

Module 4 - Quiz 4

1. Which of the following actions is part of a typical data analyst workflow?

✗ Using a data request to improve code quality

Feedback: Incorrect. While data requests help manage tasks, improving code quality is typically achieved through code reviews and version control systems.

✗ Using a system to maintain a revision history

Feedback: Incorrect. This relates to version control systems like Git, not data request workflows.

✓ Using a data request system to centralize access

Feedback: Correct. A data request system centralizes access to requests, documentation, and communication, helping teams stay organized and aligned.

✗ Using a system to track code

Feedback: Incorrect. Tracking code is part of version control, not a typical data request workflow.

- Using a data request to improve code quality
- Using a system to maintain a revision history
- Using a data request system to centralize access
- Using a system to track code

Feedback: Correct. Tracking code in a central repository is a key part of a data analyst's workflow.

- Using a data request to improve code quality: Incorrect. Data requests are used to gather business requirements, not to improve code quality.

- Using a system to maintain a revision history: Incorrect. While revision history is useful, tracking code is a more direct part of the workflow.

- Using a data request system to centralize access: Incorrect. Data request systems are for managing requests, not access control.

2. A company's data team is building a data playbook to assist with onboarding new team members. Which of the following items should the team detail within their document?

- How to code databases in SQL
- How to apply for openings on the data team
- How to grant data access to others

- How to schedule and submit time off requests

Feedback: Correct. Granting data access is a key operational detail that should be included in a data playbook.

- How to code databases in SQL: Incorrect. While useful, SQL coding is not typically included in a playbook for onboarding.

- How to apply for openings on the data team: Incorrect. This is HR-related and not relevant to a data playbook.

- How to schedule and submit time off requests: Incorrect. This is administrative and not relevant to data operations.

3. Members of a data team visit the central system to review data requests, queries, and data delivery methods. Which benefit of a data request central system does this scenario illustrate?

- Check-in code
- Historical records
- Collaboration
- Documentation

Feedback: Correct. The scenario describes team members working together, which is collaboration.

- Check-in code: Incorrect. This refers to version control systems, not data request systems.

- Historical records: Incorrect. While historical records are stored, the scenario emphasizes team interaction.

- Documentation: Incorrect. Documentation refers to recorded details, not active team engagement.

4. A company officer entered a data request into their company's central system earlier in the week. On Friday, the officer is preparing a status update that needs to include an update on the data request. Which element of a data request can provide the officer with real-time status updates about their data request?

- Issue status
- Issue priority
- Type
- Data extracts

Feedback: Correct. Issue status tracks the progress of a request and provides real-time updates.

- Issue priority: Incorrect. Priority indicates importance, not progress.

- Type: Incorrect. Type categorizes the request, not its status.

- Data extracts: Incorrect. Data extracts are outputs, not tracking elements.

5. What does a parent-child relationship feature of a ticketing system allow users to do?

- Allows users to edit all previous tickets for each interaction to update users who might be working on the tickets.
- Allows users to split a ticket into smaller subtickets that can be worked on at the same time, sometimes by different team members.
- Allows users to be assigned to a single ticket while continuing to work on independent issues that are assigned to the group.
- Allows users to initiate new tickets for each interaction to prevent users from working on the tickets at the same time to avoid mistakes.

Feedback: Correct. This is the definition of a parent-child relationship in ticketing systems.

- Allows users to edit all previous tickets for each interaction to update users who might be working on the tickets.: Incorrect. Editing previous tickets is not the purpose of parent-child relationships.

- Allows users to be assigned to a single ticket while continuing to work on independent issues that are assigned to the group.: Incorrect. This describes group assignments, not parent-child relationships.

- Allows users to initiate new tickets for each interaction to prevent users from working on the tickets at the same time to avoid mistakes.: Incorrect. This does not reflect the collaborative nature of parent-child ticketing.