

PRACTICE QUESTIONS

ServiceNow Certified Application Developer

SHARED BY [u/Grand-Ferret-8370](#) on aiexamcerts

EDITION Free Set 1

QUESTIONS 10

SHARED October 15, 2026

EXAM ServiceNow Certified Application Developer

Shared by u/Grand-Ferret-8370 on aiexamcerts

10 practice questions for the ServiceNow Certified Application Developer exam, each followed by its answer and a short explanation. More free sets and the full practice exam:

www.aiexamcerts.com/servicenow/servicenow-certified-application-developer. Spotted a mistake? Email support@aiexamcerts.com with the exam name and the question number.

1. Where should a setting such as a feature toggle be stored so that server scripts can read it with `gs.getProperty`?
 - A A dictionary entry on the incident table
 - B A system property record
 - C A choice list on the task table
 - D An application menu entry

Answer: B

A system property is a name-value record that scripts read with `gs.getProperty`, which suits settings such as feature toggles. A dictionary entry defines a field on a table, which is tempting when adding configuration, but it does not store values. A choice list holds option values for a field, and an application menu organizes navigation.

2. Server scripts read a system setting with `gs.getProperty`, and that setting is stored in the _____ table.

Answer: sys_properties (also accepted: sys_properties table, system properties)

System properties are stored in the `sys_properties` table and read through `gs.getProperty`. The `sys_user` table is tempting for per-user settings, but it holds user records rather than instance settings. The `sys_dictionary` table holds field definitions, not settings.

3. A reference field on a request form should offer only active users who belong to one group. Which feature limits the choices?
 - A A reference qualifier on the field's dictionary entry
 - B A UI policy that hides inactive records
 - C A business rule that filters the lookup after save
 - D A client script that deletes unwanted options

Answer: A

A reference qualifier filters which records the reference lookup offers, so it is the standard place to limit choices by condition. A UI policy changes field visibility or mandatory state, which is tempting for form behavior, but it does not filter the lookup. A business rule can change values during a save, but it does not shape the lookup a user sees.

4. A reference field stores the `sys_id` of the record it points to, rather than the displayed name.
- A True
 - B False

Answer: A (True)

A reference field stores the `sys_id` of the target record, which is why the displayed value can change while the link stays intact. Storing the display name would break whenever names change. The dictionary entry decides which table is referenced and how the lookup is shown.

5. Which two actions can a UI policy take without writing a script? (Choose two.)
- A Send an email to the requester
 - B Delete the related task
 - C Make a field mandatory
 - D Call an external REST endpoint
 - E Make a field read-only

Answer: C, E

UI policies can set fields to mandatory, read-only or visible based on conditions, all without code. Sending email and calling a REST endpoint need server logic that a UI policy cannot provide. Deleting a related task is a data change that belongs in a business rule.

6. A business rule must compare the values a record had before the current change with the new values. Which object holds the previous values?
- A current
 - B previous
 - C `g_form`
 - D `g_user`

Answer: B

`previous` holds the record's values before the change, so a rule can compare old and new values, such as detecting a state transition. `current` holds the values being written, which is tempting for comparisons, but it lacks the old values. `g_form` and `g_user` are client-side objects that do not exist in server business rules.

7. A client script must call a server method in a script include without reloading the page. Which pattern is standard?
- A A GlideRecord query placed directly in the client script
 - B A location reload triggered by the client script
 - C GlideAjax with a client-callable script include and a callback that reads the answer
 - D A UI action that redirects the user to another page

Answer: C

GlideAjax sends a request to a client-callable script include, and the callback reads the answer, so the browser stays on the page. A GlideRecord query in a client script is tempting, but the browser cannot query the database directly. Reloading the page or redirecting the user is slow and disruptive.

8. A UI policy changes how fields behave, such as making them mandatory or read-only, based on _____ that the developer defines, without writing a script.

Answer: conditions

UI policies apply actions such as mandatory, visible and read-only when their conditions are met, and they can run scripts when needed. Business rules run on the server, which is tempting for logic that feels similar, but they cannot change the form's appearance directly. Client scripts can do this with code, but a UI policy needs none.

9. An ACL script checks whether the current user's department matches the record's department. What does the ACL do when the script returns true?
- A It grants access for the operation the ACL controls
 - B It hides the record from every list view
 - C It sends an email to the administrator
 - D It grants access to every field on every table

Answer: A

When an ACL's condition or script returns true, access is granted for the operation it controls, so the rule opens access in that case. Granting every field on every table is an overreach that happens only when the ACL is written that way. A list view filter does not depend on the ACL script, and ACL scripts do not send email.

10. Which approach is recommended for granting access to a large group of users?
- A Assign roles directly to every individual user
 - B Assign roles to groups and add users to those groups
 - C Grant the admin role to all support staff
 - D Share one login across the team

Answer: B

Assigning roles to groups and adding users to those groups keeps access reviewable, and removing a user from a group revokes the inherited roles. Direct per-user grants are tempting for a quick fix, but they are hard to audit at scale. Granting admin to support staff is excessive, and shared logins remove accountability.