Volume III

Hacknow

TOP 50 SERVICENOW DEVELOPER SCENARIO INTERVIEW QUESTION

(WITH ANSWERS!)

Crafted Specially for ServiceNow Professionals,

GET READY TO CRACK YOUR SERVICENOW INTERVIEW!

Crafted With Love ❤ by Team HackNow. Visit : https://hacknow.co.in

NTT DATA || IBM || MPHASIS

1. What are transform maps?

A transform map is a set of field maps that govern the relationships between fields in an import set (a staging table for imported data) and the fields in an existing ServiceNow target table. It defines how the raw data should be transformed and loaded into the target table during a data import.

2. Explain Types of transform map script with scenario.

Transform map scripts allow you to add advanced logic to the data import process by executing custom JavaScript at different stages. The main types of transform scripts are:

- **onStart**: Runs once at the beginning of the transformation, before any data rows are processed.
  - **Scenario:** You need to initialize a global variable or perform a one-time check before the import begins. For example, setting up a log entry to record the start of the import job.
- **onBefore**: Runs just before a source row is transformed into a target record.
  - **Scenario:** You want to modify or validate data from the source row before it's used to create or update a record. For instance, normalizing a street address, combining first and last names, or skipping a record if certain conditions are not met.
- **onAfter**: Runs after a record has been transformed and saved to the target table.
  - **Scenario:** You need to perform a follow-up action based on the newly created or updated record. For example, creating a related record in another table, or sending a notification to the user who was just updated.
- **onComplete**: Runs once after the entire transformation process is finished.
  - **Scenario:** You want to perform a final action after all records have been processed. For example, sending a summary email to the import

administrator or updating a record count in a status table.

- **`onForeignInsert`:** Runs before a new reference record is created in a different table (from the one being transformed to).
    - **Scenario:** You are importing a list of incidents that reference a "caller" who does not exist in the `sys_user` table yet. The script will run before the new user is created in the `sys_user` table.
- **`onChoiceCreate`:** Runs before a new choice list entry is created.
    - **Scenario:** You import data with a value for a choice list field that does not exist. The script runs before the new choice is created, giving you a chance to alter the value or prevent its creation.
- **`onReject`:** Runs if a foreign record or choice creation fails.
    - **Scenario:** Used in conjunction with `onForeignInsert` or `onChoiceCreate` to handle situations where a new referenced record cannot be created.

3. What are coalesce? Can we make more than one field as coalesce true?

Coalesce is a transform map option that specifies one or more fields as a unique key for the import. It determines whether a record from the import set should be used to:

- **Update** an existing record in the target table (if a matching record is found).
- **Insert** a new record in the target table (if no match is found).

Yes, you can make more than one field as coalesce true. This is called **multi-field coalescing**. For an update to occur, the values of **all** fields marked as coalesce must match an existing record in the target table.

4. Ways we can import data into the ServiceNow?

You can import data into ServiceNow using several methods:

- **Import Sets:** A core ServiceNow feature used to load data from an external source into a staging table before transforming it into a target table.

- **Data Sources:** Integrates with various external sources such as files (CSV, XML, Excel), JDBC databases, FTP, and HTTP.

- **REST/SOAP APIs:** For programmatic imports from external systems.

- **Integrations:** Including specialized integrations for specific systems like LDAP and SCCM.

5. What are data source?

In ServiceNow, a **data source** is a record that contains the details about where data is being imported from. It defines the connection and location of the source data, such as a file on an FTP server, an external database via JDBC, or a file uploaded from a local machine. A data source is a prerequisite for using Import Sets.

6. Ways to migrate data from one instance to another?

Data migration between ServiceNow instances can be accomplished using several methods:

- **Export/Import (XML):** For smaller-scale migrations, records can be exported to an XML file from the source instance and imported into the target instance. This method can be complex due to record dependencies.

- **Update Sets:** Used to migrate configurations, not raw data. For example, moving a custom application or a business rule between instances.

- **IntegrationHub ETL:** A modern and robust tool for extract, transform, and load operations, particularly for CMDB data migration.

- **ServiceNow Restore:** A service provided by ServiceNow support to copy an entire instance backup from one company's instance to another.

- **Custom Integrations (REST/SOAP):** Using APIs to programmatically pull and push data between instances.

7. What are REST API?

A **REST API** (Representational State Transfer API) is an architectural style for an API that uses standard HTTP methods to interact with resources. It is a set of rules for building web services that are lightweight, scalable, and stateless, and they commonly use JSON or XML data formats.

8. methods used in REST.

REST APIs utilize standard HTTP methods to perform CRUD (Create, Read, Update, Delete) operations on resources:

- **GET :** Retrieves a representation of a resource. (Read)
- **POST :** Creates a new resource. (Create)
- **PUT :** Replaces an entire resource with new data. (Update/Replace)
- **PATCH :** Applies a partial update to a resource. (Partial Update)
- **DELETE :** Deletes a resource. (Delete)

9. Difference between REST and SOAP? Which is more preferable?

Feature	REST (Representational State Transfer)	SOAP (Simple Object Access Protocol)
Type	Architectural style, not a protocol.	A protocol with a strict set of standards.
Data Format	Lightweight and supports multiple formats, including JSON, plain text, and XML.	Relies exclusively on XML for messaging.
Transport	Primarily uses HTTP/HTTPS.	Can use a variety of transport protocols like HTTP, SMTP, and TCP.
Performance	Faster and more efficient due to smaller messages and caching capabilities.	Slower due to larger, XML-based messages and overhead.
Security	Supports security through HTTPS.	Has its own standard (WS-Security) for encryption and security, often with more overhead.
State	Stateless; each request is independent of previous requests.	Can be stateful, requiring the server to store context from previous interactions.

Which is more preferable?

REST is generally more preferable for modern web services due to its flexibility, efficiency, and lighter-weight nature, which makes it ideal for public APIs, mobile apps, and microservices. However, SOAP may be better for legacy enterprise applications that require a higher level of security, transaction-based reliability, and strict data consistency (ACID compliance).

10. What are inbound and outbound integration?

- **Inbound Integration:** An external system sends data to or initiates an action within ServiceNow.
  - **Example:** An external monitoring tool creates an incident record in ServiceNow via a REST API call.
- **Outbound Integration:** ServiceNow sends data to or initiates an action in an external system.
  - **Example:** A business rule in ServiceNow sends a notification to a third-party application when an incident is updated.

11. What is LDAP integration? Explain the setup.

LDAP (Lightweight Directory Access Protocol) integration is used to connect a ServiceNow instance to an LDAP directory (e.g., Active Directory) for user data and authentication. This enables users to log into ServiceNow with their LDAP credentials and automates the creation and management of user accounts.

The setup typically involves these steps:

1. **Configure the LDAP Server record:** Navigate to **System LDAP > LDAP Servers** and create a new record. Provide connection details like the server URL, port (389 for LDAP, 636 for LDAPS), and authentication credentials (Bind DN and password).
2. **Define the OU Definitions:** Specify the organizational units (OUs) that you want to import data from. This sets the search base for queries.
3. **Set up the LDAP Server configuration:** This defines the import source. You can define what information to retrieve and how it maps to your users and groups.
4. **Create an Import Set and Transform Map:** A transform map is used to map fields from the LDAP source to the target user and group tables in ServiceNow (`sys_user` and `sys_user_group`).
5. **Schedule the Import:** Schedule the LDAP import to run periodically to keep user and group data synchronized.

6. **Set up Authentication:** Configure multi-provider SSO to enable login via LDAP and set the order of authentication methods.

12. Ways we can create notifications?

You can create notifications in ServiceNow using multiple methods:

- **Notification Records:** The most common method, created via **System Notification > Email > Notifications**. This record defines the email content, recipients, and when to send it (triggered by an event or a record update).

- **Flow Designer:** Use the **Flow Designer** to create powerful, codeless notifications as part of a larger process. You can use the "Send Email" or "Create Task" actions.

- **Business Rules:** Asynchronous Business Rules can trigger an event, which in turn sends a notification.

- **Scripting:** Advanced users can create notifications directly using server-side scripts (e.g., in a Business Rule or Script Include).

- **Connect Chat and Connect Support:** These modules can also be configured to trigger notifications.

- **Flows or Subflows:** Within a flow, a `Send email` action can be added to deliver a notification.

13. What is SCCM integrations?

SCCM (System Center Configuration Manager) integration involves integrating ServiceNow with a Microsoft SCCM database to import computer and software inventory data. This populates the ServiceNow CMDB and provides accurate and up-to-date asset information.

The integration typically works by:

- Connecting to the SCCM database via a data source.

- Importing SCCM data (workstations, installed software, etc.) into import set tables.

- Using transform maps to populate the CMDB `cmdb_ci_computer` and related tables.

14. What is the base table of CMDB?

The base table for the Configuration Management Database (CMDB) in ServiceNow is `cmdb_ci`. This table stores the fundamental attributes for all Configuration Items (CIs), and more specific tables (e.g., `cmdb_ci_server`, `cmdb_ci_computer`) extend from it.

15. Explain 5 best practices of scripting.

1. **Use Script Includes for server-side logic:** Instead of putting complex scripts directly in Business Rules, use Script Includes. This promotes code reuse, makes the code easier to maintain, and helps with debugging.

2. **Use Asynchronous GlideAjax and Display Business Rules:** For client-side lookups, avoid synchronous `GlideRecord` queries and `getReference()`, which can freeze the user's browser. Instead, use asynchronous `GlideAjax` or Display Business Rules to pass data to the client using `g_scratchpad` for improved performance.

3. **Use meaningful variable and function names:** Use descriptive names that clearly explain the variable's purpose. This makes code easier to read and maintain for yourself and others.

4. **Add comments to explain complex logic:** Even with clean code, certain parts of the script may need clarification. Add comments to explain why a particular piece of code was written or to describe complex functions.

5. **Use appropriate APIs and avoid deprecated ones:** Stay up-to-date with ServiceNow's recommended APIs and avoid using outdated or inefficient methods. For example, use GlideForm instead of `document.getElementById` to interact with form fields.

16. Difference between synchronous and Asynchronous?

Feature	Synchronous	Asynchronous
Execution	Processes tasks sequentially. The next step does not start until the previous one is completed.	Processes tasks independently. It runs in the background and does not block the user or process from continuing with other tasks.
User Experience	Can cause the user interface to freeze while waiting for a task to complete, especially with long-running operations.	Provides a more responsive user experience as the interface remains interactive while background tasks run.
Timing	Occurs in real-time, with immediate feedback.	Involves a time delay between sending a request and receiving a response.
Best For	Operations where a value is needed immediately for subsequent processing.	Operations that are resource-intensive or can be handled in the background, like heavy integrations or bulk updates.

17. Difference between GlideAjax, GetReference and Display BR?

Feature	GlideAjax	GetReference()	Display Business Rule (with `g_scratchpad`)
Execution	Asynchronous. A client-side script calls a server-side Script Include and waits for the server response without freezing the browser.	Synchronous. Makes a server call and waits for the entire reference record to be retrieved before the script continues. This can cause performance issues.	Synchronous (on server), but data is pushed to the client when the form loads, making it immediately available to client scripts.
Data Retrieval	Efficient. Retrieves only the specific data fields requested by the client script.	Inefficient. Retrieves the entire record from the server, including all fields, which can impact performance.	Efficient. The Display BR can query the server and place specific data fields into `g_scratchpad` for the client to use when the form loads.
Use Case	Fetching data from the server in response to user actions (e.g., field changes) without a full page refresh.	Avoid using this method due to performance degradation.	Fetching data from the server when a form loads to make it available for multiple client-side functions without making multiple server calls.

18. Difference between After and Async BR?

Feature	After Business Rule	Async Business Rule
Execution	Synchronous. Runs immediately after a record has been inserted, updated, or deleted in the database.	Asynchronous. Runs in the background as a scheduled job after a record has been committed to the database.
Timing	The current transaction waits for the After BR to complete.	The current transaction finishes, and control is returned to the user immediately. The Async BR is queued for execution later.
Best For	Actions that must happen immediately following a database operation and that do not cause significant performance delays (e.g., updating a related field).	Long-running operations that do not need to be executed immediately, such as heavy calculations or API calls.

19. What is a Client Script? Types?

A **client script** is JavaScript that runs in a user's web browser rather than on the server. It is used to manage form interactions, perform real-time field validation, and enhance the user experience.

The main types of client scripts are:

- **onLoad()**: Executes when a form or record loads.
- **onChange()**: Executes when the value of a specific form field changes.
- **onSubmit()**: Executes when a user submits a form.
- **onCellEdit()**: Executes when a user modifies a cell in a list view.

20. Write the syntax of GlideAjax.

Here is an example of the syntax for a GlideAjax call.

1. Create a Script Include (server-side):

```
var MyAjax = Class.create();
MyAjax.prototype =
Object.extendsObject(AbstractAjaxProcessor, {
   myFunction: function() {
      var myParam = this.getParameter('sysparm_my_param');
      // Do some work on the server
      var myResult = 'Hello ' + myParam;
      return myResult;
   },
   type: 'MyAjax'
});
```

Use code with caution.

2. Call it from a Client Script (client-side):

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
   if (isLoading || newValue === '') {
      return;
   }

   var ga = new GlideAjax('MyAjax'); // Name of the Script Include
   ga.addParam('sysparm_name', 'myFunction'); // Name of the function in the Script Include
   ga.addParam('sysparm_my_param', 'World'); // Custom parameter to send to the server

   ga.getXML(myCallback); // Get the response and call the callback function
}

function myCallback(response) {
   var answer =
response.responseXML.documentElement.getAttribute('answer');
   g_form.addInfoMessage(answer); // Use the response
}
```

Use code with caution.

🚀 Struggling with **ServiceNow interview prep or scripting**? You're not alone.

While many professionals charge just to talk for 30 minutes, **HackNow** is different.

Our mission is simple: **serve people first.** That's why your consultation is 100% free.

✅ Learn. ✅ Practice. ✅ Grow with the right guidance.

👉 Book your free consultation today

From Nothing To Now ▶ It's HackNow

21. What is approval coordinator? kind of approval can we add in it?

An **Approval Coordinator** is a component within a Flow Designer flow that can trigger multiple types of approvals in a coordinated fashion. It allows you to define approval logic for a group of approvers rather than just a single approver, and it can collect and process multiple approval decisions.

You can add the following types of approvals in an Approval Coordinator:

- **User Approvals:** Approvals from specific users.
- **Group Approvals:** Approvals from specific groups.
- **Scripted Approvals:** Approvals based on complex conditions defined in a script.

22. What are the ways to trigger a notification? Can we trigger a notification through scheduled job?

Notifications can be triggered in ServiceNow through various methods:

- **Events:** The most common method. An event is fired by a Business Rule, Workflow, or Script Include, and the notification is sent when the event is processed.
- **Directly on a record update:** A notification can be configured to trigger when a record on a specific table is inserted, updated, or deleted, and certain conditions are met.
- **Flow Designer:** Flows can be configured to send a notification at any point in the process.
- **Scripting:** Using the `gs.eventQueue()` function to trigger an event.

Yes, you can trigger a notification through a scheduled job. A scheduled job can run a script that calls the `gs.eventQueue()` function, which in turn sends a notification. This is useful for sending periodic or summary reports.

23. Can we trigger a notification through business rule?

Yes, you can trigger a notification through a Business Rule. The best practice is for the Business Rule to trigger an event using the `gs.eventQueue()` function, and then a Notification record is configured to listen for that event. This decouples the Business Rule from the Notification, making it more modular and reusable.

24. A user is unable to see a field on the form, how do you troubleshoot?

Here are the steps to troubleshoot why a user cannot see a field on a form:

1. **Check the Form Layout:** Right-click the form header (**Configure > Form Layout**) to verify that the field is on the form.

2. **Check Field-Level ACLs:** Check the Access Control Lists (ACLs) for the field. Navigate to **System Security > Access Control (ACL)**, search for ACLs on the specific table and field, and ensure the user has the required roles for the "read" operation.

3. **Check UI Policies:** Check for UI Policies that might be hiding the field. Look for UI Policies on the table where the field exists and examine the "UI Policy Actions" to see if any are setting the field's visibility to `false`.

4. **Check Client Scripts:** Look for Client Scripts (onLoad) that might be hiding the field using `g_form.setVisible('field_name', false)`.

5. **Check Dictionary Entry:** Check the dictionary entry of the field (**Configure > Dictionary**) to ensure the `Active` attribute is `true`. You can also check for an `ACL` section at the bottom of the dictionary entry.

6. **Test with a different user:** Try impersonating another user with similar permissions to see if the issue is user-specific.

7. **Check View:** Ensure the user is viewing the correct form view, as different views can have different layouts.

25. SLA is not triggering on an incident, what steps will you take?

To troubleshoot an SLA not triggering on an incident:

1. **Check the SLA Definition:** Verify the SLA definition for the correct table (`incident`), ensuring it's active.
2. **Check the Conditions:** Examine the "Start conditions" on the SLA Definition. Do the conditions match the incident record? For example, does the incident have the correct priority, category, or assignment group?
3. **Check the Pause Conditions:** Check if any "Pause conditions" are met that might be preventing the SLA from starting or are pausing it immediately.
4. **Check the Stop Conditions:** Review the "Stop conditions" to ensure the SLA is not immediately stopping upon creation.
5. **Check the Attach to:** In the SLA definition, make sure the "Attach to" field is configured correctly to match the incident record.
6. **Look for other SLAs:** Check if a higher-priority SLA is being attached instead. The SLA engine prioritizes SLAs based on a defined order.
7. **Run the SLA Debugger:** Use the **SLA Debugger** (**Right-click > Show SLA Trace**) on the incident record to see the system's decision-making process for attaching SLAs.
8. **Manually Trigger the SLA:** You can use a background script in a sub-production instance to test if the SLA will attach given the record's current state.

26. Client script is not working as expected, how will you debug it?

Debugging a client script involves several steps:

1. **Browser Developer Tools:** The primary method. Press `F12` and use the console to look for errors. You can also use breakpoints in the "Sources" tab to step through the code line-by-line.

2. **Alerts:** Add `alert()` statements to the script to check if it's running and to display the value of variables at different points. This is an older method but can still be useful for simple debugging.

3. **`g_form.addInfoMessage()`:** Use this function to print variable values and messages to the top of the form, which is less disruptive than `alert()`.

4. **`console.log()`:** Use `console.log()` to print messages and variable values to the browser console. This is the recommended method for non-disruptive debugging.

5. **Check `UI Policy` and `ACLs`:** Ensure no conflicting UI Policies or field-level ACLs are interfering with your script's actions.

6. **Isolate the code:** Comment out portions of the script to determine which line is causing the unexpected behavior.

27. How do you create multi-level approval in a flow?

To create a multi-level approval in Flow Designer:

1. **Add an "Ask For Approval" Action:** Drag and drop the "Ask For Approval" action into your flow.

2. **Configure the Approval:** In the action properties, specify the record to be approved. You can select specific users or groups to be the approvers.

3. **Use an "Approval Coordinator" Action:** For complex, multi-layered approvals, use the "Approval Coordinator" action. This allows you to define multiple stages of approval.

4. **Chain Multiple Approval Actions:** For a simple multi-level process, you can simply add multiple "Ask For Approval" actions in sequence.

5. **Utilize Conditional Logic:** After each approval action, use conditional logic (`if/then` branches) to check the outcome of the approval (e.g., `if approved > do this`, `if rejected > do that`).

6. **Use Subflows:** For complex or reusable approval processes, create a subflow specifically for the approval and call it from the main flow.

28. How do you debug scripts in ServiceNow?

Different methods are used to debug scripts in ServiceNow, depending on whether they are client-side or server-side:

Client-Side Scripts:

- **Browser Developer Tools (F12):** Use the `Console` to check for errors and the `Sources` tab to set breakpoints.
- **`console.log()`:** Use this function to write messages and variable values to the browser's console.
- **`g_form.addInfoMessage()`:** Display messages on the form.

Server-Side Scripts:

- **`gs.log()` and `gs.info()`:** These functions write messages to the system log, which can be viewed under **System Logs > System Log**.
- **Script Debugger:** The Script Debugger, available in Studio, allows you to set breakpoints in server-side scripts (e.g., Business Rules, Script Includes) and step through the code execution.
- **Background Scripts:** Run code snippets in a sub-production instance to test specific functionality.
- **Execution Plan:** For workflows and flows, check the execution plan to see the status of each step and identify where it failed.

29. What are SLAs in ServiceNow and how do you configure them?

SLAs (Service Level Agreements) are records in ServiceNow that define the expected timeframes for completing a task, such as an Incident or Service Request. They ensure that IT teams meet specific performance metrics.

To configure an SLA:

1. **Create an SLA Definition:** Navigate to **Service Level Management > SLA > SLA Definitions** and create a new record.
2. **Define Conditions:** Set the "Start," "Pause," and "Stop" conditions based on field values on the task record (e.g., Start when priority is 1, Pause when state is "On Hold", Stop when state is "Closed").
3. **Set Duration:** Define the target time for the SLA (e.g., 8 business hours for resolution).
4. **Specify Schedule and Timezone:** Define the working hours to be used for the SLA calculation.
5. **Create Escalation Actions:** Define actions to be taken when the SLA reaches specific thresholds (e.g., sending an alert when it's 75% complete).

30. Write a script to create a problem ticket from an incident.

This script would typically be placed in a Business Rule on the Incident table.

```
// This script would be run as an After Business Rule on the Incident table.
// Ensure the condition is met (e.g., "Problem" field changes to "Create New Problem").

var prob = new GlideRecord('problem');
prob.initialize();
prob.short_description = current.short_description;
prob.description = current.description;
prob.priority = current.priority;
prob.company = current.company;
prob.assignment_group = current.assignment_group;
prob.assigned_to = current.assigned_to;
var sysID = prob.insert();

// Relate the problem back to the incident
current.problem_id = sysID;
current.update();
```

```
// Add a work note to the incident
var msg = 'Problem ' + prob.number + ' created from this incident.';
current.work_notes = msg;
current.update();
```

31. What is the difference between standard and emergency change?

- **Standard Change:** Pre-authorized, low-risk, and routine changes that follow a standardized process and are often automated. They don't require an approval board and follow a set schedule.
- **Emergency Change:** High-impact, urgent changes that must be implemented as quickly as possible to resolve a major incident or critical issue. They bypass the normal approval process but still require an emergency approval, often from a senior manager or designated group.

32. How do you customize ITSM forms and fields?

- **Customize Forms:**
  - **Form Layout:** Use the Form Layout editor to add, remove, and arrange fields on a form.
  - **UI Policies:** Control the behavior of form fields, such as making them visible, mandatory, or read-only, without scripting.
  - **Client Scripts:** Use JavaScript to control form behavior based on user actions.
  - **UI Actions:** Create buttons, links, or context menu items to trigger specific actions.
- **Customize Fields:**
  - **Dictionary Entry:** Modify a field's properties, such as its name, type, and mandatory status.
  - **Dictionary Overrides:** For tables that extend a parent table, you can override dictionary settings on the child table.

33. What is the use of import sets?

Import sets are used to import data from external sources into ServiceNow. They serve as a staging area where raw data is temporarily held before being transformed and inserted into a target table.

34. What parameters we can pass in `gs.eventQueue`?

The `gs.eventQueue()` function takes up to five parameters:

1. **`event_name` (String):** The name of the event to be fired.

2. **`target_record` (GlideRecord):** The record associated with the event.

3. **`parm1` (String):** An optional string to be passed to the event. This is often the user's `sys_id` or a custom message.

4. **`parm2` (String):** Another optional string parameter.

5. **`queue_name` (String):** The name of the queue to process the event.

35. What are scheduled jobs?

Scheduled jobs are automated scripts that execute at a predefined time or interval. They can be used for various tasks, such as generating reports, cleaning up old data, or triggering events for notifications.

36. What are inbound email Actions?

Inbound Email Actions process incoming emails sent to your ServiceNow instance. They check the email's content, determine its type (e.g., reply, new incident), and perform a specific action, such as updating an existing record or creating a new one.

🚨 ***"I failed my first ServiceNow interview… not because I didn't know ServiceNow — but because I couldn't explain my debugging approach."***

That's what one of our learners told us before joining **HackNow Scripting Season 3**

He knew the platform.
He could configure flows.
But when the interviewer asked:

👉 *"What would you do if your scheduled job caused duplicate records in production?"*

He froze.

💡 After 3 weeks of **scenario-based prep** (mock interviews + HackNow scripting guides), he not only answered confidently — he cracked an offer at a **global consulting firm**.

🎯 Thinking about learning ServiceNow? Want to go from *beginner to pro*?
At **HackNow**, we've guided **700+ candidates** with a success rate of **87%** — and counting.

👉 It's not just about learning ServiceNow. **It's about building the confidence to solve real-world problems.**

👉 Book your free consultation today

From Nothing To Now ▶ It's HackNow

36 (Duplicate). What were your responsibilities in ServiceNow upgrade?

This answer requires personal experience, but common responsibilities include:

- **Reviewing Release Notes:** Identifying new features, deprecations, and potential impacts.
- **Testing:** Thoroughly testing existing functionality, custom code, integrations, and reports in a sub-production instance after the upgrade.
- **Fixing Bugs:** Addressing any issues or bugs discovered during testing, ensuring functionality is restored.
- **Regression Testing:** Verifying that existing functionality continues to work as expected.
- **Communication:** Notifying stakeholders and users about the upgrade schedule and expected changes.
- **Monitoring:** Observing system performance and stability after the upgrade to ensure a smooth transition.

37. Bugs found in the latest ServiceNow Release? How did you fix them?

This also requires personal experience. An example response might be:

"In the *Vancouver* release, we discovered a bug with a custom UI Action. It was a client-side script that used DOM manipulation, which is no longer supported. The fix involved rewriting the script to use the standard `g_form` API. Another issue involved a Business Rule that was taking too long to execute due to inefficient GlideRecord queries. We optimized the query by adding `setLimit()` and appropriate `addQuery()` clauses to improve performance."

38. Explain the SSO integration

SSO (Single Sign-On) integration allows users to log into ServiceNow using the same credentials they use for other enterprise applications. This is typically achieved using SAML (Security Assertion Markup Language).

SSO Integration Process:

1. **Configure IdP:** Set up a ServiceNow application in your Identity Provider (IdP) (e.g., Okta, ADFS) with the necessary metadata.

2. **Install SAML 2.0 Plugin:** Activate the `Integration - Multiple Provider Single Sign-On` plugin in ServiceNow.

3. **Create Identity Provider Record:** Create a new IdP record in ServiceNow and import the IdP metadata.

4. **Configure SAML Settings:** Define parameters like user ID and login/logout URLs.

5. **Test the Integration:** Use the built-in test function to verify that SAML is working correctly.

6. **Enable SSO:** Once tested, enable the SSO integration to redirect users to the IdP for authentication.

39. How do you customize the Related tab on the form?

The "Related" tab is a related list. You can customize related lists by:

1. **Configure Related List:** Right-click the header of the related list and select **Configure > List Control**. This allows you to set filters, controls, and conditions.

2. **Update Table Relationships:** Navigate to **System Definition > Relationships** to create or modify relationships between tables, which will then appear as related lists.

3. **Write Scripts:** Use a Script Include and a `setQuery` attribute on the related list to filter the records displayed dynamically.

40. Difference between query BR and ACL? Which one executes first?

Aspect	Query Business Rule	Access Control List (ACL)
Purpose	Filters the records returned by a query before they are displayed to the user. It controls which records are retrieved from the database and seen in lists, reports, or reference fields.	Restricts access to data based on roles, conditions, and scripts. ACLs can control access to records (rows) and specific fields (columns) and support CRUD operations (Create, Read, Update, Delete).
User Impact	The user will not be aware that records have been excluded, leading to a "cleaner" user experience. The system doesn't display a "rows removed by security constraints" message.	The user may see a message at the bottom of a list indicating that "rows have been removed by security constraints".
Performance	Can be more performant than script-based read ACLs, especially on large tables, because the filtering is applied at the database query level. This reduces the number of records retrieved from the server.	Script-based ACLs can cause performance issues on large tables because the script needs to run for each row returned. ACLs are cached, but business rules are not, so ACLs are generally preferred for performance reasons when possible.
Flexibility	Less granular than ACLs. Only controls record-level read access.	More granular and flexible, as they can enforce security on specific fields and for different operations (create, write, delete).

Execution order: The **Query Business Rule executes first**. Before a database query returns results, any active "before query" business rules are executed. Only after the Query Business Rule has filtered the result set are the ACLs evaluated for the remaining records.

41. Types of Script Includes

There are three primary types of Script Includes:

- **On-demand/Classless:** A collection of reusable functions not tied to a specific class. They are useful for grouping related, reusable logic, like utility functions. They are not extensible.

- **Extend an existing class:** A Script Include that extends an existing ServiceNow class using the `extendsObject()` method. This allows you to add custom functions to existing ServiceNow classes while inheriting all their base functionalities.

- **Define a new class:** A Script Include that creates a new, reusable server-side class, often following a standard naming convention like `Utils` (e.g., `UserUtils`). This approach is useful for object-oriented programming where a set of related properties and methods are grouped together. The `initialize` function is automatically called when the class is instantiated.

42. How can we call one Script Include from another Script Include?

To call a Script Include from another Script Include, you must use the `new` keyword to instantiate the target Script Include and then call its methods. The syntax depends on the scope of the Script Include.

- **Within the same scope:** You can directly instantiate the class.

  ```
  var myUtils = new MyUtils();
  var result = myUtils.myFunction();
  ```
  Use code with caution.
- **Across different scopes:** You must prepend the Script Include's name with its unique scope namespace.

```
// Calling a Script Include from the 'x_cld_travel' scope from another script
var travelScheduler = new x_cld_travel.ItineraryConflict();
var conflict = travelScheduler.checkConflict();
```

Use code with caution.

- **From a scoped application calling a global Script Include:**

```
var globalUtils = new global.ArraysUtils();
var array = globalUtils.getArray();
```

Use code with caution.

43. How to debug the script?

Debugging in ServiceNow can be done using a variety of methods:

- **Script Debugger (Server-side):** The primary tool for debugging synchronous server-side scripts like Business Rules and Script Includes.
  - Set breakpoints to pause script execution at specific lines.
  - Step through code line-by-line.
  - Inspect variables and the call stack.
  - The debugger is session-based and available to users with the `admin` or `script_debugger` role.
- **Session Log (Server-side):** Provides a log of executed business rules and other background transactions for the current user's session.
- **`gs.info()`, `gs.warn()`, `gs.error()`:** Use these server-side logging functions to print messages to the system log, helping you trace script execution and variable values.
- **`jslog()` (Client-side):** A client-side method for writing messages to the browser's JavaScript console.

- **Browser Developer Tools:** For client-side scripts, use the browser's built-in developer tools to set breakpoints, inspect variables, and step through code.
- **Field Watcher (Server-side):** A tool to track field changes on a form, which is helpful for debugging why a field's value is changing unexpectedly.
- **Automated Test Framework (ATF):** Run scripted tests to validate expected behavior, which can include debugging steps.

44. Have you used Flow Designer?

Flow Designer is a tool for automating business processes in a low-code/no-code environment. If you have experience with it, you would discuss:

- **Core Concepts:** Using flows, triggers, actions, and subflows to create and manage multi-step processes.
- **Benefits:** Mention ease of use, reduced need for scripting, reusability of components, and improved efficiency.
- **Use Cases:** Automating approvals, sending notifications, creating tasks, and integrating with external systems using IntegrationHub.
- **Experience:** Provide specific examples of flows you have built, such as onboarding new employees, automating approval chains for catalog items, or integrating with a third-party application.

45. Name some Client and Server Side API.

Client-Side APIs:

- **`g_form`:** Used to customize forms by getting/setting field values, hiding/showing fields, and adding messages.
- **`g_user`:** Provides access to information about the current user, such as their name, ID, and roles.
- **`GlideAjax`:** Allows client-side scripts to make asynchronous calls to server-side Script Includes.
- **`g_list`:** Used to customize lists in the UI.

- **g_modal** / **spModal**: Displays modal windows on the client-side.

Server-Side APIs:

- **GlideRecord**: Performs database operations like querying, inserting, updating, and deleting records.
- **GlideSystem (gs)**: Provides a variety of server-side functions, such as logging, getting session information, and printing messages.
- **GlideDateTime**: Manages date and time operations.
- **RESTAPIRequest** / **RESTAPIResponse**: Used for creating scripted REST APIs.

46. Write a GlideRecord query to retrieve all inactive incidents created in the year 2024 with priority 1, 2, or 3 and an empty assignment group.

```
var incidentGR = new GlideRecord('incident');
incidentGR.addQuery('active', false);
incidentGR.addQuery('priority', 'IN', '1,2,3');
incidentGR.addQuery('assignment_group', '');
incidentGR.addQuery('sys_created_on', '>=', '2024-01-01 00:00:00');
incidentGR.addQuery('sys_created_on', '<=', '2024-12-31 23:59:59');
incidentGR.query();

while (incidentGR.next()) {
   gs.print('Inactive Incident found: ' + incidentGR.number);
}
Use code with caution.
```

47. What is ITSM and ITIL?

- **ITSM (IT Service Management):** The overall process of how IT teams manage the end-to-end delivery of IT services to customers. It focuses on aligning IT services with the needs of the business to deliver maximum value.
- **ITIL (Information Technology Infrastructure Library):** A widely adopted framework of best practices for delivering ITSM. ITIL provides a set of guidelines for IT processes such as Incident Management, Change Management, and Service Request Management. ServiceNow's ITSM modules are based on ITIL principles.

48. Explain the life cycle of a Service Request.

The Service Request lifecycle, often managed through the Service Catalog, typically includes these stages:

1. **Request Submission:** A user submits a request through the Service Catalog. This creates a `Request` (`sc_request`) record and one or more `Requested Item` (`sc_req_item` or RITM) records.
2. **Approval (Optional):** The request may require one or more approvals (e.g., from a manager or budget owner). The workflow pauses until approval is granted or denied.
3. **Fulfillment:** A series of `Catalog Tasks` (`sc_task`) are created to fulfill the request. These tasks can be assigned to different teams or individuals based on the required work.
4. **Work in Progress:** As agents work on the individual tasks, the status of the RITM and the overall Request progresses.
5. **Completion and Closure:** Once all catalog tasks are completed, the RITM is set to a "Closed Complete" state, and subsequently, the Request is closed.

49. What is a RITM and how is it related to a REQ?

- **REQ (Request):** The overarching, high-level request record created when a user submits a Service Catalog item. A single REQ can contain one or more RITMs if multiple items are ordered at once. It provides a summary of the entire user request.

- **RITM (Requested Item):** A specific, itemized request within the overall service request (REQ). Each RITM represents a single item ordered from the catalog and can have its own workflow for fulfillment. The RITM is the record that holds the specific variables and is used to drive the fulfillment process.

Relationship: A Request (REQ) is the parent of one or more Requested Items (RITMs). The REQ acts as a container for the individual RITMs, which represent the actual items and services being fulfilled.

50. Latest version of the platform?

As of September 2025, the latest official ServiceNow release is **Yokohama**, released in Q2 2025. The next release, **Zurich**, is planned for Q4 2025.

ADDITONAL RESOURCES

SEASON 1: SCRIPTING SERIES

Part_1: https://lnkd.in/eUgcai-6
Part_2: https://lnkd.in/gB6h_TX2
Part_3: https://lnkd.in/gkF9weSr
Part_4: https://lnkd.in/ghtxGW4W
Part_5: https://lnkd.in/gHsyqSp5
Part_6: https://lnkd.in/gTUjaCQc
Part_7: https://lnkd.in/g6QR87CY
Part_8: https://lnkd.in/ekuJD3Pg
Part_9: https://lnkd.in/gw2sjGhv
Part_10: https://lnkd.in/gU9qG8Re
Part_11: https://lnkd.in/eMcg6D5w
Part_12: https://lnkd.in/gvVkkqHM
Part_13: https://lnkd.in/gK6Hs4mN
Part_14: https://lnkd.in/gRnewht8
Part_15: https://lnkd.in/gghqqjkt
Part_16: https://lnkd.in/gGPc5vzz
Part_17: https://lnkd.in/gsSbBw2d
Part_18: https://lnkd.in/gb5i6K3x
Part_19: https://lnkd.in/gbmSqjRT
Part_20: https://tinyurl.com/24mv4cka

SEASON 2: SCRIPTING SERIES

Part_1: https://lnkd.in/gfTv2rwv
Part_2: https://lnkd.in/g9WFXx6z
Part_3: https://lnkd.in/ga62m_zA
Part_4: https://lnkd.in/gfEsjf_F
Part_5: https://lnkd.in/g8-R4nuE
Part_6: https://lnkd.in/gqGXPb66
Part_7: https://lnkd.in/gR5Wm_Zb
Part_8: https://lnkd.in/gef9gpMV
Part_9: https://lnkd.in/g-rEg64a
Part_10: https://lnkd.in/gnauU2Vp
Part_11: https://lnkd.in/gC3VVzMm
Part_12: https://lnkd.in/g4scDp_d
Part_13: https://lnkd.in/gKYUeH9H
Part_14: https://lnkd.in/gTqiPt7f
Part_15: https://lnkd.in/gXBw9YFF
Part_16: https://lnkd.in/gbCb8Akb
Part_17: https://lnkd.in/gZdxbmTA
Part_18: https://lnkd.in/gCrxpx4w
Part_19: https://tinyurl.com/2aamp25t
Part_20: https://tinyurl.com/2b3bo4nb

SEASON 3: SCRIPTING SERIES

Part_1: https://tinyurl.com/2xmyynv9

Integration Series

SOAP GUIDE: https://tinyurl.com/2267wayd
Integration eBook: https://tinyurl.com/28oxb4ae

www.hacknow.co.in

READY TO MAKE YOUR SERVICENOW SKILLS THE STUFF OF LEGENDS?

Or are you worried those tricky interview questions will leave you in a pickle? Don’t sweat it—we’ll turn those tricky questions into easy-peasy, extra-cheesy wins!

Think your ServiceNow chops are sharp? Let’s make you the big cheese in your next interview! At HackNow, we transform your skills from “pretty good” to “absolutely unforgettable.’

- ✓ Mock Interviews—Tailored for ServiceNow superstars (yes, that’s you!)
- ✓ Real-world scenario and battle-tested questions that separate pros from pretenders
- ✓ Rockstar resume and Linkedin glow-up you can brag about
- ✓ Core concepts and project Walk-throughs loaded with “aha!” moments
- ✓ UNLIMITED interview calls with top companies—no secret code requlred

Why just ace your interview when you can crush it like a nacho under a mountain of cheese? Whether you’re just starting out or making a dramatic career comeback. we’ll put your ServiceNow talents center stage so you never hear another silent rejection.

READY TO WOW YOUR INTERVIEWERS” LET’S PUT ON A PERFORMANC... SO GOOD. THEY'LL NEED POPCORN

[www.hacknow.co.in]

Don’t dream it. DO IT. The Next ServiceNow Superstar could be YOU.

Call/WhatsApp: +91 8984988593

Call/WhatsApp: +91 8984988593

Call/WhatsApp: +91 8984988593

★ THANK YOU! ★

With the People,
United in Purpose!
For the People;

We deeply appreciate your support and trust in HackNow. Together, we strive for **freedom, equality, and innovation** – empowering every learner to succeed.

From Nothing ▶ to Now, It's HackNow...

Let's keep growing, learning, and hacking forward –– now. and always.