プロセス中にデータの整合性が失われることはありません。

\* この方法では、ジオデータベースの構造と関係が維持されます。

2. 他の選択肢はないのでしょうか？

\* ファイルジオデータベースへのエクスポート:

\* ファイル ジオデータベースにエクスポートするとデータの転送が可能になりますが、ユーザー権限、バージョン管理、レプリケーション構成などのエンタープライズ ジオデータベース構造は保持されません。

\* 双方向レプリケーション:

\* レプリケーションは、ジオデータベース全体を新しいサーバーに移動するためではなく、ジオデータベース間のデータ変更を同期するために設計されています。また、一部の管理構成が残される場合もあります。

3. RDBMSエクスポート/インポートの手順

\* データベースをエクスポートします:

\* RDBMS ツール (PostgreSQL の場合は pg\_dump、SQL Server Management Studio など) を使用して、ジオデータベースの完全バックアップを作成します。

\* 関連するすべてのスキーマ、インデックス、メタデータが含まれていることを確認します。

\* 新しいサーバーにインポート:

\* 新しいサーバーに RDBMS をセットアップし、エンタープライズ ジオデータベース用に構成します。

\* バックアップ ファイルをインポートして、新しいサーバー上のジオデータベースを復元します。

\* 移行後の手順:

\* ArcGIS Pro または ArcGIS Server で接続を再構成し、新しいジオデータベースを指すようにします。

\* すべての機能が期待どおりに動作することを確認するためにテストします。

Esri ドキュメントおよび学習リソースからの参照:

\* エンタープライズ ジオデータベースのバックアップと復元

\* バックアップと復元のための RDBMS ツール

結論:

RDBMS エクスポート/インポート プロセスにより、すべての構成とデータの整合性を維持しながら、エンタープライズ ジオデータベースを新しいサーバーに完全に移行できます。

### 最新問題: 16

ユーザーが誤ってエンタープライズ ジオデータベースのフィーチャ データセットを削除しました。

この問題を解決するにはどのテクノロジーを使用すべきでしょうか?

A. 高可用性

B. バックアップ

C. アーカイブ

**Answer:** ([解答を表示する](#))

シナリオを理解する:

\* エンタープライズ ジオデータベース フィーチャ データセットが誤って削除されました。

\* 組織はデータセットを元の状態に復元する必要があります。

利用可能なテクノロジー:

\* 高可用性: 高可用性設定 (例: フェイルオーバー システム) により、ハードウェアまたはソフトウェアの障害発生時にもジオデータベースへの継続的なアクセスが確保されます。ただし、高可用性では誤って削除されたデータは復元されません。

\* バックアップ: バックアップは、特定の時点で取得されたジオデータベースのスナップショットです。これにより、管理者は削除されたデータセットを復元したり、データ損失シナリオから回復したりできます。

\* アーカイブ: アーカイブはバージョン管理されたジオデータベース内の履歴編集を追跡しますが、誤って削除されたデータセットの回復は提供しません。

データセットを回復する手順:

\* エンタープライズ ジオデータベースの最新のバックアップを特定します。

\* ジオデータベースを復元するか、バックアップから特定のフィーチャ データセットを抽出します。

\* 復元されたデータを確認し、必要に応じて進行中の更新と同期します。

参考文献:

\* Esri ドキュメント: バックアップと復元。

\* データ保護のベスト プラクティス: データの損失を防ぐために定期的なバックアップを実装するためのガイドライン。

正解が B である理由: バックアップは、誤って削除されたフィーチャ データセットを回復するための最も信頼性の高いソリューションです。高可用性により稼働時間は保証されますが、データの回復には対応していません。また、アーカイブではデータセット全体を保存するのではなく、編集内容を追跡します。

有効な **EGMP2201** 問題集は GoShiken.com が提供された合格しやすい EGMP2201 試験問題集! GoShiken.com が最新の **EGMP2201** 試験問題集を提供しています。

GoShiken.com EGMP2201 試験問題は最新で、解答が正確でございます。最新の GoShiken.com EGMP2201 問題集をゲットする人はこちら:

<https://www.goshiken.com/Esri/EGMP2201-mondaishu.html> (6730%OFF問題集溶と正解付きで 30%w 特別割引コード: **Freepdfdumps**)

#### 最新問題: 17

AGIS データベース管理者は、大規模なエンタープライズ ジオデータベース

facility\_inspections フィーチャクラスでのクエリ パフォーマンスを向上させるためにインデックスを作成する必要があります。

最適化するクエリは次のとおりです。

施設タイプ、検査日を選択

facility\_inspections から、inspector\_name = 'JQSmith' で、inspection\_date で並べ替えます。

A. 管理者はどの列にインデックスを作成する必要がありますか?

B. 施設タイプ、検査日

C. facility\_type.inspector\_name

D. 検査官名、検査日

**Answer: D (メッセージを残す)**

シナリオを理解する:

\* クエリは、inspector\_name に基づいて行をフィルタリングし、結果を Inspection\_date で並べ替えます。

\* インデックスを作成すると、クエリ実行中にスキャンされる行数が減り、データの取得が最適化されます。

インデックス作成に関する重要な考慮事項:

\* プライマリ フィルター列:inspector\_name 列は、WHERE 句のプライマリ フィルターです。

この列にインデックスを付けると、フィルター条件に一致する行をすばやく識別できます。

\* ソートの最適化: ORDER BY Inspection\_date 句では、inspection\_date を複合インデックスの 2 番目の列としてインデックス付けすることで、フィルタリングされた結果のソートが高速化されます。

インデックスの選択:

\* Inspection\_date と Inspection\_name の複合インデックスにより、効率的なクエリ実行が可能になります。

\* inspector\_name は高速フィルタリングを保証します。

\* Inspection\_date はソートを最適化します。

\* facility\_type のような無関係な列はクエリの一部ではないため、インデックス作成は不要です。

インデックスを作成する手順:

\* エンタープライズ ジオデータベースでは、データベース管理ツールまたは SQL コマンドを使用して複合インデックスを作成します。

インデックス idx\_facility\_inspections を作成します

ON facility\_inspections (検査官名、検査日);

参考文献:

\* Esri ドキュメント: インデックスの作成と管理。

\* SQL インデックスのベスト プラクティス: クエリを最適化するための複合インデックス作成のガイドライン。

正解が D である理由: inspector\_name と Inspection\_date の複合インデックスはクエリ構造を直接処理し、WHERE フィルターと ORDER BY 句の両方を最適化します。

最新問題: 18

大規模な政府機関では、すべての部門がスタンバイ データ センターに同等のデータ プレゼンスを確立することを義務付けています。

GIS データベース管理者はどのテクノロジーを推奨すべきでしょうか?

A. データベースのレプリケーション

B. ジオデータベースのレプリケーション

C. 切断された同期

**Answer: A (メッセージを残す)**

スタンバイ データ センターに同等のデータが存在することを必要とする大規模な政府機関にとって、データベース レプリケーションは理想的なソリューションです。

1. データベースレプリケーションとは何ですか?

\* データベース レプリケーションでは、プライマリ データベースからセカンダリ データベースにほぼリアルタイムで、またはスケジュールに基づいてデータを複製します。

\* これにより、両方のデータベースが同期され、一方のデータベースに障害が発生した場合でもデータを提供できるようになります。

2. データベースレプリケーションが要件に適合する理由

\* スタンバイ データ センター: データベース レプリケーションにより、セカンダリ データセンターのデータの完全に同等なコピーが提供されます。

\* 高可用性と災害復旧: プライマリ データベースが使用できない場合、スタンバイ データベースがすぐに引き継ぎ、ビジネスの継続性を確保します。

3. 他の選択肢はないのでしょうか?

\* ジオデータベースレプリケーション:

\* ジオデータベース コンテンツを複製するように設計されていますが、通常はフィールド編集の同期など、GIS 固有のワークフローに使用されます。データベースの非空間コンポーネントの同等性は保証されません。

\* 大規模な組織全体のレプリケーションのニーズには適していません。

\* 切断された同期:

\* これは、デバイスが後で編集内容を中央データベースと同期するオフライン編集ワークフローで使用されます。同等のスタンバイ データベースを維持するのには適していません。

#### 4. データベースレプリケーションの種類

\* 非同期レプリケーション: 更新はスケジュールされた間隔でレプリケートされるため、柔軟性は得られますが、若干の遅延が発生します。

\* 同期レプリケーション: 更新はリアルタイムで行われるため、両方のデータベースが常に同一であることが保証されます。

データベースレプリケーションを実装する手順:

\* 組織の DBMS (SQL Server、PostgreSQL、Oracle など) でプライマリ データベースとスタンバイ データベースを構成します。

\* DBMS の組み込みレプリケーション ツール (SQL Server の Always On、PostgreSQL の ストリーミング レプリケーションなど) を使用します。

\* レプリケーション プロセスが正しく機能していることを確認するために監視を設定します。

Esri ドキュメントおよび学習リソースからの参照:

\* DBMSにおけるデータベースレプリケーション

\* データベースレプリケーションによる災害復旧

結論:

データベース レプリケーションは、スタンバイ データ センターに同等のデータを確立し、高可用性と災害復旧を保証するために推奨されるテクノロジーです。

#### 最新問題: 19

井戸フィーチャクラスには、井戸ごとに 1 つの行があります。well\_inspections テーブルには、井戸が検査された時間ごとに 1 つの行があります。すべての検査日は、マップ上の各井戸の周りに集められたラベルとして表示する必要があります。

この要件を満たすにはどのような種類の関連付けを使用する必要がありますか?

A. 関係クラス

B. 関連付ける

C. 参加する

**Answer:** ([解答を表示する](#))

#### 最新問題: 20

エンタープライズ ジオデータベースは、フェデレーション データベースからの「プッシュ」バッチ SQL プロセスを介して定期的に更新されます。

ビジネス ルールにより、1 つのフィールドの INSERT 値の 1 対 1 の一致が防止されます。GIS 管理者は、ジオデータベースの属性値が正しく入力されていることを確認する必要があります。管理者はどのオプションを使用する必要がありますか？

- A. 属性ルール
- B. DBMS トリガー
- C. 空間ビュー

**Answer: A (メッセージを残す)**

シナリオを理解する:

- \* エンタープライズ ジオデータベースは、フェデレーション データベースからの「プッシュ」SQL プロセスを通じて定期的に更新を受け取ります。
- \* ビジネス ルールでは、データ挿入プロセス中に属性値の変換または修正が必要になります。
- \* GIS 管理者には、属性値がこれらのルールに準拠していることを確認するためのメカニズムが必要です。

オプションの概要:

- \* 属性ルール:
- \* 属性ルールは、データ編集に計算や制約を適用することで、データの整合性を強化できます。
- \* 計算ルールを使用すると、特定のロジックに基づいて属性値を自動的に導出できるため、値がビジネス ルールに準拠していることが保証されます。
- \* DBMS トリガー: DBMS トリガーは、データベース イベント (挿入など) が発生したときにプロシージャを実行します。

ただし、これらはジオデータベース環境の外部で管理されるため、ArcGIS ワークフローとの統合性が低くなります。

- \* 空間ビュー: 空間ビューは空間データと非空間データを組み合わせますが、データの読み込み中または編集に属性値を適用または変更することはありません。

属性ルールを構成する手順:

- \* 変換が必要なフィールドに計算属性ルールを定義します。
- \* ビジネス ルールに基づいて属性値を導出または検証するロジックを指定します。
- \* ルールをデータセットに適用します。
- \* データが挿入されると、ルールによって値が自動的に更新または修正されます。

参考文献:

- \* Esri ドキュメント: 属性ルール。
- \* 属性ルール タイプ: エンタープライズ ジオデータベースで計算ルールと検証ルールを使用するためのベスト プラクティス。

正解が A である理由: 属性ルールは、属性値を管理し、データ編集にビジネス ロジックに準拠することを保証するように特別に設計されています。DBMS トリガーは外部にあ

り、ジオデータベース中心ではありませんが、空間ビューは挿入中の値の変換をサポートしていません。

#### 最新問題: 21

GIS 管理者は、従来のバージョン管理されたフィーチャ データセット内のデータを編集できないというユーザーからの報告を受けています。メンテナンス期間中にフィーチャ データセットにフィーチャクラスが追加されました。次のトラブルシューティング手順を実行しましたが、動作は修正されませんでした。

- \* フィーチャデータセットの権限を確認しました
- \* バージョン管理タイプの接続ファイルをチェックしました
- \* インデックスと統計を再構築

管理者は何をすべきでしょうか？

- A. データ所有者として接続し、データを編集する
- B. フィーチャデータセットのバージョン登録を解除します
- C. フィーチャデータセットをバージョン管理対象として再登録する

**Answer: C (メッセージを残す)**

新しいフィーチャクラスが追加された後、ユーザーが従来のバージョン管理されたフィーチャ データセットを編集できない場合は、フィーチャ データセットをバージョン管理されたものとして再登録する必要があります。

#### 1. バージョン管理として再登録する理由

- \* バージョン管理されたフィーチャ データセットにフィーチャクラスを追加するには、フィーチャ データセット全体をバージョン管理用に再登録する必要があります。
- \* この手順により、新しいフィーチャクラスがバージョン管理システムに含まれ、バージョン管理されたワークフローに参加できるようになります。

#### 2. 他の手順を実行しても問題が解決しなかったのはなぜですか？

- \* チェックされた権限: 正しい権限では、フィーチャクラスを追加した後に再登録する必要がなくなります。
- \* 確認済みの接続ファイル: 接続ファイルで従来のバージョン管理が使用されていることを確認しても、登録の欠落は解決されません。
- \* インデックスと統計の再構築: これによりパフォーマンスは向上しますが、バージョン管理には影響しません。

#### 3. 他の選択肢はないのでしょうか？

- \* データ所有者として接続し、データを編集します。
- \* データ所有者であっても、フィーチャデータセットが再登録されるまで編集はできません。
- \* フィーチャデータセットのバージョン管理を解除します。
- \* 登録を解除するとバージョン情報が削除され、デルタ テーブルのデータ損失が発生する可能性があります。

フィーチャ データセットをバージョン管理済みとして再登録する手順:

- \* ArcGIS Pro を開き、データ所有者として接続します。

\* フィーチャ データセットを右クリックし、[管理] > [バージョン対応として登録] を選択します。

\* 必要に応じて、ベース テーブルへの編集内容を保持するオプションを選択します。

\* フィーチャデータセットの編集を保存してテストします。

Esri ドキュメントおよび学習リソースからの参照:

\* データをバージョン管理として登録する

\* バージョン管理されたフィーチャデータセットの管理

結論:

管理者は、新しく追加されたフィーチャクラスを含め、従来のバージョン管理環境での編集の問題を解決するために、フィーチャ データセットをバージョン管理対象として再登録する必要があります。

## 最新問題: 22

GIS アナリストは、組織と LIDAR データの大規模なリポジトリを共有する必要があります。この LIDAR データには、ブレークラインに適用されるサーフェス制約があります。

GIS アナリストはどのタイプのデータセットを使用すべきでしょうか?

A. モザイクデータセット

B. 特徴データセット

C. LAS データセット

**Answer: C (メッセージを残す)**

シナリオを理解する:

\* GIS アナリストは、LIDAR データの大規模なリポジトリを共有する必要があります。

\* データには、地形やサーフェスのルールを適用するために使用されるブレークラインなどのサーフェス制約が含まれます。

データセットの種類の概要:

\* モザイク データセット: 画像や標高グリッドなどのラスター データの大規模なコレクションを管理するために設計されています。LIDAR ポイント クラウド データ用に特別に最適化されているわけではありません。

\* フィーチャ データセット: ジオデータベース内の関連フィーチャクラスのコンテナ。LIDAR データやサーフェス制約の管理とは無関係です。

\* LAS データセット: LIDAR ポイント クラウドを管理するために設計された特殊なデータセット。ポイント分類、サーフェス制約 (ブレークラインなど)、LIDAR データの効率的なクエリまたは視覚化をサポートします。

LAS データセットを作成して共有する手順:

\* ArcGIS Pro で LAS データセットを作成し、LIDAR ファイル (LAS または ZLAS 形式) を追加します。

\* LAS データセットのプロパティでサーフェス制約 (ブレークライン) を定義します。

\* LAS データセットを組織アクセス用のサービスまたはパッケージとして共有します。

参考文献:

\* Esri ドキュメント: LAS データセット。

\* LAS データセットでのブレイクラインの管理: サーフェス制約を組み込むための手順。正解が C:LAS データセットである理由は、ブレイクラインなどのサーフェス制約を伴う LIDAR データを管理および共有するために明示的に設計されているからです。モザイク データセットとフィーチャ データセットはこの目的には適していません。

#### 最新問題: 23

GIS 管理者は、フィーチャクラスの編集に行われた編集内容をすべてのユーザーがすぐに確認できるようにする必要があります。

どの構成を使用すればよいですか?

- A. 従来のバージョン編集
- B. バージョン管理されていない編集
- C. ブランチバージョン編集

**Answer:** ([解答を表示する](#))

シナリオを理解する:

\* フィーチャクラスに対して行われた編集は、すべてのユーザーにすぐに表示される必要があります。

\* これは、バージョン管理のオーバーヘッドなしで基本テーブルを直接編集する必要がありますを示しています。

編集構成の概要:

\* 従来のバージョン編集: デルタ テーブルに対して編集が行われ、他のユーザーに表示されるようにするには、バージョンの調整と投稿が必要です。これにより遅延が発生し、即時の可視性の要件を満たしません。

\* バージョン非対応編集: 編集内容はフィーチャクラスのベース テーブルに直接適用され、すべてのユーザーにすぐに表示されます。これは、シナリオに適した選択です。

\* ブランチ バージョン編集: 編集はブランチ バージョンで行われ、編集がデフォルトのブランチに保存されない限り、他のユーザーにはすぐには表示されません。これは、すぐに表示したい場合は最適な選択ではありません。

バージョン管理されていない編集を構成する手順:

\* エンタープライズ ジオデータベースでフィーチャクラスが非バージョン編集モードに設定されていることを確認します。

\* すべてのユーザーにベーステーブルを編集するための必要な権限を付与します。

\* データセットに加えられたすべての編集は、すべてのユーザーに即座に反映されます。

参考文献:

\* Esri ドキュメント: バージョン管理されていない編集。

正解が B である理由: バージョン管理されていない編集は、すべてのユーザーが編集内容をすぐに確認できる唯一の構成です。従来のバージョン管理とブランチ バージョン管理には、編集内容の確認を遅らせるワークフローが伴います。

#### 最新問題: 24

GIS 管理者は、ブランチ バージョン データセットの同期コピーを作成する必要があります。編集は両方のコピーに対して実行する必要があります。

データはどのように複製すればよいですか？

- A. 分散コラボレーション
- B. ジオデータベースのレプリケーション
- C. DBMSレプリケーション

**Answer: B (メッセージを残す)**

シナリオの概要:

\* GIS 管理者は、ブランチ バージョン データセットの非同期コピーを作成する必要があります。

\* 両方のコピーは編集可能でなければなりません。

ジオデータベース レプリケーションを使用する理由

\* ジオデータベース レプリケーションでは、親ジオデータベースと子ジオデータベースの両方で編集を許可しながら、データセットの同期コピーの作成をサポートします。

\* ブランチ バージョン データの場合、レプリケーションにより、親ジオデータベースまたは子ジオデータベースのいずれかで行われた編集が双方向レプリカを使用して同期されることが保証されます。(ArcGIS ドキュメント: ジオデータベース レプリケーション) このシナリオでのジオデータベース レプリケーションの主な機能:

\* 双方向レプリケーションにより、変更を同期しながら両側で編集が可能になります。

\* ブランチのバージョン管理をサポートし、バージョン管理されたワークフローがそのまま維持されることを保証します。

\* 両方のジオデータベース間でスキーマの一貫性を維持します。

代替オプション:

\* オプションA: 分散コラボレーション

\* コラボレーションは、ArcGIS Enterprise 環境間でデータを共有するのに適していますが、両側での編集のためのアクティブな同期はサポートしていません。

\* オプション C: DBMS レプリケーション

\* DBMS レベルのレプリケーションは生データのレプリケーションを処理しますが、ブランチ バージョン管理などのジオデータベース固有の機能は保持されません。

したがって、ジオデータベース レプリケーションは、両方のジオデータベースでブランチ バージョン データセットを同期および編集するための適切な方法です。

**Valid EGMP2201 Dumps** shared by GoShiken.com for Helping Passing EGMP2201 Exam! GoShiken.com now offer the **newest EGMP2201 exam dumps**, the GoShiken.com EGMP2201 exam **questions have been updated** and **answers have been corrected** get the **newest** GoShiken.com EGMP2201 dumps with Test Engine

here: <https://www.goshiken.com/Esri/EGMP2201-mondaishu.html> (67 Q&As Dumps,  
**30%OFF Special Discount: Freepdfdumps**)