ITSM Interview Questions & Answers with ServiceNow – Manideep Arja

TOPICS: SLA, BUSINESS RULES, CLIENT SCRIPTS, UI POLICIES & ACLs.

1. **What is an SLA in ServiceNow?**

   Answer: An SLA (Service Level Agreement) defines the time frame for responding to or resolving incidents or requests. It ensures that the service provider meets agreed-upon performance standards.

2. **How do you create an SLA definition in ServiceNow?**

   Answer: You create an SLA definition by navigating to the SLA Definition table, defining the conditions, schedules, and targets for specific processes (like incidents or requests).

3. **Explain the difference between an SLA and an OLA (Operational Level Agreement).**

   Answer: An SLA is a commitment between the provider and the customer, while an OLA is an internal agreement between different parts of an organization to support the service delivery process.

4. **What is an SLA Schedule, and how is it configured?**

   Answer: An SLA Schedule defines the working hours for SLAs. You configure it by setting up the start time, end time, holidays, and working hours in the SLA Schedule table.

5. **What are SLA conditions, and how do they work?**

   Answer: SLA conditions define the triggers for when an SLA should start, pause, or stop. For example, an SLA might start when an incident is created and pause when waiting for customer input.

6. **How do you define the SLA start, stop, and pause conditions?**

   Answer: These conditions are defined in the SLA Definition record, where you specify conditions like "State is New" (start), "State is Awaiting User Info" (pause), and "State is Resolved" (stop).

7. **What happens when an SLA breaches in ServiceNow?**

   Answer: When an SLA breaches, and it is marked as "Breached." This may trigger actions such as notifications or escalations depending on your SLA rules.

8. **How can you calculate SLA compliance in ServiceNow**

   Answer: SLA compliance is calculated based on whether the SLA was completed within the target time defined in the SLA definition. The metric is usually tracked as “On Time” or “Breached.”

9. **What is the role of the SLA Definition table in ServiceNow**

Answer: The SLA Definition table holds the rules for defining SLAs, such as the conditions, targets, and schedules. It serves as a reference for generating SLA tasks.

10. **How do you handle escalations in SLAs in ServiceNow?**

Answer: You can handle escalations by using SLA Escalation rules, which automatically notify stakeholders or trigger actions when an SLA breaches or is close to breaching.

11. **What are the various types of SLA timers available in ServiceNow?**

Answer: The main SLA timers are:
Start Timer: Begins when the SLA is triggered.
Pause Timer: Stops the timer (e.g., when waiting for user input).
Stop Timer: Ends when the SLA condition is met (e.g., incident resolution).

12. **How do you pause an SLA timer and what triggers the pause?**

Answer: You can pause an SLA timer by setting conditions such as "State is Awaiting User Info." This ensures the timer stops until the condition is cleared.

13. **What is a Business Rule in ServiceNow?**

Answer: A Business Rule is a server-side script that executes when a record is inserted, updated, deleted, or queried.

14. **What are the types of Business Rules available in ServiceNow?**

Answer:
Before Business Rule: Executes before the record is saved.
After Business Rule: Executes after the record is saved.
Async Business Rule: Runs asynchronously after the record is saved.
Display Business Rule: Executes before the record is sent to the client.

15. **Explain the difference between a “Before” and “After” Business Rule.**

Answer: A "Before" rule executes before the record is saved, allowing modifications to the data, while an "After" rule executes after the data is saved and is used to trigger actions or updates.

16. **How would you prevent an update from happening in a Business Rule?**

Answer: You can prevent an update by using **current.setAbortAction(true);** in a Business Rule script to cancel the save or update.

17. **What is a “Display Business Rule” used for?**

Answer: A Display Business Rule is used to execute before the data is displayed to the user, allowing for pre-processing or modifying data for display purposes

.

18. **What is an “Async Business Rule,” and when should you use it?**

Answer: An Async Business Rule runs asynchronously after the record is saved, typically used for tasks like sending notifications or performing lengthy operations without affecting the user’s experience.

19. **How can you modify the behaviour of a Business Rule using scripts?**

Answer: You can modify behaviour by writing script logic using the current and previous objects in the Business Rule to check field values and apply changes accordingly.

20. **What happens if multiple Business Rules affect the same record at the same time?**

Answer: The rules will execute in the order defined in the "Order" field. Conflicts between rules may be resolved by adjusting the sequence or condition logic.

21. **Can you create a Business Rule to trigger an SLA?**

Answer: Yes, you can create a Business Rule to trigger an SLA by updating the SLA Task record or creating a new SLA task under specific conditions.

22. **What is the impact of a Business Rule on performance?**

Answer: Business Rules that execute on a large volume of records or complex logic can affect system performance, so it’s important to optimize them for efficiency.

23. **Can a Business Rule execute on a scheduled basis?**

Answer: Yes, a Business Rule can be combined with a Scheduled Job to execute at specific intervals or times.

24. **What is the significance of “Current” and “Previous” objects in Business Rules?**

Answer: The current object represents the record in its current state, while the previous object represents the record before it was modified.

25. **Can you cancel a Business Rule execution in a script?**

Answer: Yes, you can cancel a Business Rule by using current.setAbortAction(true); to prevent saving or updating the record.

26. **How do you use Business Rules to implement complex business logic in ServiceNow?**

Answer: Complex logic is implemented by writing scripts that evaluate multiple conditions, perform calculations, update related records, or integrate with external systems.

27. . **What are Client Scripts in ServiceNow, and where are they used?**

Answer: Client Scripts are scripts that execute on the client-side (user's browser). They are used for dynamic form behaviour, validation, and field interactions.

28. **What are the different types of Client Scripts available in ServiceNow?**

Answer: They are 4 Types
onLoad(): Executes when the form loads.
onChange(): Executes when a field value changes.
onSubmit(): Executes when the form is submitted.
onCellEdit(): Executes when a record is edited in a list.

29. **How does the onLoad Client Script work?**
Answer: The onLoad script runs when a form is loaded, allowing you to perform actions like setting default values or hiding fields based on conditions.

30. **What is an onChange Client Script and when would you use it?**
Answer: An onChange Client Script runs when a field's value changes. It can be used to dynamically change the value of other fields or validate inputs.

31. **How can you make a field read-only using a Client Script?**
Answer: You can make a field read-only by using `g_form.setReadOnly

32. **What is the purpose of the g_form object in Client Scripts?**
Answer: The g_form object is used to interact with form fields, including setting values, making fields visible or hidden, setting mandatory fields, and triggering form actions.

33. **How can you use Client Scripts to control field behavior in Service Catalog items?**
Answer: Client Scripts can control field behavior in Service Catalog items by modifying the g_form object to change field values, visibility, and validation rules based on user inputs.

34. **What is the difference between a Client Script and a UI Policy?**
Answer: A Client Script is a script that executes in the user's browser, whereas a UI Policy is a configuration rule that applies conditions to fields (e.g., making fields mandatory or visible) without coding.
35. **Can Client Scripts interact with other records in ServiceNow?**
Answer: No, Client Scripts are limited to the current form or record. For interacting with other records, you must use a Business Rule or GlideAjax.

36. **What are some best practices for writing Client Scripts?**
Answer: Best practices include minimizing script execution time, using g_form efficiently, and avoiding long-running synchronous code that can impact user experience.
37. **How would you create a dynamic pop-up or message in Client Scripts?**
Answer: You can create dynamic pop-ups using g_form.showFieldMsg() or browser-based JavaScript alert functions like alert().

38. **What is the purpose of the onSubmit Client Script?**
Answer: The onSubmit script executes when the form is submitted, allowing you to validate data before submitting or prevent submission if necessary.

39. **How would you pass data from a Client Script to the server side?**
Answer: Data can be passed from Client Scripts to the server using GlideAjax to call Script Includes and return results asynchronously.

40. **What are the key advantages of using Client Scripts?**
Answer: Client Scripts offer the advantage of real-time, interactive user experiences, enabling dynamic form behaviors, validations, and condition-based actions without server communication.
41. **How do you optimize Client Scripts for performance?**
Answer: Optimize Client Scripts by minimizing complex logic, reducing the number of calls to g_form functions, and avoiding excessive manipulation of form fields.
42. **What is a UI Policy in ServiceNow?**
Answer: A UI Policy that allows you to dynamically control the behavior of form fields, such as making them mandatory, visible, or read-only based on conditions.

43. **How do UI Policies differ from Business Rules?**
Answer: UI Policies are configuration-based and execute on the client-side, while Business Rules execute on the server-side and involve more complex logic.
44. **How can you make a field mandatory using a UI Policy?**
Answer: You can make a field mandatory by creating a UI Policy and setting the field's mandatory property to true based on the specified condition.

45. **Can UI Policies be triggered based on changes to other fields?**
Answer: Yes, UI Policies can trigger based on changes to other fields, and you can set conditions to evaluate and apply actions accordingly.
46. **What is the role of the "Run Script" option in UI Policies?**
Answer: The "Run Script" option allows you to add custom scripting logic within a UI Policy, giving you more control over the behavior of fields.

47. **What is a UI Action, and how is it different from a UI Policy?**
Answer: : A UI Action is a button, link, or menu item that performs actions when clicked. It differs from a UI Policy, which controls field behavior dynamically based on conditions.
48. **How can you hide a field using a UI Policy?**
Answer: You can hide a field by creating a UI Policy and setting the visible property of the field to false based on specific conditions.
49. **Can UI Policies be used to affect fields in a related list?**
Answer: No, UI Policies only affect fields on the form level, not fields in related lists. To manipulate related lists, you would need to use UI Actions or other scripting methods.
50. **How do you test a UI Policy in ServiceNow?**

Answer: You can test a UI Policy by creating test records and verifying if the field properties (like mandatory, visible, or read-only) are applied as expected.
51. **How does a UI Policy interact with Client Scripts?**

Answer: UI Policies typically handle field properties (visibility, read-only, mandatory), while Client Scripts can provide more complex functionality. Both can be used together, but they serve different purposes.
52. **What is an ACL (Access Control List) in ServiceNow?**

Answer: An ACL defines the permissions required to access or modify a record in ServiceNow. ACLs are applied to tables, fields, and records to manage data access.
53. **What types of ACLs exist in ServiceNow?**

Answer: Types of ACLs include:
Table ACLs: Control access to entire tables.
Field ACLs: Control access to individual fields.
Record ACLs: Control access to specific records.
54. **How is an ACL rule defined in ServiceNow?**

Answer: An ACL rule is defined by specifying the table, operation (read, write, create, delete), and the condition (who can access based on roles or other conditions).
55. **What is the role of roles in ACLs?**

Answer: Roles define what users or groups can access certain records, fields, or tables. An ACL checks the user's role to determine if access should be granted.

56. **How do you troubleshoot ACL access issues in ServiceNow?**

Answer: Troubleshooting ACL issues involves verifying user roles, checking ACL conditions, reviewing the "Access Control" logs, and using the "Test Access" feature in ACL records.

57. **What is the difference between a "Read" and "Write" ACL?**

Answer: A "Read" ACL controls whether a user can view a record or field, while a "Write" ACL controls whether the user can modify a record or field.

58. **How can you restrict access to a specific field using ACLs?**

Answer: You can create a Field ACL with the specific field name and define conditions based on user roles or other criteria to restrict access.

59. **Can you use scripted ACLs in ServiceNow?**

Answer: Yes, scripted ACLs can be used for more complex access control logic, allowing you to write scripts to determine whether access should be granted or denied.\

60. **What is a wildcard ACL in ServiceNow and how does it work?**

Answer: A wildcard ACL uses * to apply permissions more broadly. For example, incident.* applies to all operations (read, write, create, delete) on the incident table

61.What is table.None in ACLs and how does it affect access?

Answer: table.None refers to a record-level ACL that controls access to a table as a whole, not a specific field or operation. For example, incident.None controls whether the user has any access to the record at all. If this fails, no field- or operation-level ACLs (like read, write) will be evaluated.

62.What is the difference between *.field and table.* in ServiceNow ACLs? Which one is more specific?

Answer: *.field applies to all fields across all tables.
table.* applies to all operations (read, write, etc.) on a specific table.
table.* is more specific than *.field, because it targets a specific table instead of all tables.