



Esri EGMP2201 Practice Questions

Shared by Clarke on 14-10-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page



Question 1

Question Type: MultipleChoice

A GIS data administrator creates a replica pair to publish changes from the organization's production server to a consultant's server. Edits are being performed on the data in the consultant's replica and are overwritten as they conflict with edits applied during synchronization.

Which replication type is causing this issue?

Options:

- A- Checkout/check-in
- B- One-way, parent to child
- C- One-way, child to parent

Answer:

B

Explanation:

The issue arises because One-way, parent to child replication is being used. In this type of replication, changes from the parent replica (production server) overwrite the data in the child replica (consultant's server) during synchronization, regardless of edits made in the child replica.

1. One-way Replication Workflow

Parent to Child: Changes are pushed from the parent to the child. The child can make local edits, but these edits are not sent back to the parent, and they can be overwritten when synchronizing.

In this case, the consultant's edits are overwritten because the synchronization is unidirectional from the production server to the consultant's server.

2. Issue with Conflicting Edits

Since One-way, parent to child replication does not support bi-directional synchronization or conflict detection, local changes in the child replica are not preserved if the parent replica pushes updates that conflict with them.

3. Why Not Other Options?

Checkout/check-in:

This replication type allows edits to be made in the checkout replica and reconciled back to the parent during check-in. This ensures that conflicting edits are addressed.

One-way, child to parent:

In this replication, edits flow from the child replica to the parent replica. The issue described does not align with this setup.

Reference from Esri Documentation and Learning Resources:

Geodatabase Replication---ArcGIS Pro Documentation

Understanding One-Way Replication

Conclusion:

The issue occurs because One-way, parent to child replication is being used, where edits made on the child replica are overwritten by updates from the parent replica during synchronization.

Question 2

Question Type: MultipleChoice

A data owner creates a one-way replica parent-to-child for a single feature class to share data from a production geodatabase to a public-facing geodatabase.

- * The data owner synchronizes once a week to share updated data
- * In time, the data owner wants to add a new attribute field/field type and calculates new attribute values
- * The data owner synchronizes the replicas, but the new field and values are not present in the child replica
- * In the public-facing geodatabase, the data owner adds the same attribute field and field type
- * The data owner synchronizes the replicas again, and the values are not replicated in the child replica

How should the data owner resolve this issue?

Options:

- A- Unregister the replica pair?, run Enable Replica Tracking and Synchronize Change?
- B- Unregister the replica pairs, run Feature Compare and Synchronize Changes
- C- Unregister the replica pairs, recreate the replica, and Synchronize Changes

Answer:

C

Explanation:

Scenario Overview:

A one-way replica from parent to child geodatabase is created for a single feature class.

The data owner adds a new attribute field in the parent geodatabase, calculates values, and attempts to synchronize the replica.

The new field and its values do not appear in the child replica, even after manually adding the field to the child geodatabase.

Why Recreate the Replica?

The issue arises because schema changes (e.g., adding new fields) are not automatically propagated in one-way replication workflows. Synchronization only applies to data changes, not schema updates.

To ensure the schema changes are recognized, the replica pair must be recreated with the updated schema. (ArcGIS Documentation: Geodatabase Replication and Schema Changes)

Steps to Resolve the Issue:

Unregister the Replica: Remove the existing replica pair from both the parent and child geodatabases.

Recreate the Replica: Create a new one-way replica between the parent and child geodatabases. This new replica will include the updated schema.

Synchronize Changes: Perform synchronization to transfer data, including the new field and calculated values, to the child geodatabase.

Alternative Options:

Option A: Enabling replica tracking does not address schema synchronization and would not resolve the issue.

Option B: Running Feature Compare is helpful for analyzing schema differences but does not propagate schema changes.

Thus, the data owner must unregister the replica pairs, recreate the replica with the updated schema, and synchronize changes to resolve the issue.

Question 3

Question Type: MultipleChoice

A user accidentally deletes an enterprise geodatabase feature dataset.

Which technology should be used to resolve the issue?

Options:

- A- High availability
- B- Backup
- C- Archiving

Answer:

B

Explanation:

Understanding the Scenario:

An enterprise geodatabase feature dataset is accidentally deleted.

The organization needs to recover the dataset to its original state.

Available Technologies:

High Availability: High availability setups (e.g., failover systems) ensure continuous access to geodatabases during hardware or software failures. However, high availability does not restore accidentally deleted data.

Backup: A backup is a snapshot of the geodatabase taken at a specific point in time. It allows administrators to restore deleted datasets or recover from data loss scenarios.

Archiving: Archiving tracks historical edits in versioned geodatabases but does not provide recovery for accidentally deleted datasets.

Steps to Recover the Dataset:

Identify the most recent backup of the enterprise geodatabase.

Restore the geodatabase or extract the specific feature dataset from the backup.

Verify the restored data and synchronize it with ongoing updates if necessary.

Reference:

Esri Documentation: Backup and Restore.

Best Practices for Data Protection: Guidelines for implementing regular backups to prevent data loss.

Why the Correct Answer is B: A backup is the most reliable solution for recovering an accidentally deleted feature dataset. High availability ensures uptime but does not address data recovery, and archiving tracks edits rather than preserving entire datasets.

Question 4

Question Type: MultipleChoice

A GIS data administrator needs to restrict access to some fields in a feature class from unauthorized users.

Which method should the data administrator use?

Options:

- A- Layer file
- B- Query layer
- C- Database view

Answer:

C

Explanation:

To restrict access to specific fields in a feature class, the GIS data administrator should use a database view.

1. What is a Database View?

A database view is a virtual table created by a SQL query. It displays data from one or more tables or feature classes, but only the specified fields and rows are accessible.

Views allow the administrator to control the fields and records visible to users without altering the underlying data.

2. Why Use a Database View?

Provides fine-grained control over data access by limiting which fields (or rows) are visible to

specific users.

Helps enforce data security policies in multi-user environments.

Can be shared as a read-only layer or with restricted update permissions, depending on the use case.

3. Why Not Other Options?

Layer File:

A layer file (*.lyr) only defines how data is symbolized and displayed. It does not restrict field access at the database level. Unauthorized users can still access hidden fields through direct database connections.

Query Layer:

A query layer allows for custom SQL queries when displaying data in ArcGIS but is not a security measure. Users can modify or bypass the query to access all fields.

Steps to Create a Database View:

Use SQL to define the view, specifying only the required fields:

```
CREATE VIEW restricted_view AS
```

```
SELECT field1, field2
```

```
FROM feature_class
```

```
WHERE <condition>;
```

Grant permissions to the view for authorized users while restricting access to the base table.

Publish the view in ArcGIS as a read-only layer if needed.

Reference from Esri Documentation and Learning Resources:

Creating and Using Database Views

Data Security in Enterprise Geodatabases

Conclusion:

A database view is the most effective method to restrict access to specific fields in a feature class, ensuring data security while providing flexibility in data sharing.

Question 5

Question Type: MultipleChoice

After running a Compress, the GIS administrator needs to check if the Adds and Deletes tables for Buildings are empty before unregistering as versioned.

What should be referenced by the GIS administrator?

Options:

- A- sdelayers
- B- table_registry
- C- gdbitems

Answer:

A

Explanation:

To determine if the Adds and Deletes tables for the 'Buildings' dataset are empty before unregistering as versioned, the GIS administrator needs to reference the sde_layers table.

1. Purpose of the sde_layers Table

The sde_layers table tracks the relationship between base tables and the associated delta tables (Adds and Deletes).

For each versioned dataset, the sde_layers table contains entries linking the dataset to its corresponding A and D tables (e.g., A_<ObjectID> and D_<ObjectID>).

2. Steps to Verify Adds and Deletes

Identify the ObjectID of the Buildings dataset in the sde_layers table.

Query the Adds table (A_<ObjectID>) and Deletes table (D_<ObjectID>) associated with the Buildings dataset:

```
SELECT COUNT(*) FROM A_<ObjectID>;
```

```
SELECT COUNT(*) FROM D_<ObjectID>;
```

If both queries return 0, the Adds and Deletes tables are empty, and it is safe to unregister the dataset as versioned.

3. Why Not Other Options?

table_registry: This table tracks registered datasets but does not provide information about delta tables or their contents.

gdb_items: This table stores metadata for datasets but does not have details on delta table

contents.

Reference from Esri Documentation and Learning Resources:

Compressing a Geodatabase---ArcGIS Pro Documentation

Delta Tables in Versioned Geodatabases

Conclusion:

The GIS administrator must query the sde_layers table to verify the Adds and Deletes tables before unregistering the dataset as versioned.

Question 6

Question Type: MultipleChoice

ArcGIS Pro users must be able to use the Undo and Redo buttons while editing a dataset. At the same time, SQL users must be able to edit this dataset.

How should the ArcGIS data administrator configure this dataset?

Options:

- A- Nonversioned editing
- B- Traditional versioning
- C- Branch versioning

Answer:

B

Explanation:

Understanding the Scenario:

ArcGIS Pro users need Undo/Redo functionality, which is available in versioned workflows.

SQL users also need to edit the dataset, requiring direct access to the database tables.

These requirements point to a need for a versioning method that supports both ArcGIS client workflows and SQL-based edits.

Versioning Methods Overview:

Nonversioned Editing: Nonversioned editing allows direct editing of the database but does not support Undo/Redo functionality in ArcGIS Pro, making it unsuitable for this scenario.

Traditional Versioning:

Supports Undo/Redo functionality for ArcGIS Pro users.

Stores edits in delta tables (adds and deletes) to manage versions.

SQL users can access and edit the base tables, making it compatible with their needs.

Branch Versioning: Branch versioning supports modern workflows and web services but requires a service-based approach for editing. It does not allow direct SQL edits, making it unsuitable for this scenario.

Steps to Implement Traditional Versioning:

Enable traditional versioning on the dataset in the enterprise geodatabase.

Ensure appropriate permissions are set for SQL users to access and edit the base tables.

ArcGIS Pro users will work in the versioned environment, allowing Undo/Redo operations during their edits.

Reference:

Esri Documentation: Understanding Versioning.

Traditional Versioning Concepts: Best practices for using traditional versioning with multiple user types.

Why the Correct Answer is B: Traditional versioning fulfills both requirements: Undo/Redo functionality for ArcGIS Pro users and SQL accessibility for direct edits.

Question 7

Question Type: MultipleChoice

A GIS administrator creates a SQL command to update values in a feature class. In a test environment, the command is run against the feature class table. All the values do not seem to get updated.

Which configuration is causing this issue?

Options:

A- Nonversioned feature class that is partitioned

- B- Traditional versioned data with edits performed
- C- Archiving enabled on the feature class

Answer:

B

Explanation:

The issue arises because traditional versioned data stores edits in delta tables (Adds and Deletes) instead of the base table. SQL updates applied directly to the base table bypass the delta tables, resulting in incomplete or inconsistent updates.

1. How Traditional Versioning Affects Updates

In traditional versioning, edits are recorded in delta tables:

A_<ObjectID> (Adds): Tracks newly inserted rows.

D_<ObjectID> (Deletes): Tracks deleted rows.

When SQL commands are executed directly on the base table, they do not affect the data in the delta tables, which causes the feature class to reflect incomplete updates.

2. Why Not Other Options?

Nonversioned Feature Class that is Partitioned:

Partitioning organizes data for performance optimization but does not interfere with SQL commands updating the entire table.

Archiving Enabled on the Feature Class:

Archiving tracks historical changes in separate archive tables but does not directly impact SQL commands on the feature class.

Steps to Resolve the Issue:

For traditional versioned data, use the reconcile and post process to update values. This ensures that changes are correctly applied across delta tables and the base table.

Alternatively, use tools like ArcGIS Pro or ArcPy to programmatically update data instead of executing direct SQL commands.

Reference from Esri Documentation and Learning Resources:

Traditional Versioning Overview

Delta Tables and Traditional Versioning

Conclusion:

The issue occurs because the data is traditional versioned, and direct SQL commands do not account for the delta tables where edits are stored. Use the reconcile and post workflow or ArcGIS tools to apply updates correctly.

Question 8

Question Type: MultipleChoice

A GIS data manager needs to allow editors to see changes made to the default version in their child versions without reconciling.

How should the GIS data manager register the dataset?

Options:

- A- Traditional versioned with the option to move edits to base
- B- Branch versioned with global IDs and editor tracking enabled
- C- Traditional versioned with archiving and editor tracking enabled

Answer:

A

Explanation:

To allow editors to see changes made to the default version in their child versions without reconciling, the dataset must be registered as traditional versioned with the option to move edits to base.

1. How This Versioning Method Works

When registering data as traditional versioned with the option to move edits to base, changes made in the default version are immediately visible to all child versions without requiring reconciliation.

This method directly updates the base table for default version edits, bypassing the delta tables used in standard traditional versioning workflows.

2. Why Use This Method?

Immediate Visibility: Editors can see default version updates in their child versions without the need for reconciliation or posting.

Simplified Workflow: Reduces the overhead of managing version reconciliation and posting for

users who only need to view updates.

3. Why Not Other Options?

Branch Versioned with Global IDs and Editor Tracking Enabled:

Branch versioning is designed for feature services and web-based editing workflows. It does not allow direct visibility of default version changes in child versions.

Traditional Versioned with Archiving and Editor Tracking Enabled:

This method tracks historical changes but does not allow child versions to see updates in the default version without reconciliation.

Steps to Register Data with Move Edits to Base Option:

In ArcGIS Pro, right-click the dataset or feature class and select Manage > Register as Versioned.

Check the option 'Move edits to base' during registration.

Save the changes and verify that editors can view default version changes without reconciliation.

Reference from Esri Documentation and Learning Resources:

Traditional Versioning with Move Edits to Base

Conclusion:

Registering the dataset as traditional versioned with the option to move edits to base ensures that editors can see changes made to the default version in their child versions without requiring reconciliation.

Question 9

Question Type: MultipleChoice

A GIS administrator needs all users to immediately see the edits that are made while editing a feature class.

Which configuration should be used?

Options:

- A- Traditional versioned editing
- B- Nonversioned editing

C- Branch versioned editing

Answer:

B

Explanation:

Understanding the Scenario:

Edits made to a feature class need to be immediately visible to all users.

This indicates a requirement for direct edits to the base tables without any versioning overhead.

Editing Configuration Overview:

Traditional Versioned Editing: Edits are made to delta tables and require version reconciliation and posting to be visible to other users. This introduces a delay and does not meet the requirement for immediate visibility.

Nonversioned Editing: Edits are applied directly to the base table of the feature class, ensuring they are immediately visible to all users. This is the correct choice for the scenario.

Branch Versioned Editing: Edits are made in a branch version and are not immediately visible to other users unless the edits are saved to the default branch. This is not the best choice for immediate visibility.

Steps to Configure Nonversioned Editing:

Ensure the feature class is set to nonversioned editing mode in the enterprise geodatabase.

Grant necessary permissions for all users to edit the base table.

All edits made to the dataset will instantly reflect for all users.

Reference:

Esri Documentation: Nonversioned Editing.

Why the Correct Answer is B: Nonversioned editing is the only configuration that ensures immediate visibility of edits to all users. Traditional and branch versioning involve workflows that delay edit visibility.

To Get Premium Files for EGMP2201 Visit

<https://www.p2pexams.com/products/egmp2201>

For More Free Questions Visit

<https://www.p2pexams.com/esri/pdf/egmp2201>

20%
DISCOUNT

P2P
exams