1. **What is Global Business? (Fujitsu First Round)**

Global Business Rule is a business rule that will run every time a page loads in the system. The selected table for this is global and hence there are no restrictions or table conditions.

Using global business, will impact the system performance as it runs every time. Instead of global business rule, we can use script include as it will run only when invoked.

Steps to create Global Business Rule:

1. Create a new business rule in Global Scope and select the table as global.
2. Click on Advanced and Remove Everything and code any function.



1. Save the BR.
2. Create another BR with some conditions and use the created function directly.



1. You will see that it runs every time.
2. **Why we still have Global Business Rule if there is script include feature present? (Fujitsu First Round)**

I believe that Global BR is still available only for backward compatibility and in past it was also used as globally accessible function.

According to ServiceNow, global business rules can load on every page of a system, but there is no benefit to loading scripts on every page. Instead, you can move the function definition to a Script Include, and the name of the Script Include must match the name of the function

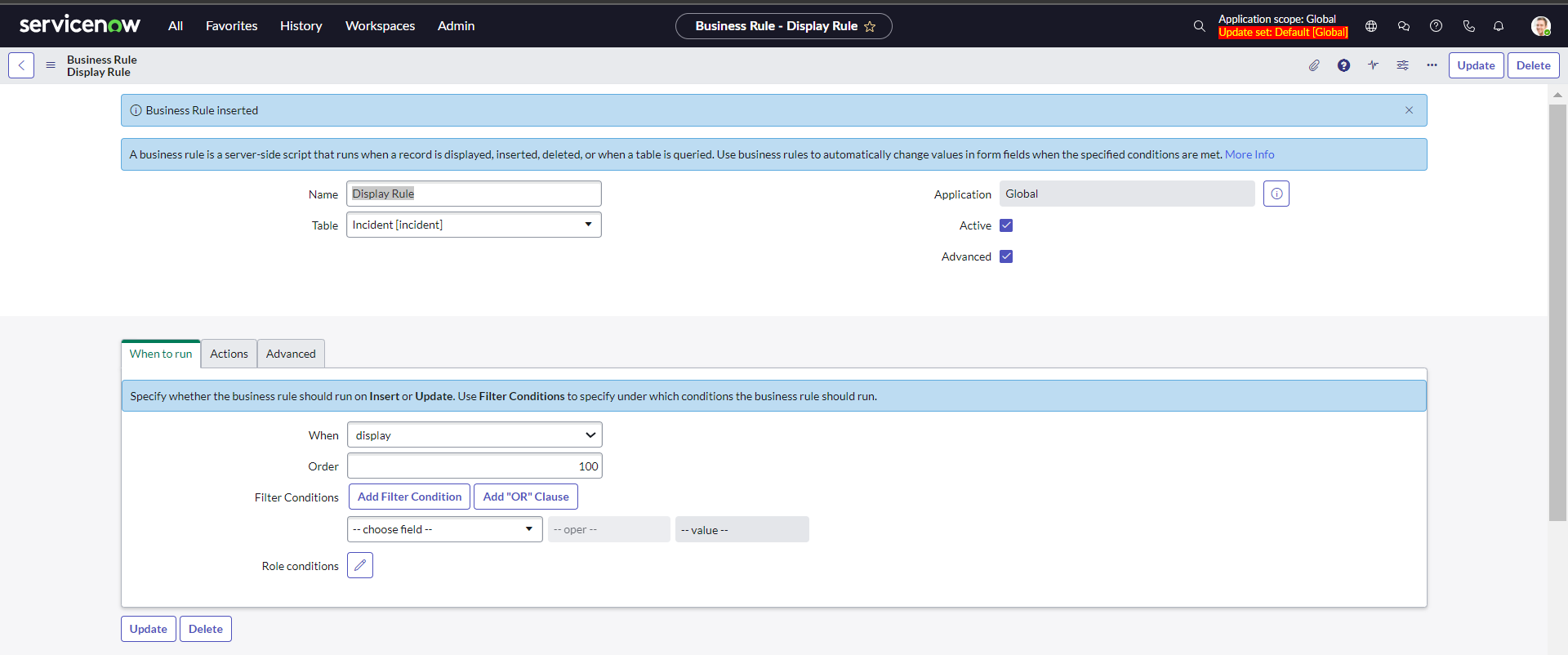
1. **What is Display Business Rule? (Fujitsu First Round)**

Display Business Rule runs before the form is displayed or presented to the user, just after the data is fetched from the database.

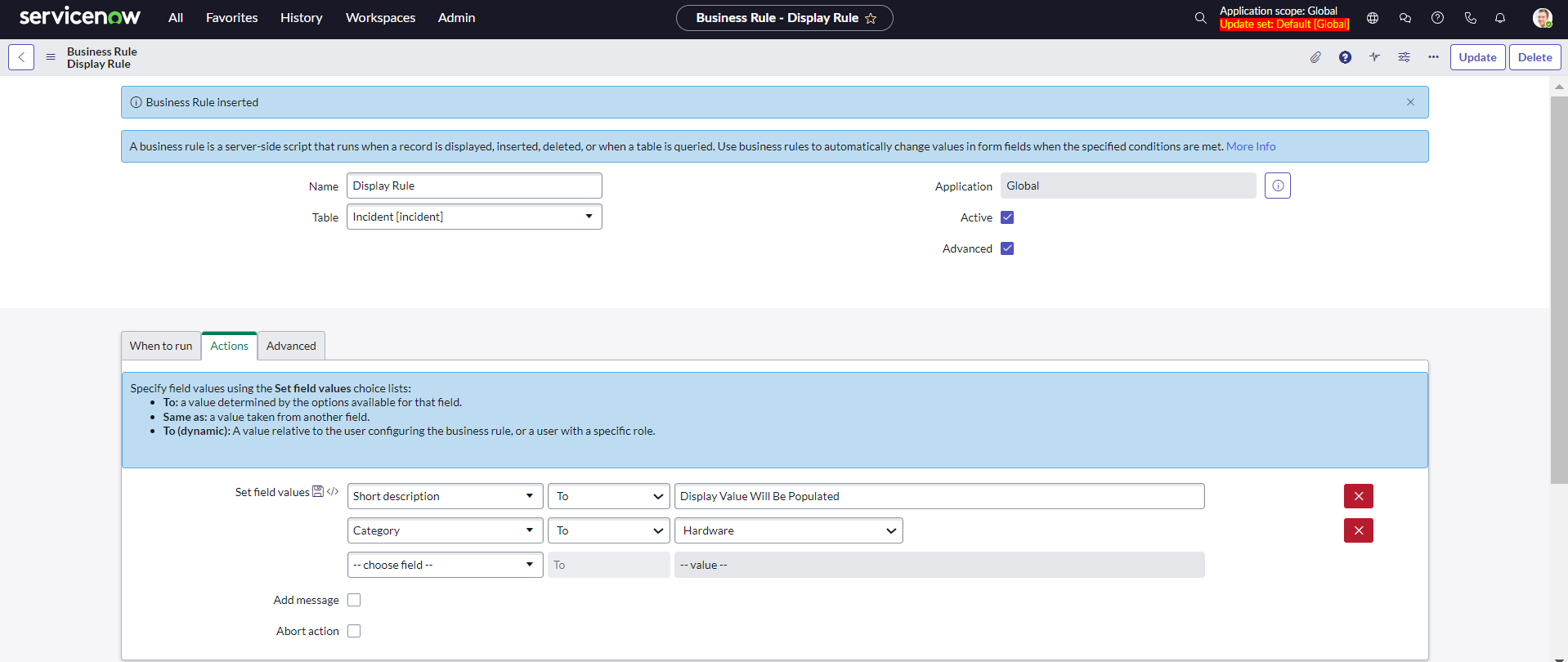
It can be used to pre-populate some fields like category, short description and etc, before the form is displayed to the user.

To create a Display Business Rule: -

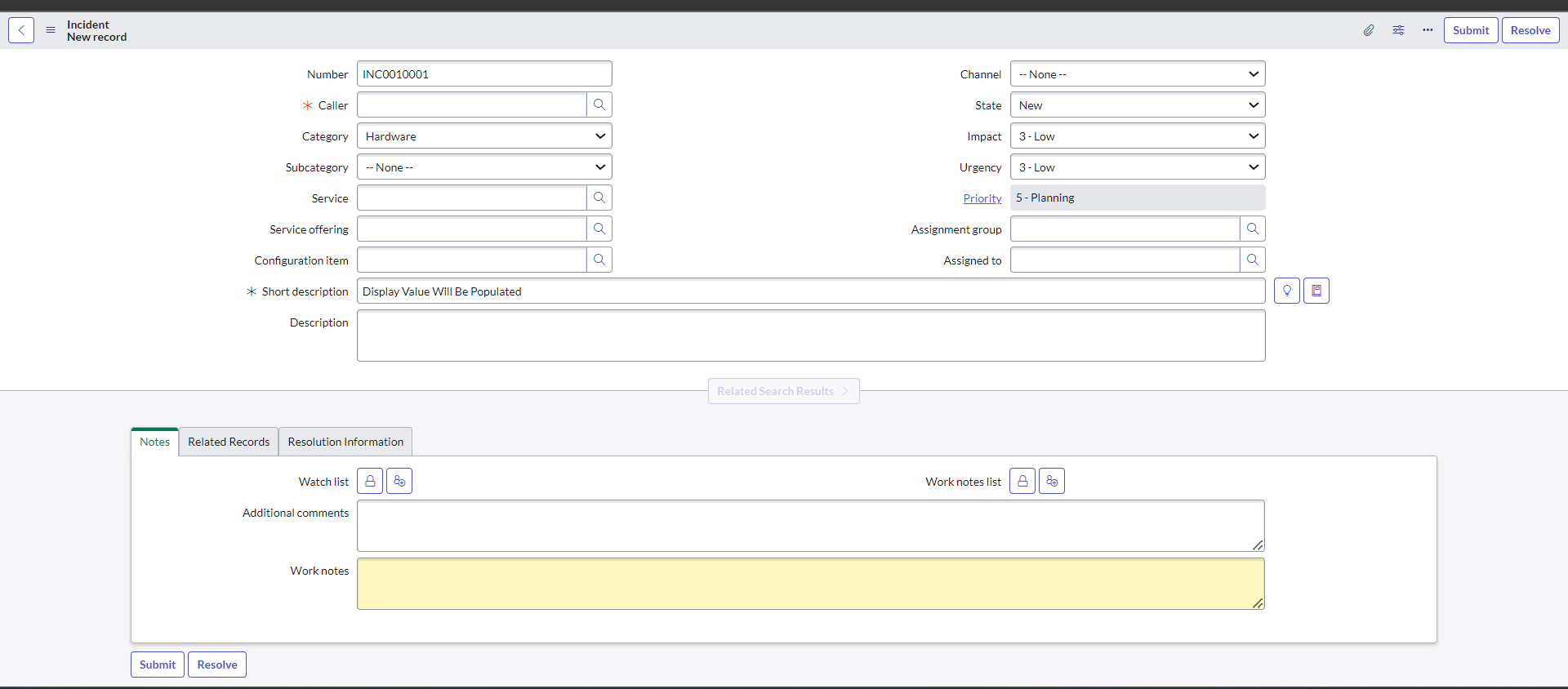
1. Create a new Business Rule and Select and Table.
2. In When to Run Section, select when to Display and select filter conditions if required.



1. You can choose to write script from advanced section by clicking on advanced checkbox or directly population field from the action section.



1. Save the Business Rule
2. Try a new record and you will see the fields are pre populated.

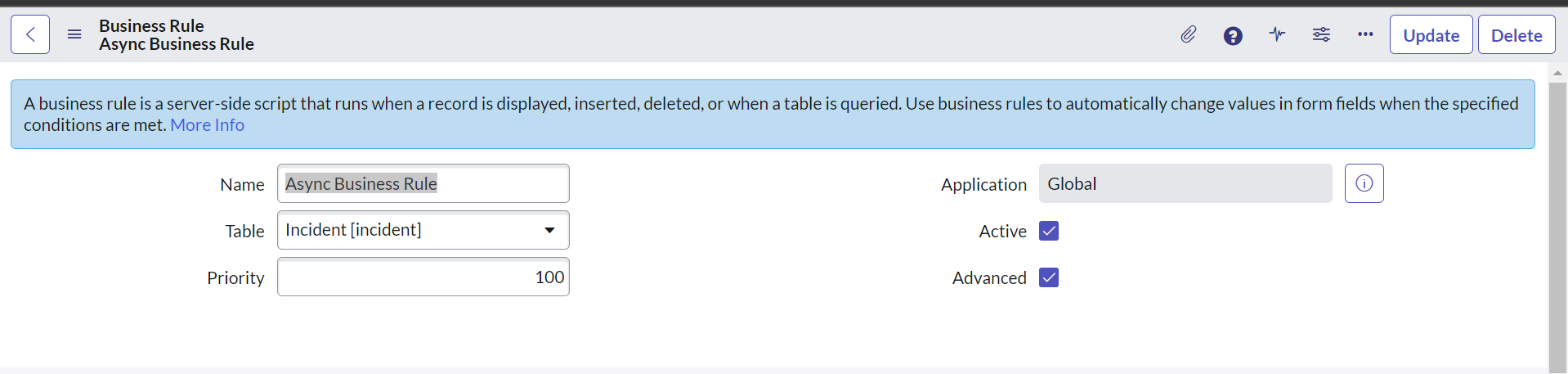


1. **What is Async Business Rule? (Fujitsu First Round)**

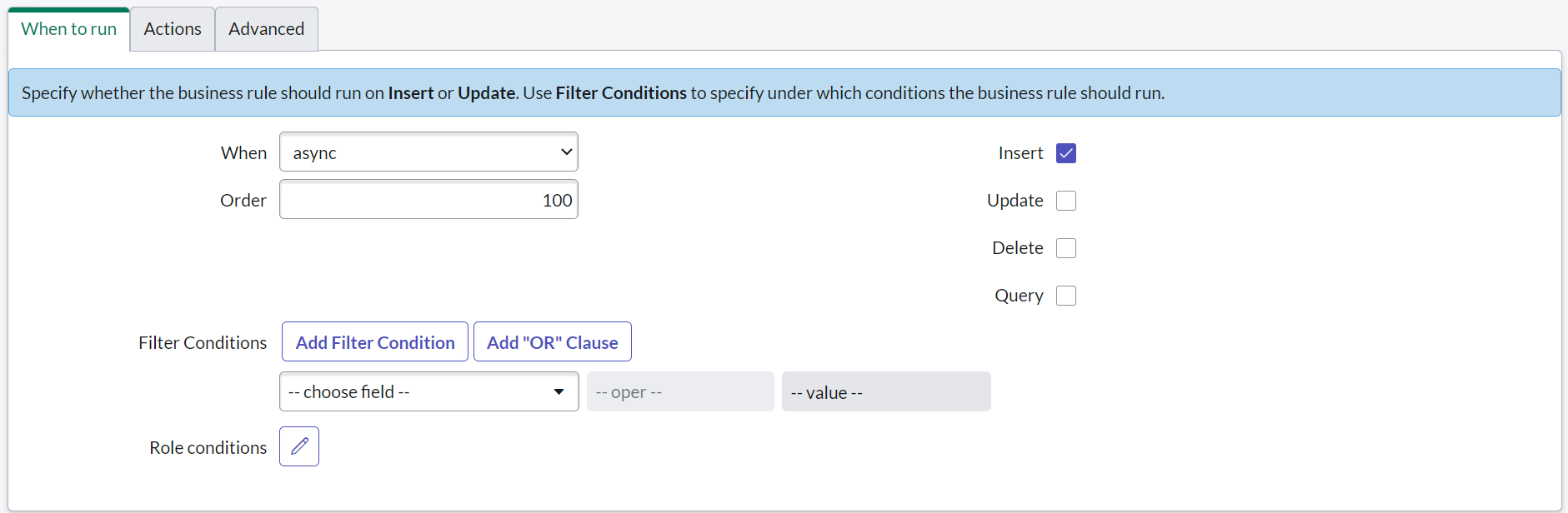
Asynchronous Business Rules (ABRs) run after database commits in the background, simultaneously with other processes. This allows ServiceNow to return control to the user sooner, but may take longer to update related objects. ABRs are similar to **After Business rules**. The only difference is that **After** business rules return a response to the client and might take time due to the ongoing transaction, whereas ABRs run in the background. This means that when the form reloads, the results of any **After** business rules will be shown, but not the results of ABRs as it run in the background.

To Create an Async Business Rule, follow these steps:

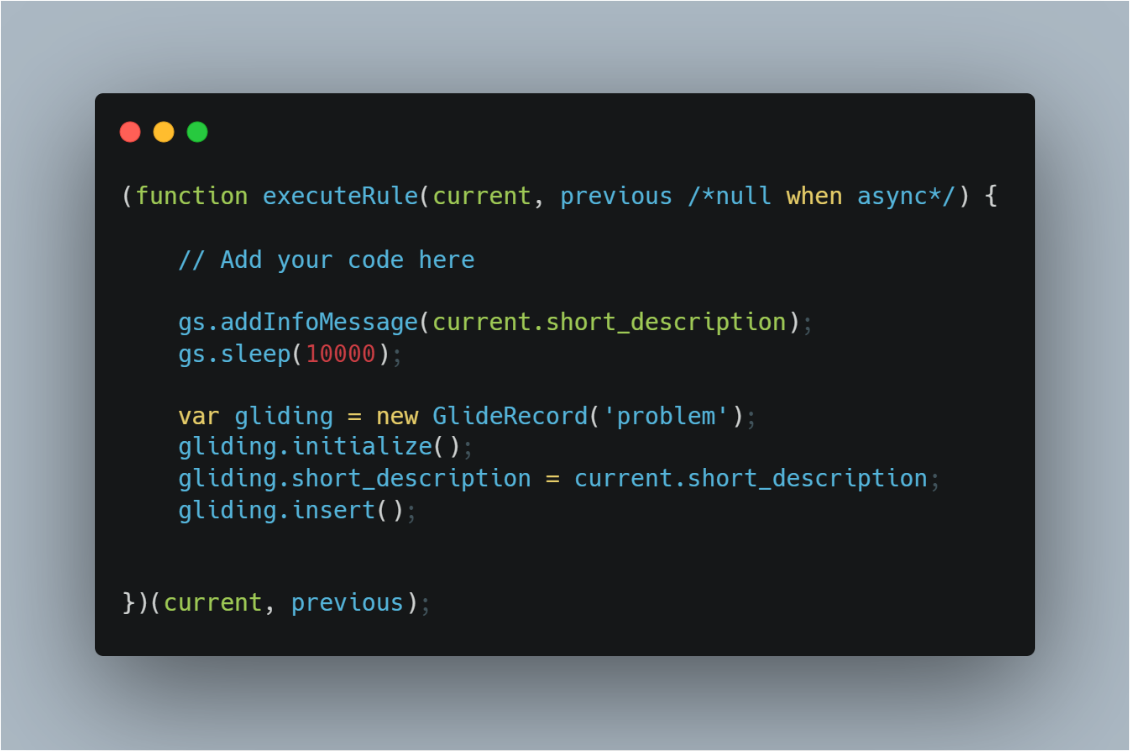
1. Create a new Business rule on Any Table and check the Advanced checkbox to bring the when section



1. Select Async from the When Dropdown and check Insert or Update or any checkbox to select the action timing.



1. Navigate to Script section and write an glide record



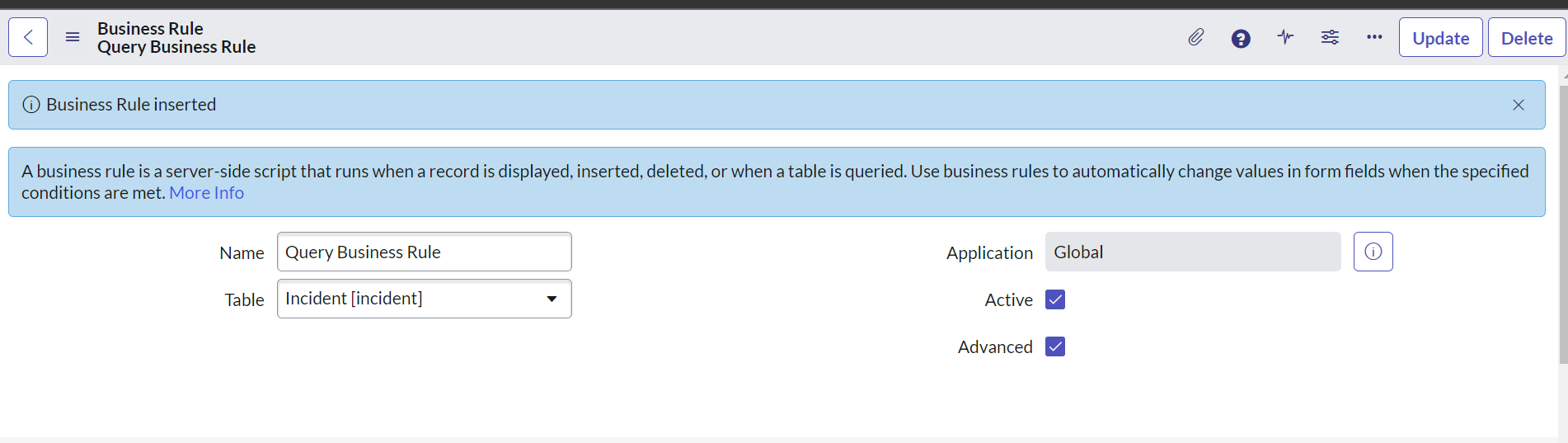
1. Create a new incident. You will see that it never sleeps for 10 seconds because of the async BR, the transaction is processing in the background.
2. **What is Query Business Rule? (Fujitsu First Round)**

Query Business Rule runs on any query operation. So, whenever we open any record or change something or just navigate to list view, ServiceNow is querying the database behind the scene. So, query is the first thing performed whenever we try to do something on a table.

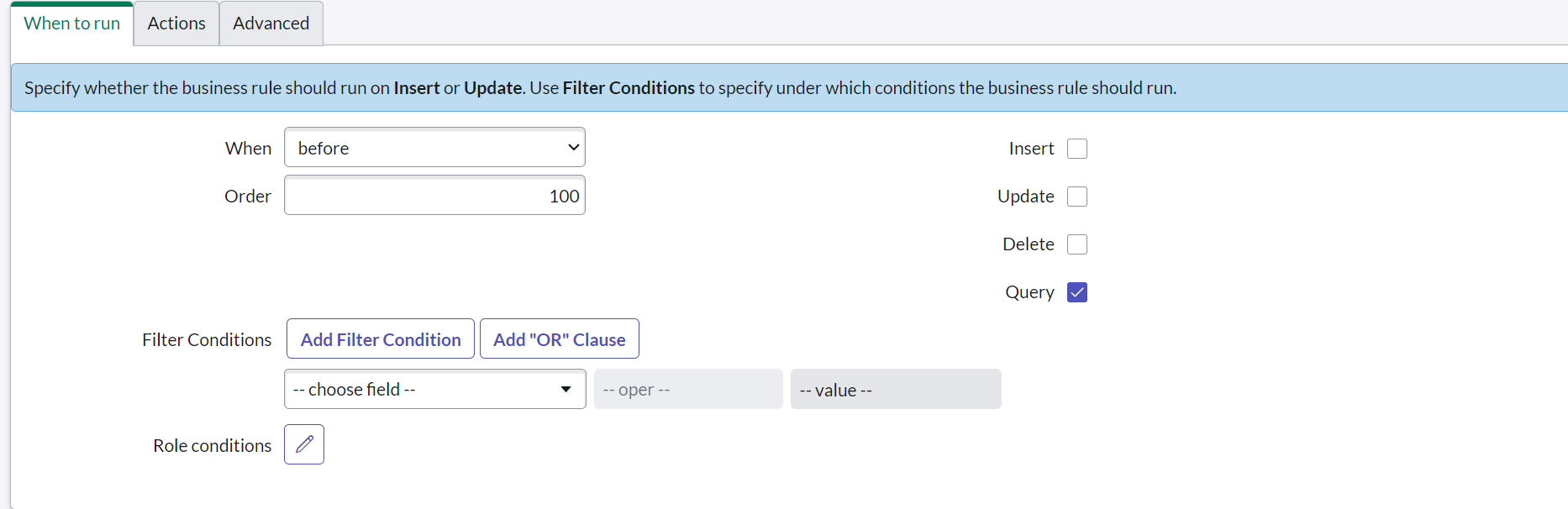
Example: If you open incident list view, then ServiceNow is querying behind the scene to fetch all the incident records and display it in list view.

To create a query business rule, follow these steps;

1. Create a new Business Rule in the same scope as the table you are going to select and click on the advanced tab.



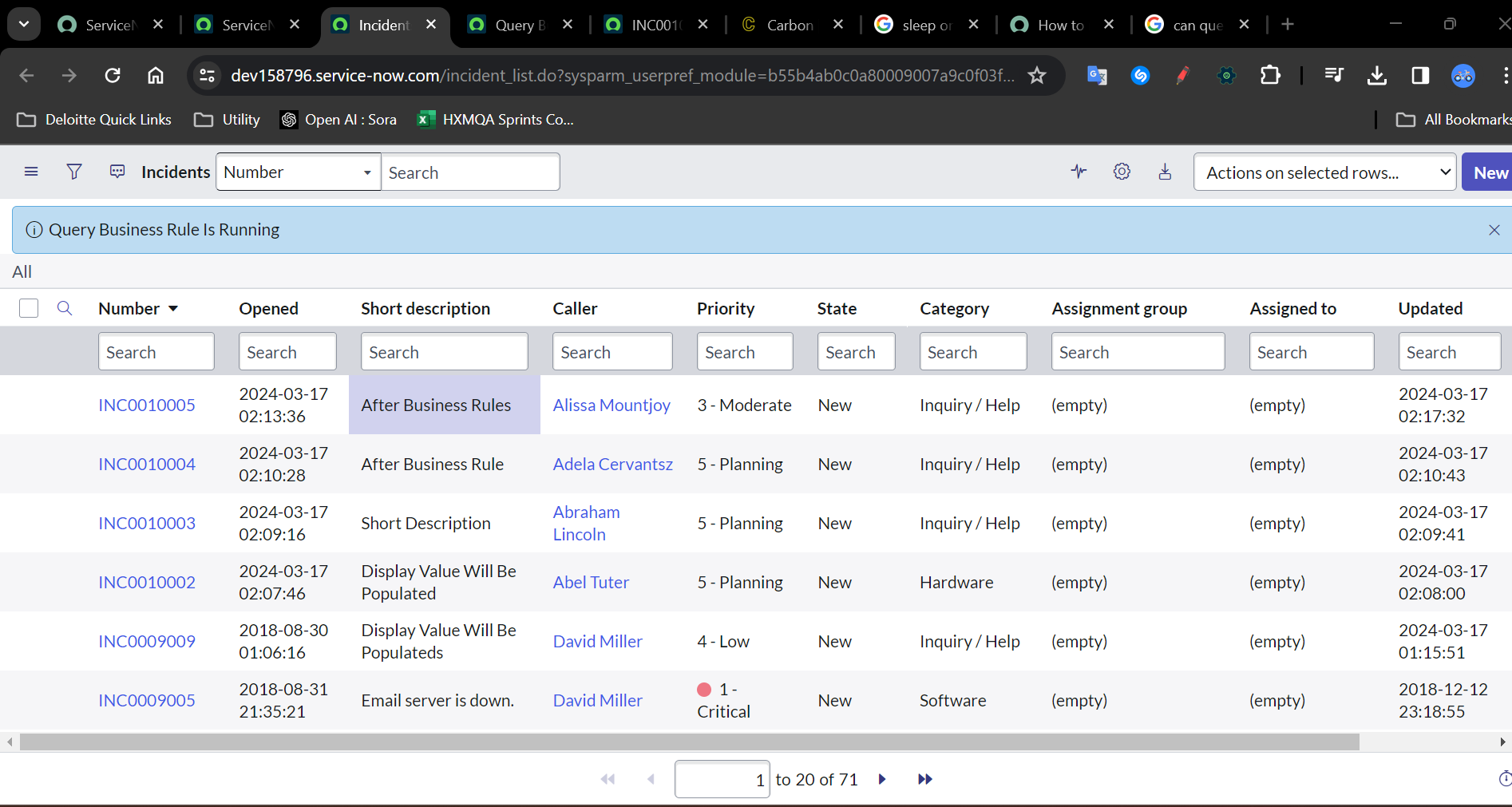
1. In when to run section, select when and check the query checkbox.



1. In the advanced section, add an info message.



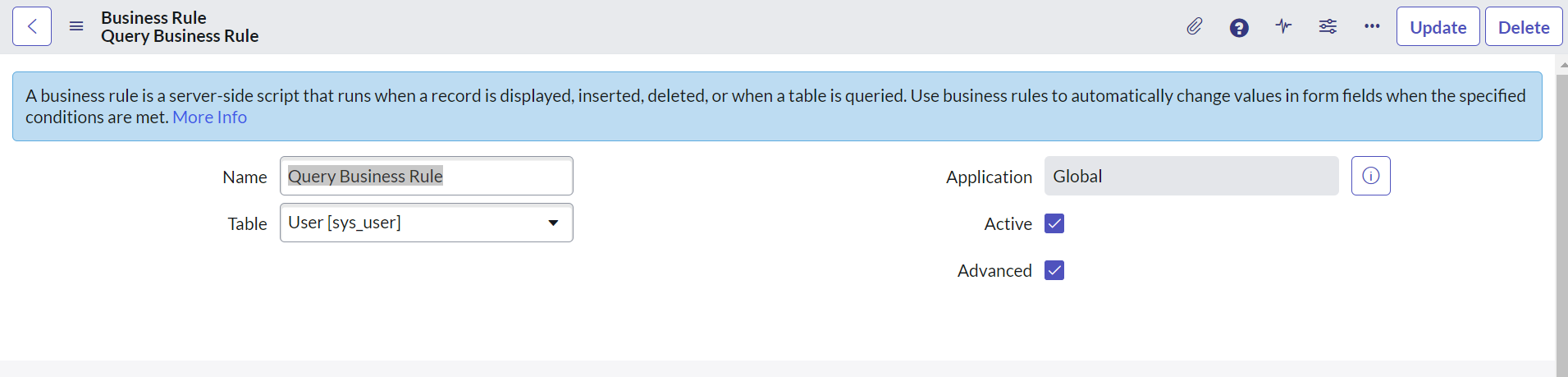
1. Save the record and navigate to the list view of the Incident, you will see the info message there.



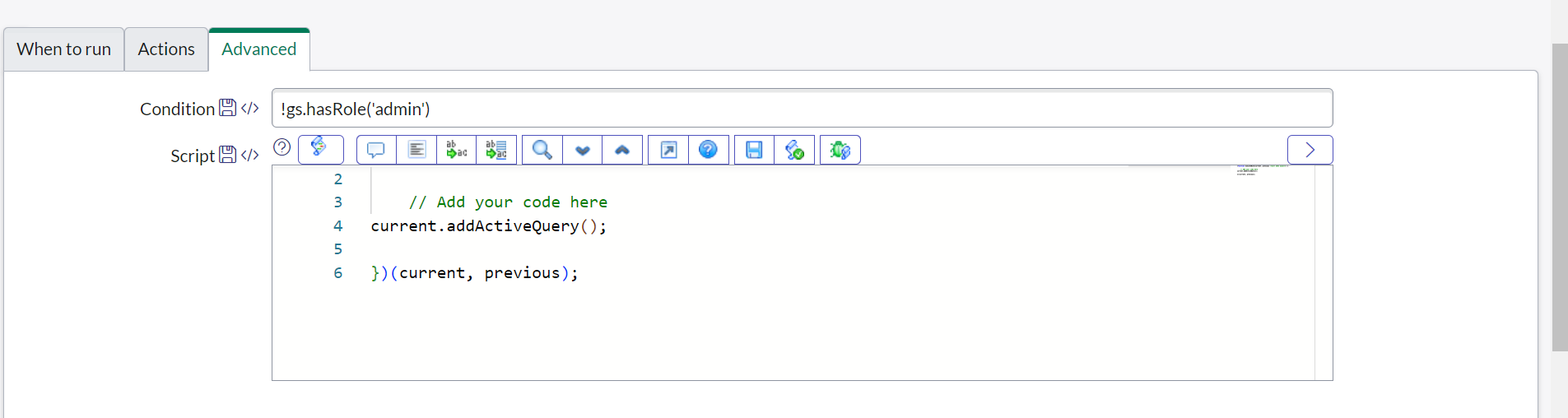
1. Try to make any change in the list view and see that the BR runs on every change.

An industry use case of Query BR can be to show only active users to non-admins

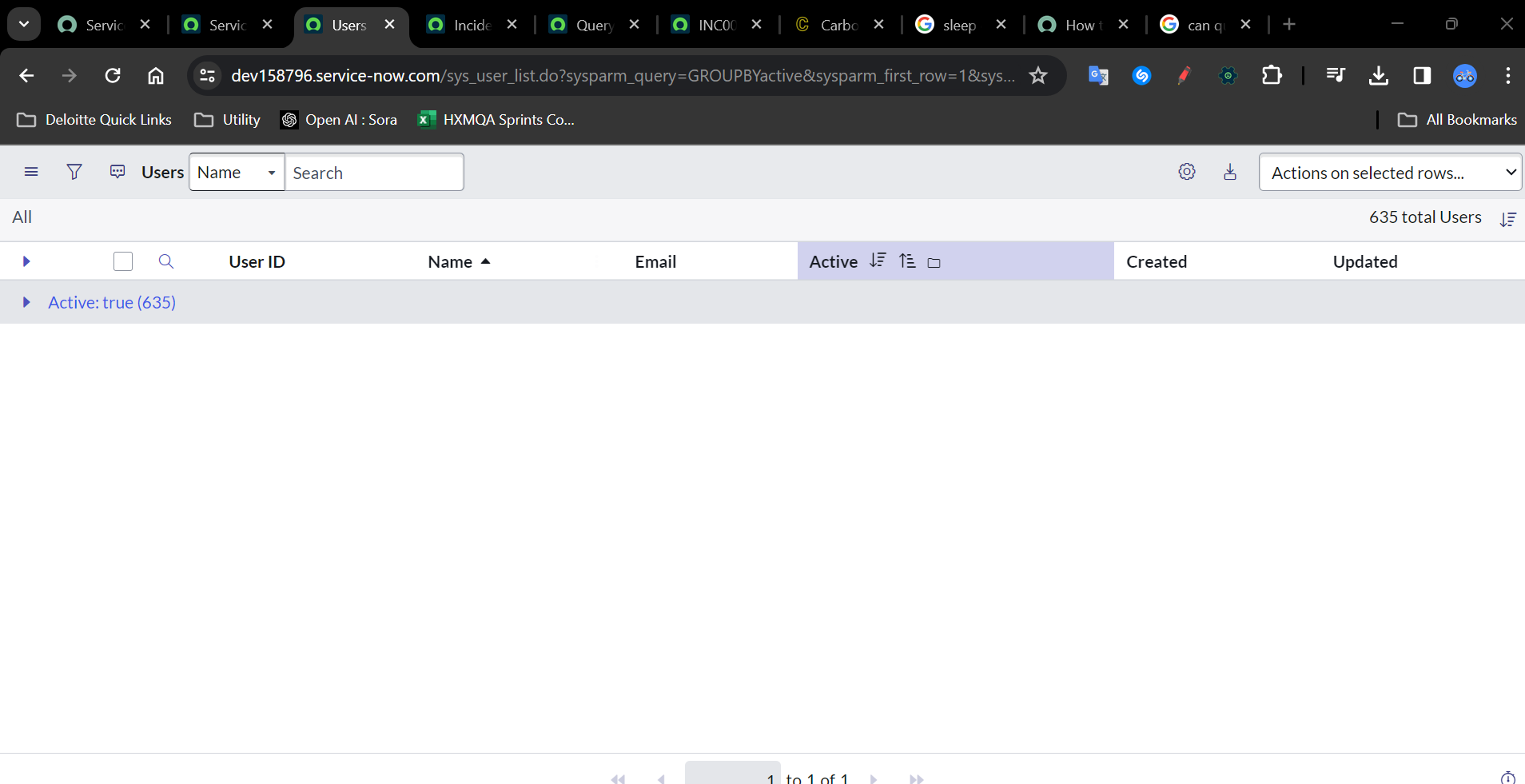
1. Change the table in the Query BR to user.



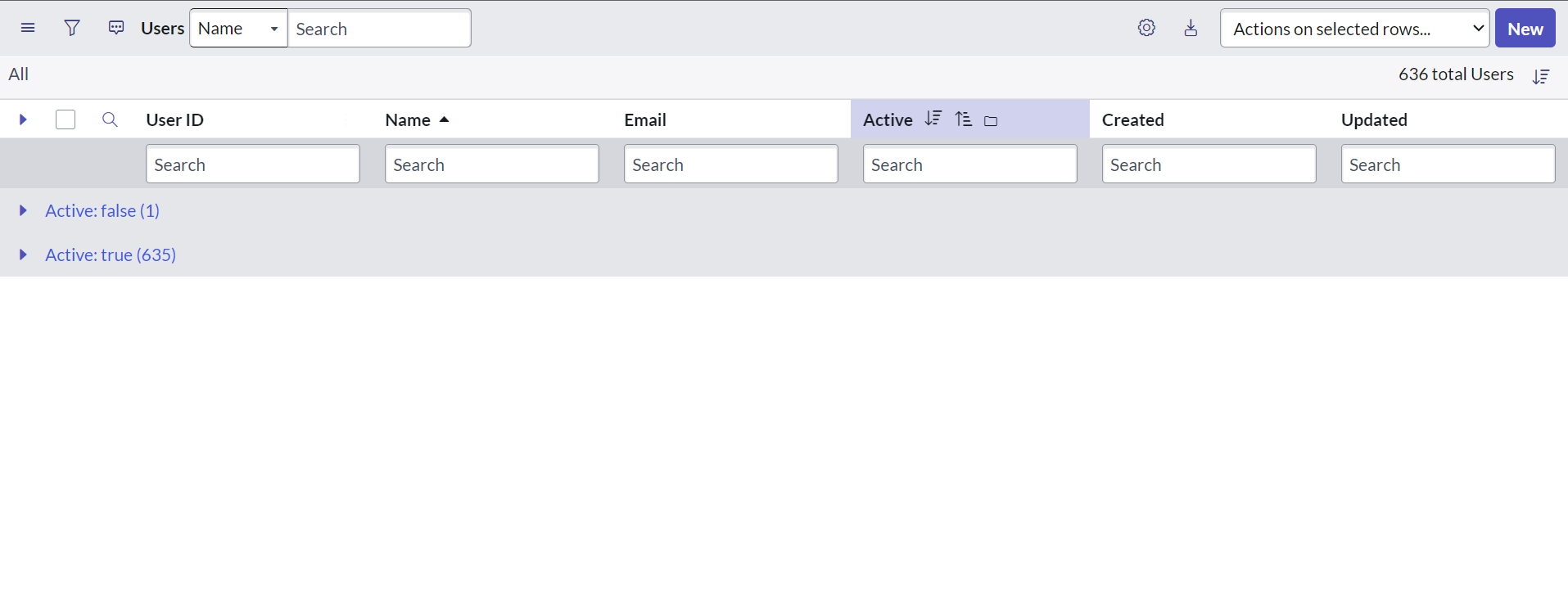
1. Navigate to Advanced section and add condition and modify the script.



1. Save the form and impersonate a non admin user and check the user table.
2. You will notice that you can only fetch active users there.



1. Now, end impersonation and check with admin user and see the difference.

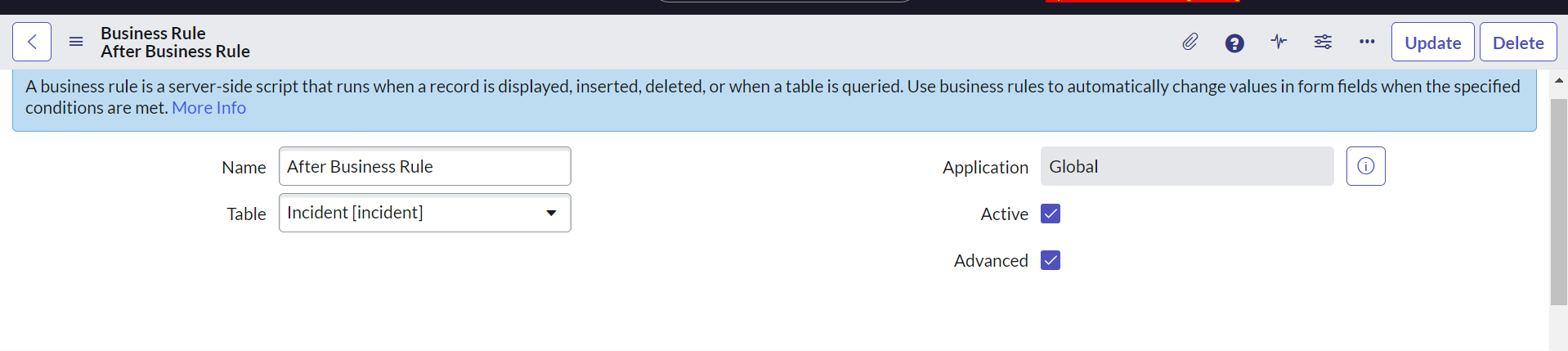


1. **What is After Business Rule? (Fujitsu First Round)**

After Business Rule runs after the data is inserted to database. It is similar to Async BR. The only difference is Async BR runs in the background and gives the control back to the user, meanwhile the control will not be transferred back to user until the After BR has executed successfully.

To Create an After Business Rule, follow these steps:

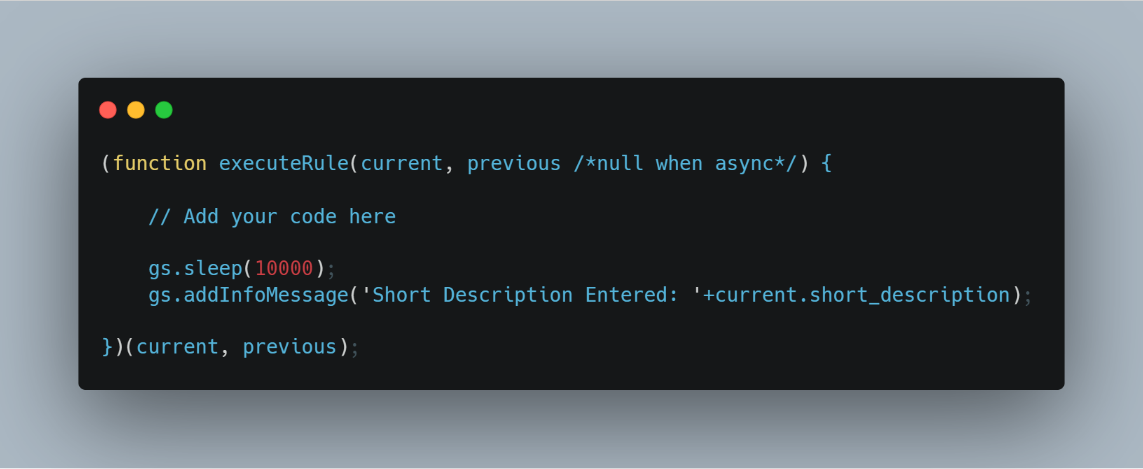
1. Create a new Business rule on Any Table and check the Advanced checkbox to bring the when section



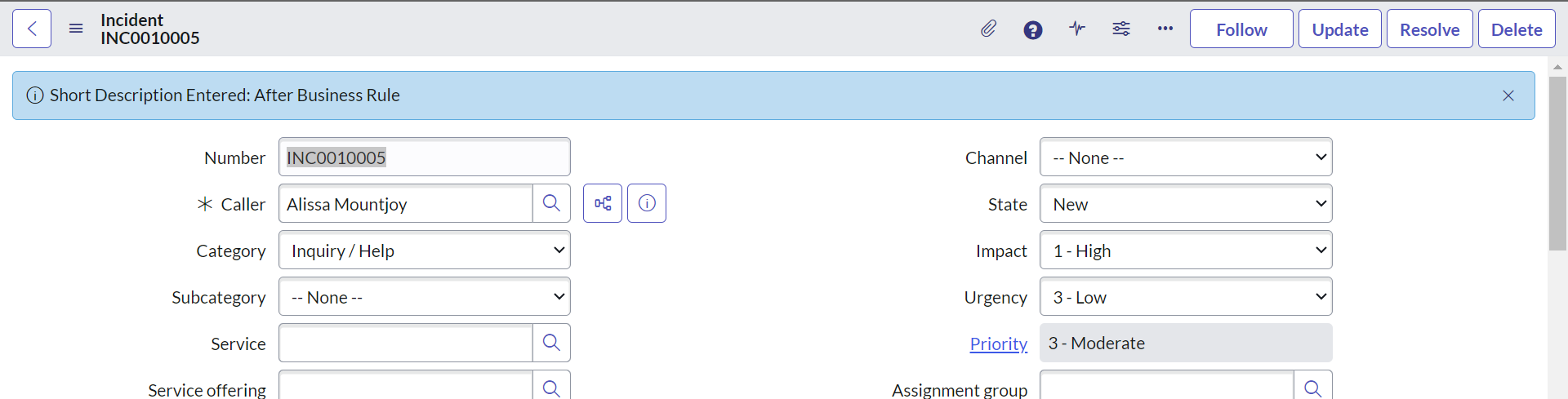
1. Select After from the When Dropdown and check Insert or Update or any checkbox to select the action timing. Also populate any filter condition if you want



1. Navigate to Script section and write an gliderecord



1. Create a new incident. You will see that it will sleep for 10 seconds before displaying the business rule as after rule will not control back until the transaction is complete.

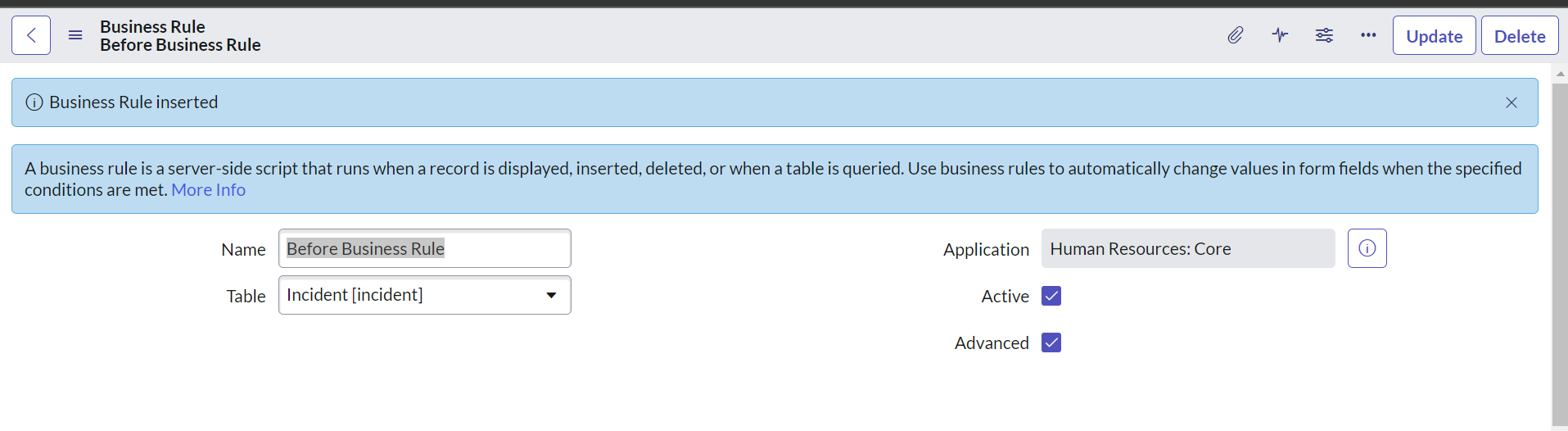


1. **What is Before Business Rule? (Fujitsu First Round)**

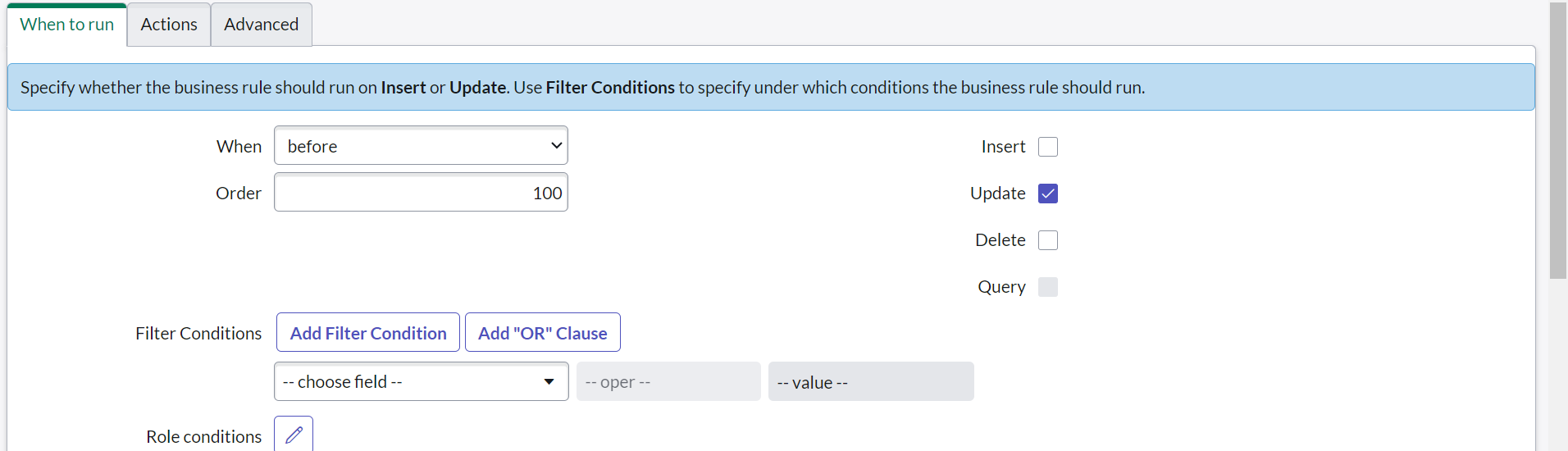
Before Business Rule runs before the data is inserted or updated into database. It can be used to create a validation on Server-Side end like validating if user has sufficient balance in the account before making transaction.

To create a before business rule, follow the below steps:

* 1. Create a new Business rule on Any Table and check the Advanced checkbox to bring the when section



* 1. Select Before from The When Dropdown and check Insert or Update or any checkbox to select the action timing.



* 1. Add any script in the advanced section



* 1. Save the rule and try to update an insert record and you will see the info message.



1. **Difference between Display, Before and After Business Rule (Fujitsu First Round)**

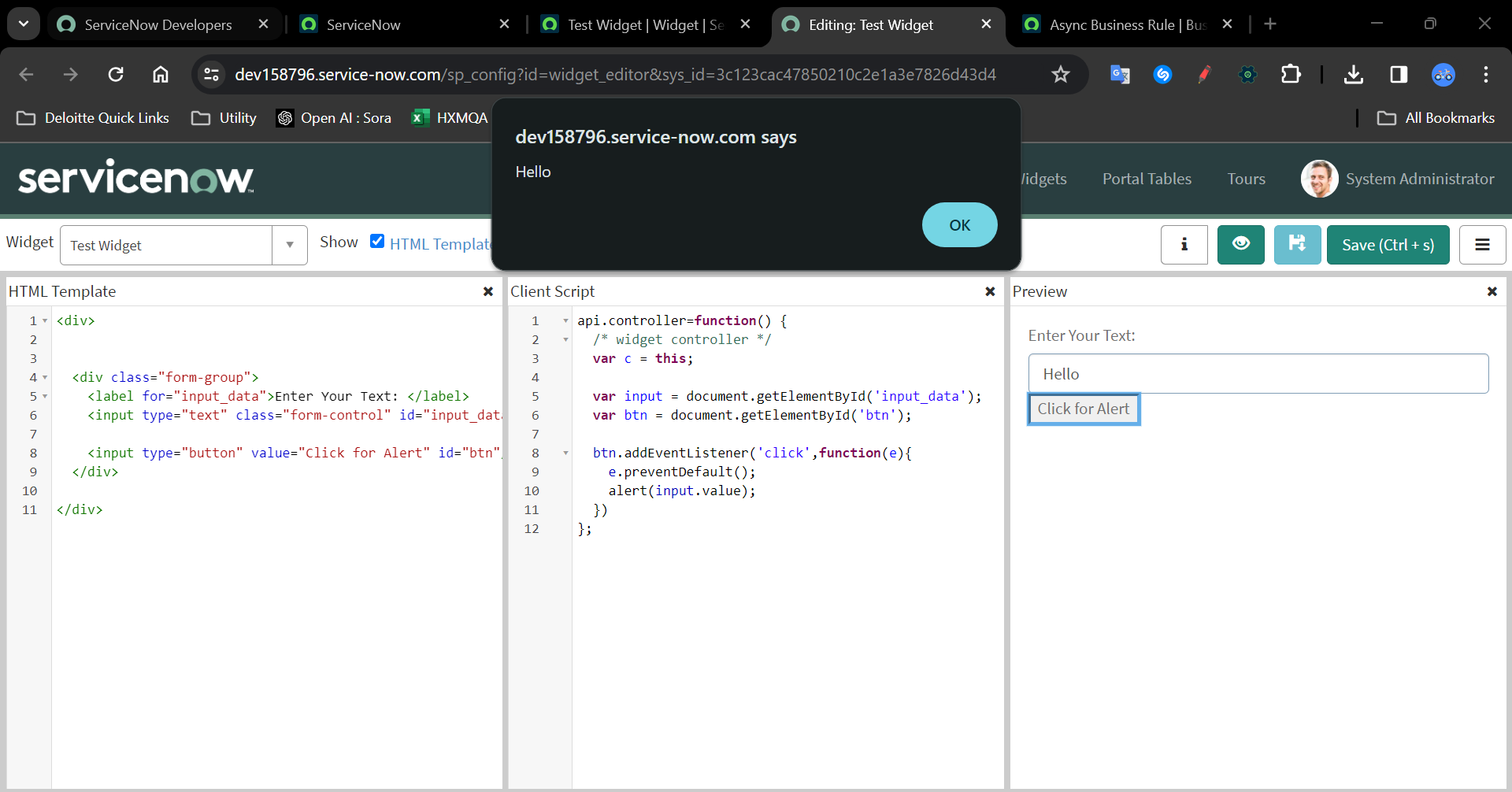
|  |  |  |  |
| --- | --- | --- | --- |
| **Aspect** | **Before Business Rule** | **After Business Rule** | **Display Business Rule** |
| Definition | Executed before a certain event or action. | Executed after a certain event or action. | Relates to how information is displayed. |
| Implementation | Typically enforced through constraints or validation logic. | Implemented using triggers. | Enforced at the application layer. |
| Purpose | Validates data or conditions before processing further. | Enforces additional constraints or triggers actions post-event. | Governs data presentation to users. |
| Example | Check if account has sufficient funds before withdrawal. | Update inventory levels after a purchase transaction. | Display data in light or dark mode. |

1. **In what order will Before, After, Async and Display Business Rule Run? (ServiceNow First Round)**

First **Display Business Rule** will run when the form loads. Next, **Before Business Rule** will run, before inserting data into the Database, Next **After Business Rule** Will run after inserting the data into Database and then **Async Business Rule** will run if there is any background transaction.

**Display->Before->After->Async**

1. **Create a widget that will take an input as text and show an alert with the input value entered on button click. (Cognizant First Round)**

****

**HTML:**

****

**Client Script:**

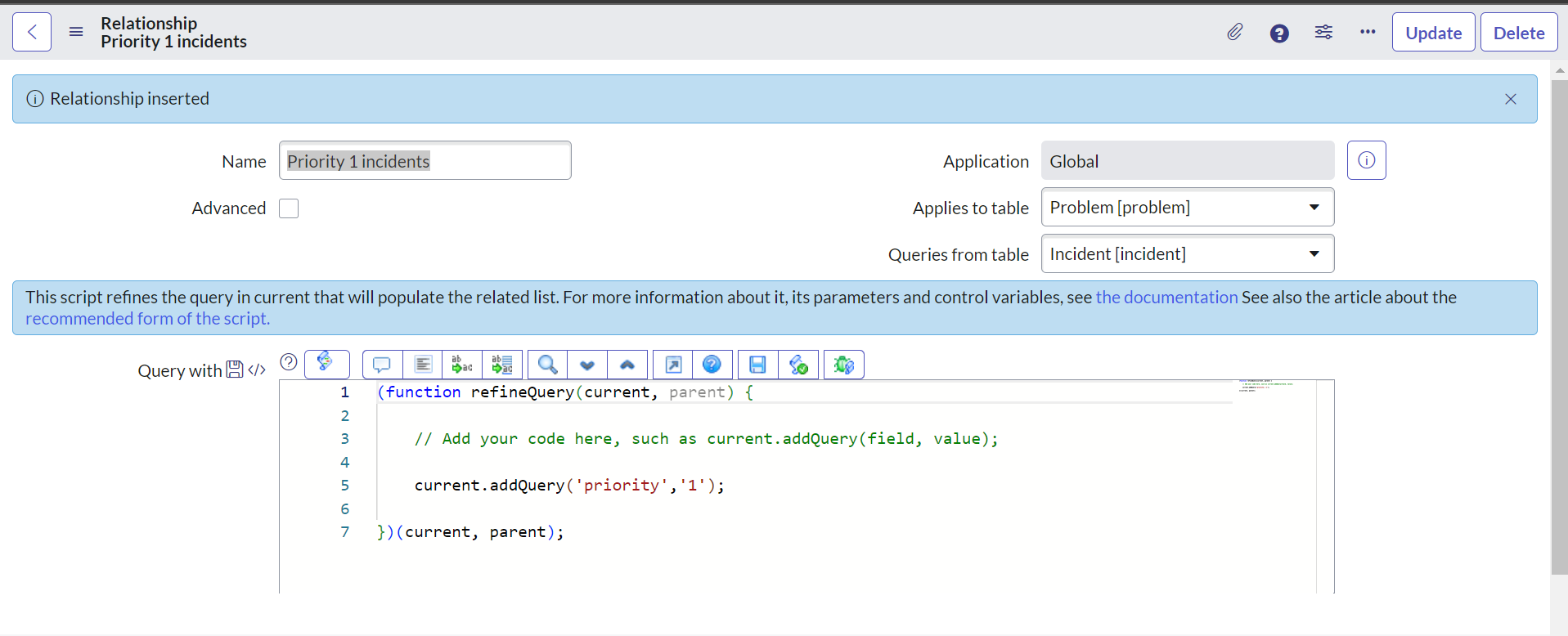
****

1. **How to create a related list and add it on a form? (Fujitsu First Round)**

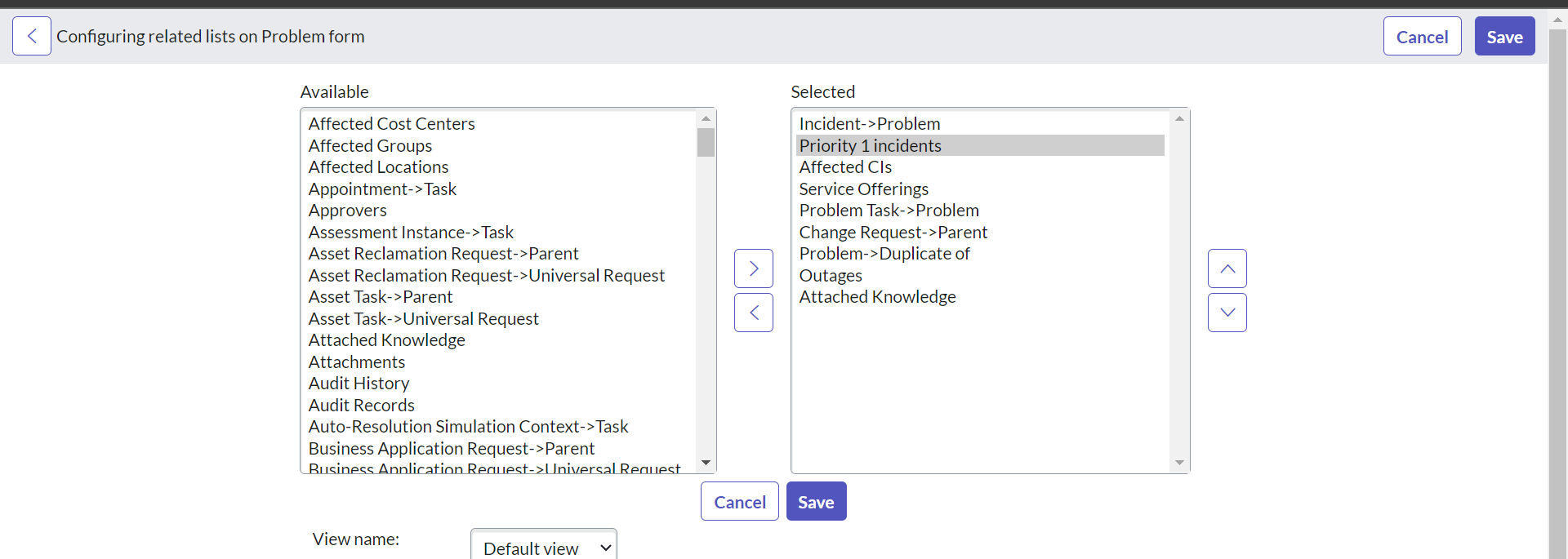
Before creating a related list, lets understand what is a related list.  
As the name suggest, related list is a list that holds data from a table that is related to the current table.  
  
Example: A case form can have Knowledge Articles or HR Task Related list, meanwhile an incident form can have problem or priority Related List.

Steps to create a related List:

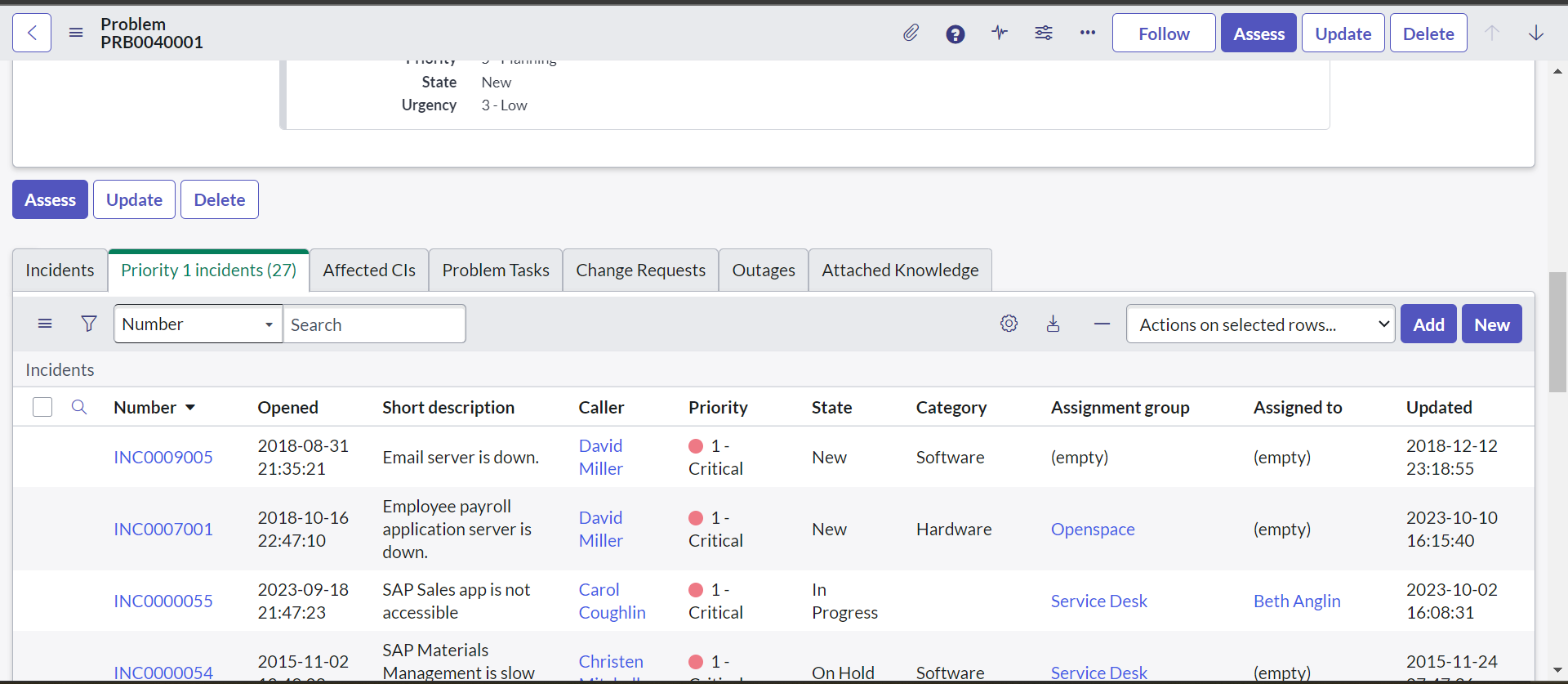
1. Navigate to System Definition -> Relationships and create a new record.
2. Fill in the fields like Applies to Table means in which table you want to show it
3. Queries from table means from which table you want the related data to be shown.
4. Query with section can handle any additional filter queries
5. Save the form



1. Navigate to the record, where you want to see the related list
2. From the hamburger icon, navigate to Configure -> Related List and pick and arrange the related list you have created and save the slushbucket.



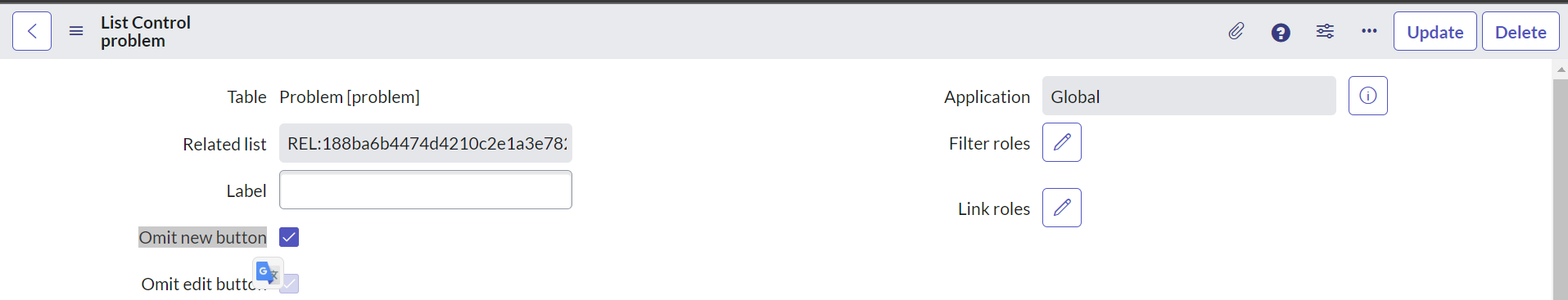
1. Check If the related list is visible.



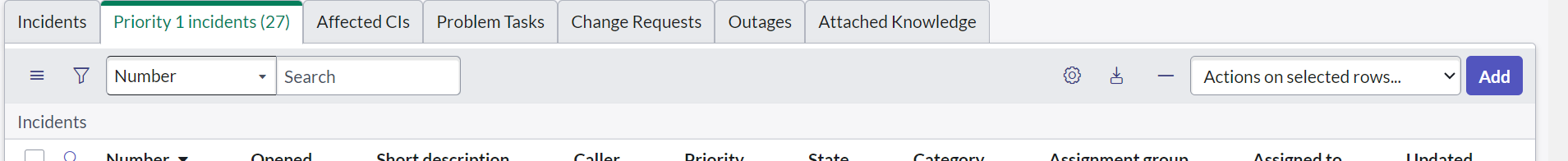
1. **How to hide UI Actions from a Related List?**

A Related list can have some UI actions like Add or New. To hide a New UI Action from a related list, follow these steps:

1. Right click on any column of the related list
2. Go to Configure -> List Control
3. Check the Omit new button and save the form



1. Check if the New UI Action is still visible,



1. **How to make a server-side field mandatory? (Fujitsu First Round)**

It is not directly possible to make a field mandatory on server side in Servicenow. However, one can write some validation using before business rule to check if the value is not blank of the field.

Script for doing this is given below:



Other resources like Data Policies can be used to make a field mandatory on the server side.

For example, Let’s say if user id on user table is made mandatory using Data Policy. If an integration is trying to create a record without populating the user id field, the update will fail.

1. **What is g\_scratchpad?**

g\_scratchpad is an object that is used to pass data that is not available on the form from server side to client side. It is recommended that it is only used with Display Business Rule or Workflows.

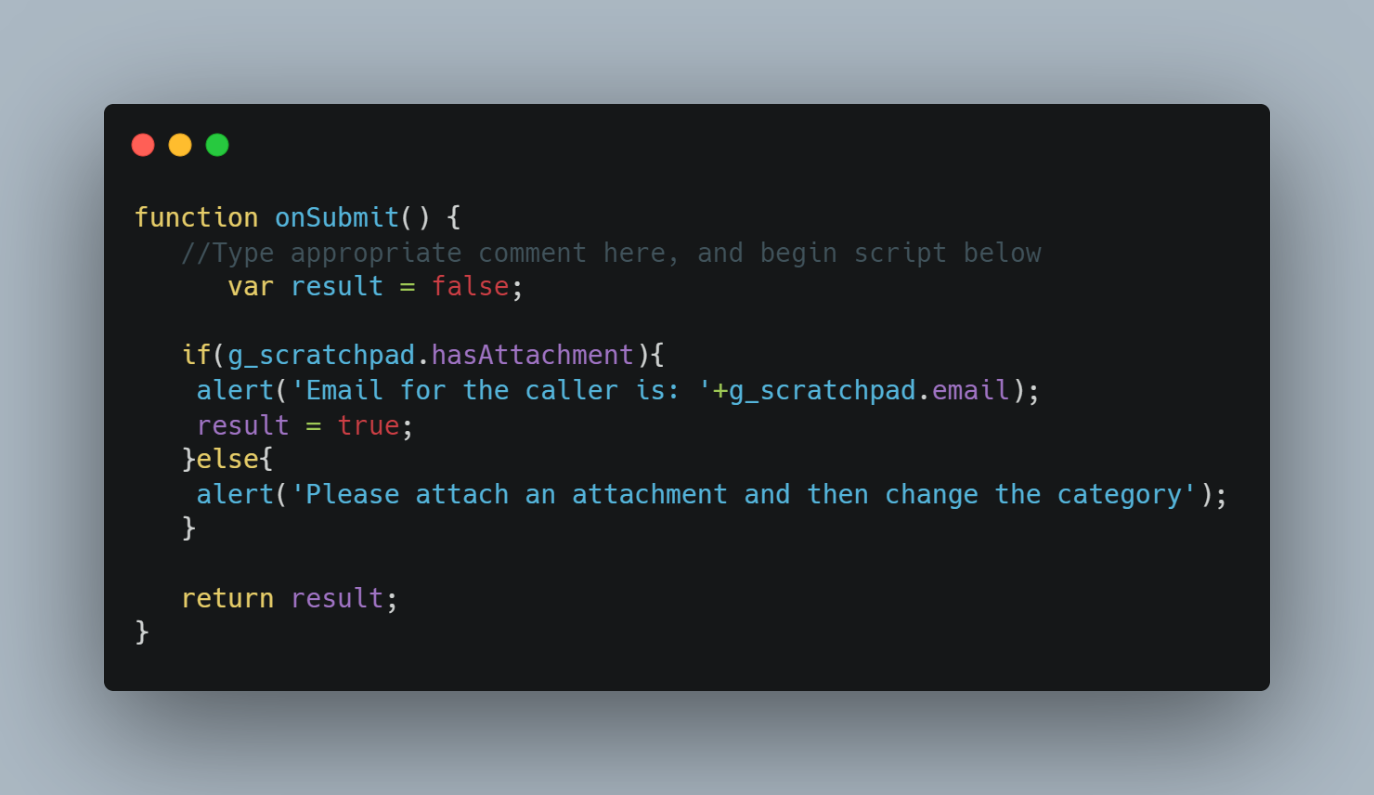
If we know that what information the client need from the server before the form loads, then in that case, once can use a display business rule with g\_scratchpad object to hold the data.

Steps for implementing it:

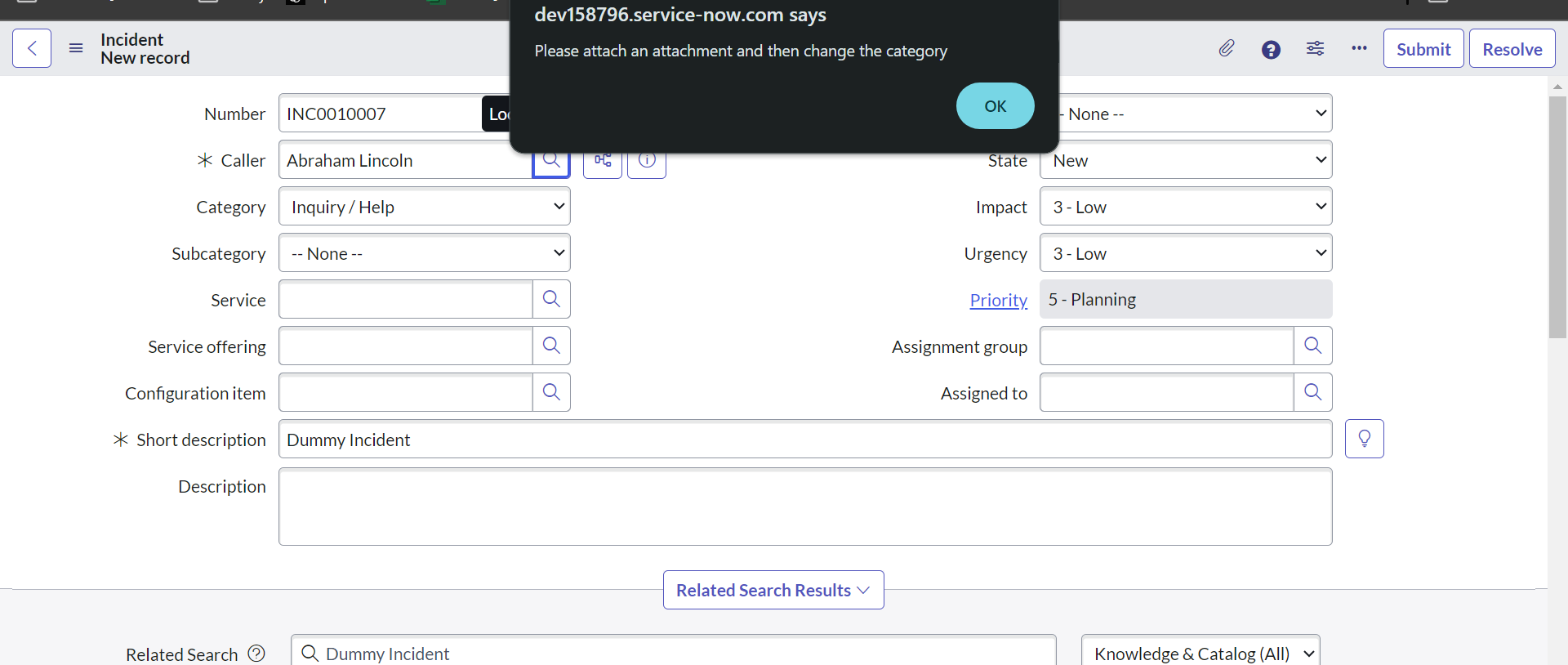
1. Create a display business rule and add the following script
2. Write the following script to check if incident has attachment.



1. Create a client script that will check on submit that If attachments are present on the case.



1. Check a already created incident and try to re-submit something and check the alert message there



1. **Difference between server.update() and server.get() (Fujitsu First Round)**

**c.server.get()** and **c.server.update(),** both methods are used in Service Portal to query the server. Most of the functionality of these method is same but there is a major difference in both the functions

a) **c.server.get()** method is used to fetch data from server side. It can fetch data without reloading the form. c.server.get method should only be used to fetch data from server. It can also be used to update data but as best practice, it should only be used to fetch data like list of records.

For example, you might use it to retrieve user preferences, fetch a list of records, or get configuration settings.

b) **c.server.update()** method is used to update data to server side. It can also be used to fetch data from server side but as best practice, it should only be used to update data on server side like update short description of a case.

For example, you might use it to save changes made by a user in a form, update the state of a record, or trigger a workflow based on user interactions.

1. **An example where you have used c.server.get() (Fujitsu First Round)**

Created a widget that will fetch total count from a particular table on a click of a button.

HTML:



CSS:



Client Script:



Server Script:



1. **What will happen if we check the public checkbox on any widget or portal page?**

If we check the public checkbox on any widget or portal, then that widget or page will be visible to any user without logging into the instance.

For example: i have widget which has list of html fields and i have added on form. so if user will hit on the portal URL without logging then he can see those html fields because the widget we have checked as public

1. **Widget to Widget Communication (Cognizant Technical Round)**

Widget can communicate with other widget on the same page or the widgets that are embedded into a widget i.e Child and Parent Widgets.

$emit and $broadcast are two functions that are use to pass data from child to parent or parent to child widget.

$emit – Emit Function is used to pass data from Child Widget To Parent Widget

$broadcast – Broadcast function is used to pass data from parent widget to child widget or to pass data from one widget to another widget on same page.

1. **Embedded Widgets (Cognizant Second Round)**

One can embed widget inside another widget by using the following code:



Embedding a widget helps in maintaining the code and makes the code look cleaner. Plus, the functionality can be re-used again in any other widget if required.

To pass data to child or embedded widget one can use the $broadcast function and to pass data back to parent widget, one can use the $emit function.

1. **ACLs (Fujitsu First Round)**
2. **SLAs**
3. **Data Policies**
4. **Transform Maps**
5. **Workflows and Flow Designer (Cognizant First Round)**
6. **Why to use Scripted Rest APIs? (Cognizant First Round)**

Scripted Rest APIs are used to create custom web services APIs endpoints that can be used by other system to fetch data from ServiceNow Instance.

Scripted Rest Apis follows REST structure and once can customize them to use different conventions.

One can define service endpoints, query parameters and headers for scripted Rest APIs

An example of Scripted Rest API to fetch HR Case for Incoming Email ID. The Active Flag is optional.



API Call:



Response From ServiceNow:



1. **MID Server**
2. **Rest Messages (Cognizant First Round)**

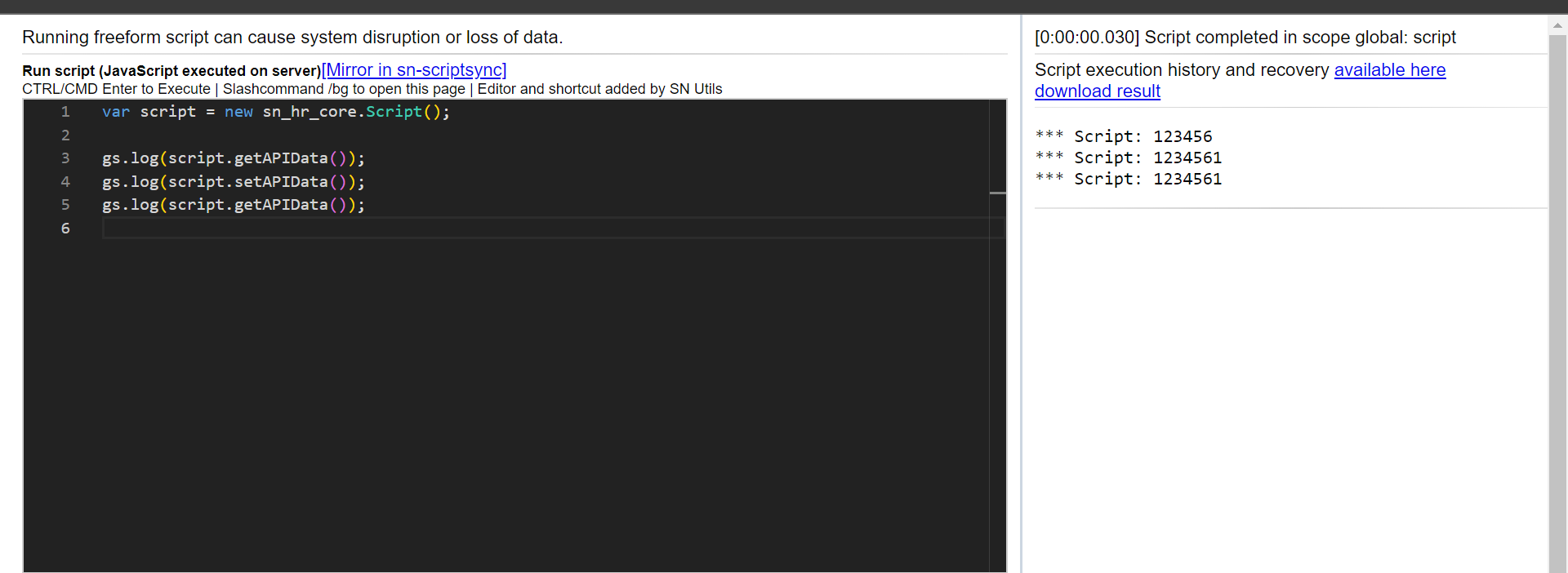
REST messages are used to send HTTP requests to remote systems. This allows you to integrate ServiceNow with other applications or services by consuming their APIs. REST messages can be configured to use different HTTP methods like GET, POST, PUT, DELETE, etc., and can also handle authentication methods such as Basic Auth, OAuth, API keys, etc.

1. **Order Guide and Catalog Items**
2. **Domain Separation? (Fujitsu First Round)**
3. **Custom Applications (Fujitsu First Round)**
4. **What is current.update() and why should we avoid it?**
5. **What is Class.Create() method used for in Script include? (Cognizant First Round)**
6. **What is initialize:function(){} Is used for in Script Include?**

The initialize function can be used to initialize global variables in the script include.  
These global variables will be accessible across the script include and any method or function can use these variables directly.



Calling and Output:



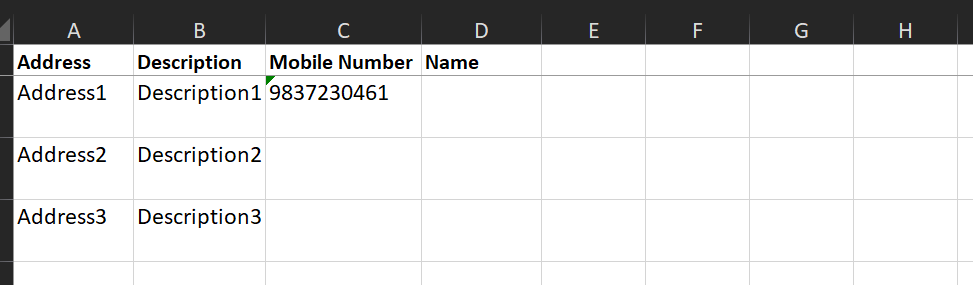
1. **What is Data Policy?**

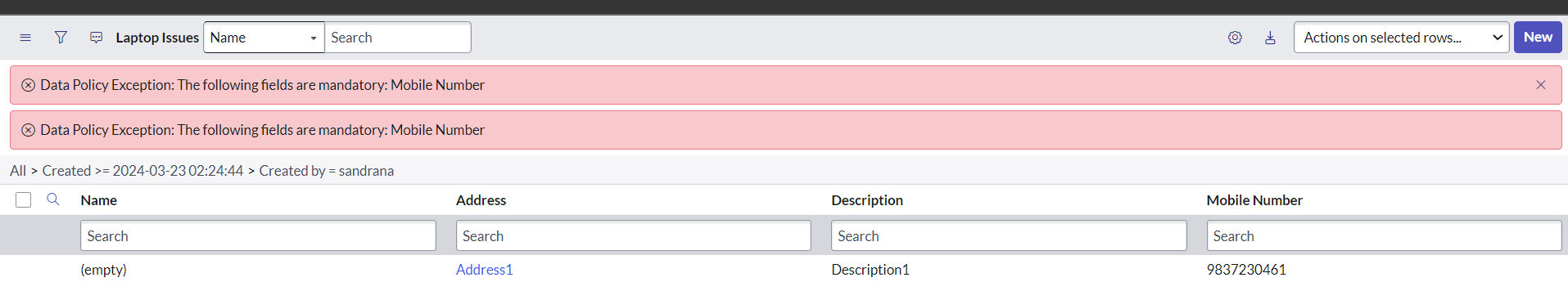
Data policies are same as of UI policy in functionality. They also provide a mechanism to make a field mandatory or read only. However, where UI Policy works on the client side, data policies work on the server side.

Another difference is that the UI Policy can make field hidden while in Data Policy that feature is not present.

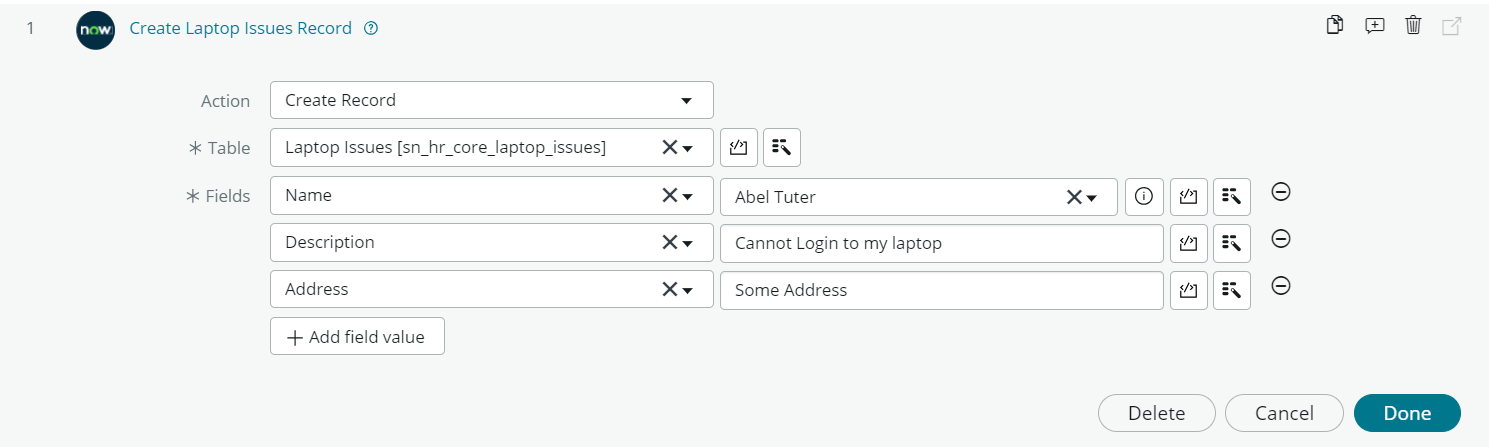
Data policies ensure that the value of a field is populated (if the field is mandatory) during any import or transform or any integration.

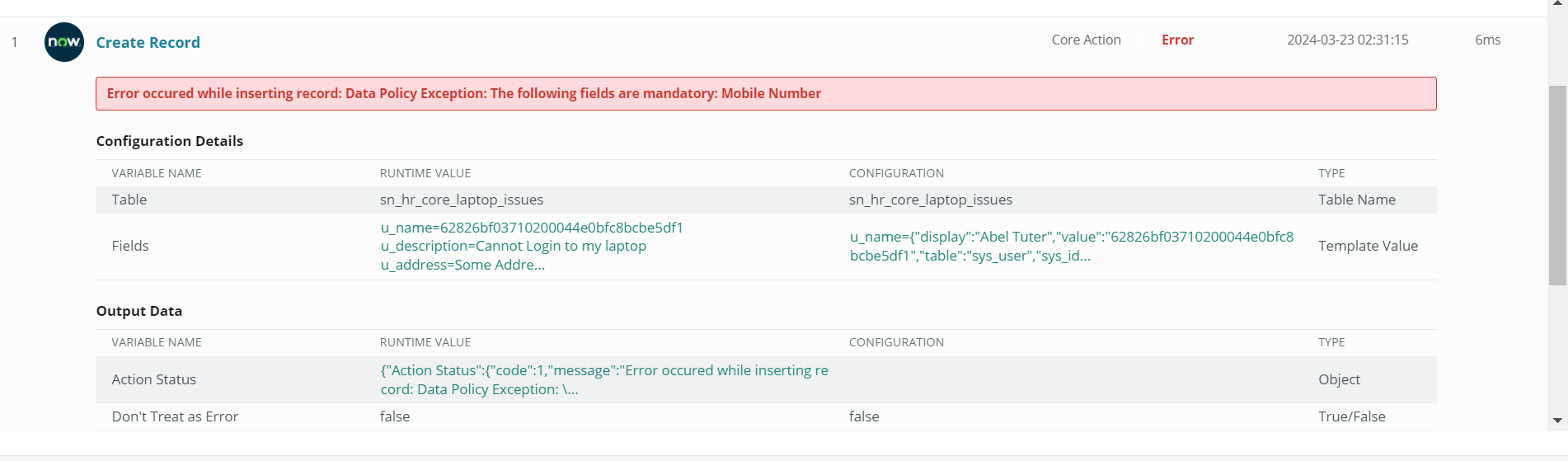
Let’s consider this scenario:  
There is a table Laptop Issues and the field Mobile Number is marked mandatory using Data Policy  
1) When you try to import an excel with some data and mobile number is not populated in that excel, then the upload will fail.





2) Now, lets assume that the data is being populated from Integration via flow designer and the mobile number is not populated there as well, then the flow will crash and the record will not be created.





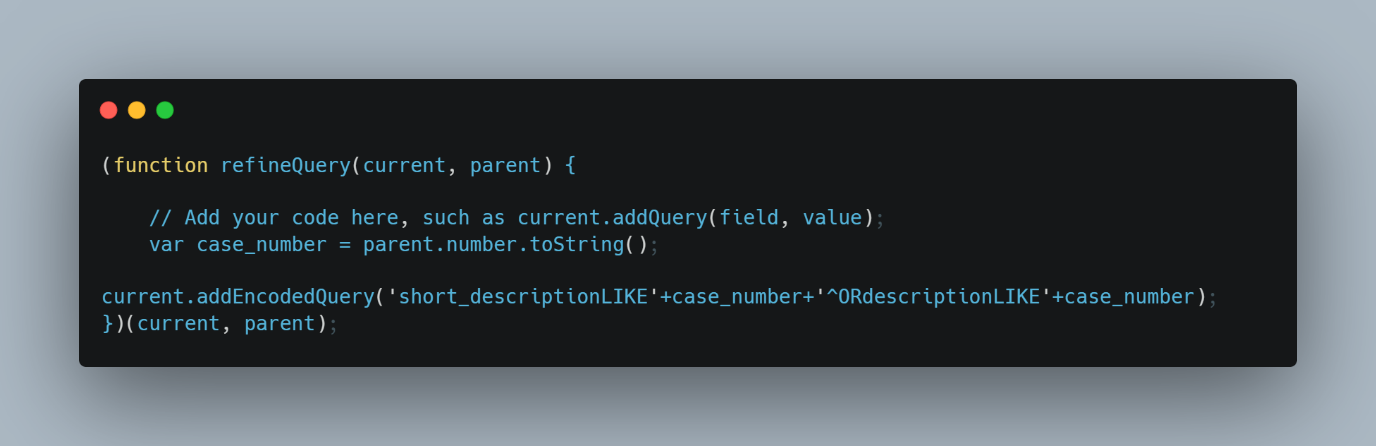
1. **Integration between two ServiceNow instance HR Profile.**
2. **Difference between Opened for and Subject Person**

The subject person field refers to any user who is affected by any request or incident whereas the Opened For field refers to any user on whose behalf a case or request was created.

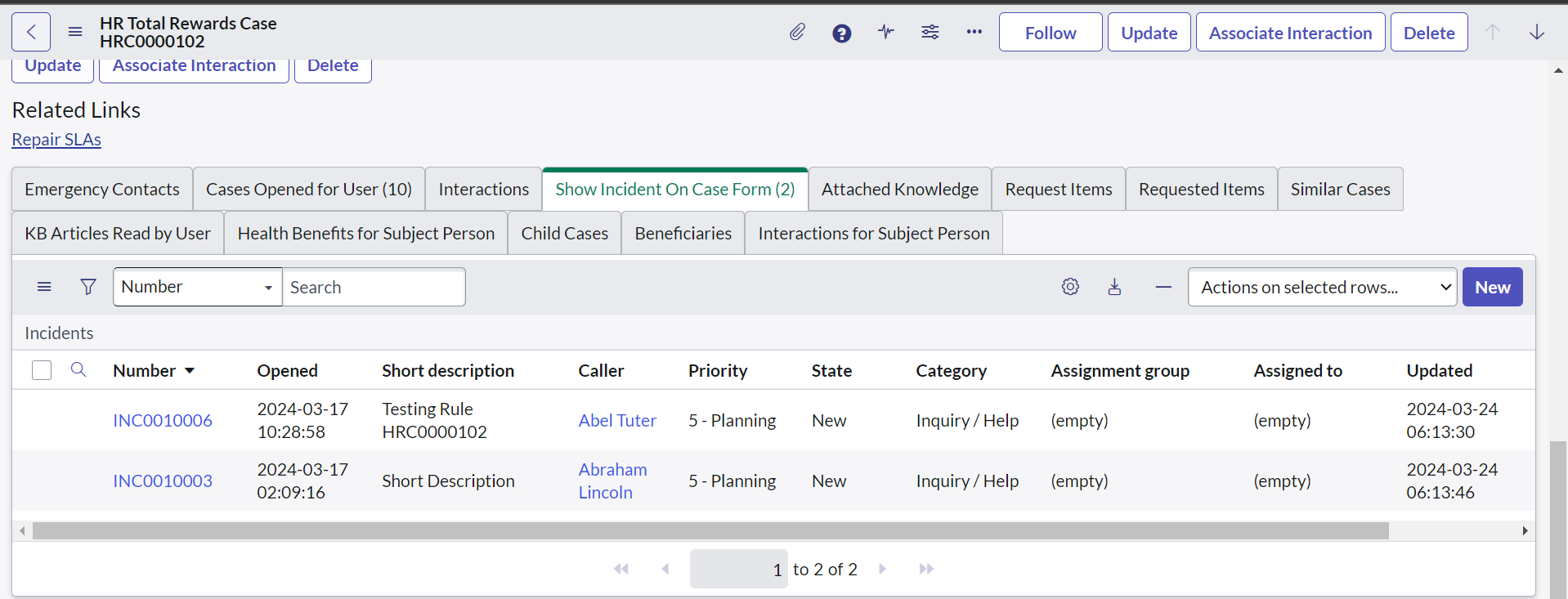
1. **Given a dynamic payload, fetch a particular email (sand@man.com) and get the user’s name and employee number from that email. (Cognizant First Round)**

****

1. **Add a related list on HR Case Table to all incidents that are created for that HR Case. (Fujitsu First Round)**
2. Navigate to System Definition -> Relationships
3. Create a new Record with name “Incident For Case Record”
4. Applies to Table will be HR Case
5. Queries from table will be Incident
6. Add a script to fetch incidents related to HR Case.



1. Navigate to any HR Case record and click on hamburger icon and select configure-> related list
2. Select the Relationship just created from the slush bucket and arrange accordingly on the related list tabs
3. Save the form and you will see the new related list on the case.



**(Cognizant Technical Round Questions [22 – 32])**

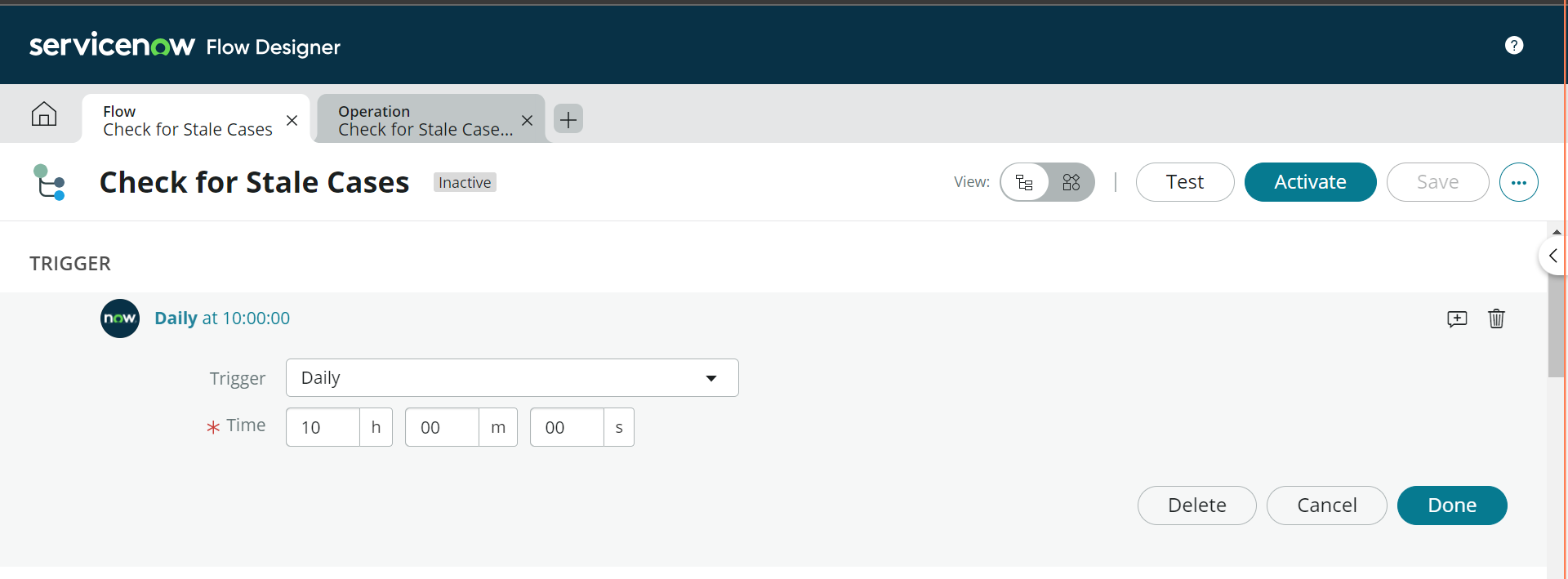
1. **In Transform Script, what is the difference between the OnAfter and**

**onComplete function difference**

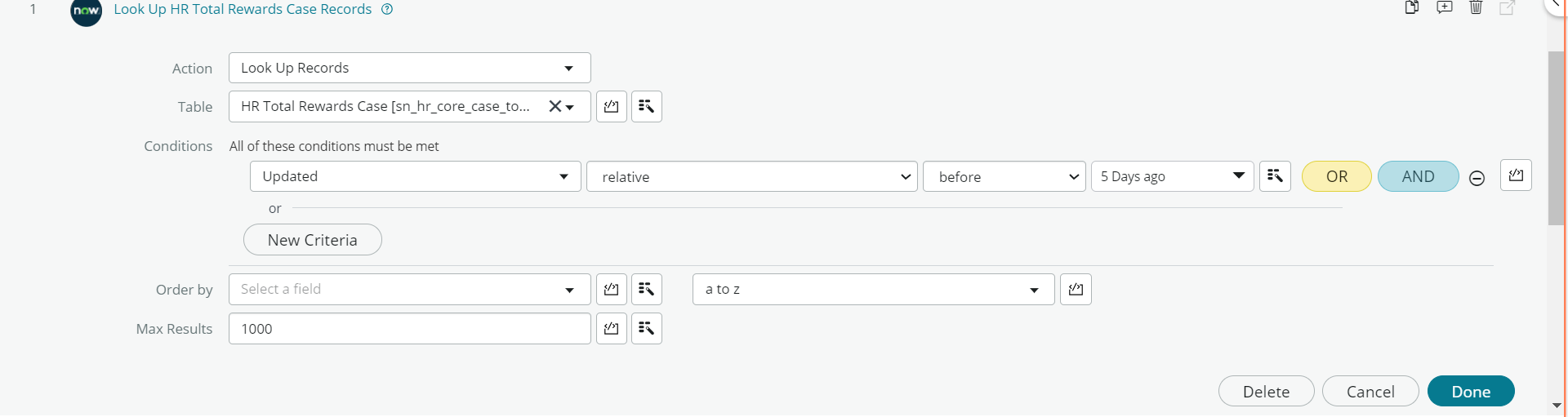
1. **Create a flow that will send notification to assigned to when the case is not updated for last 5 days**

Follows these steps:

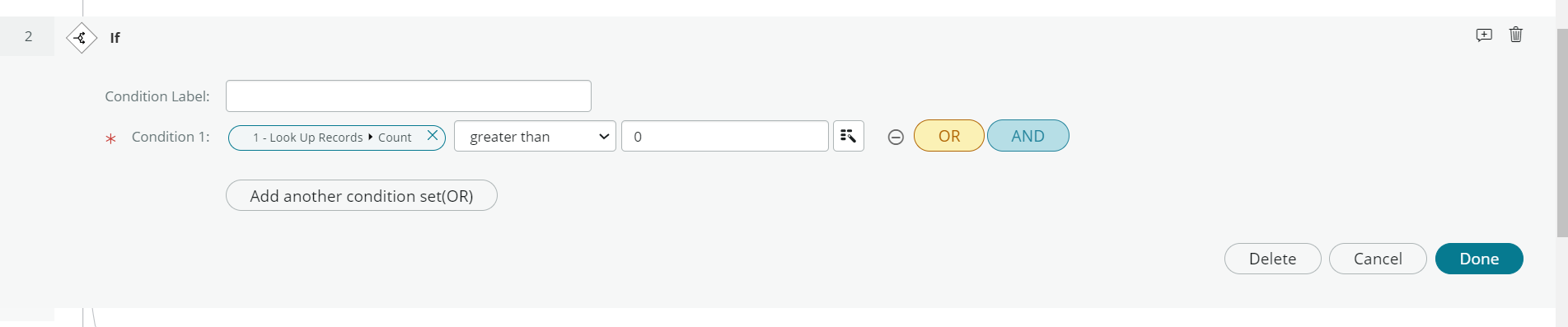
1. First Create a new Flow with Daily Trigger. Select Any time.



1. Modify a look up records action to get all the cases that were updated 5 days ago.



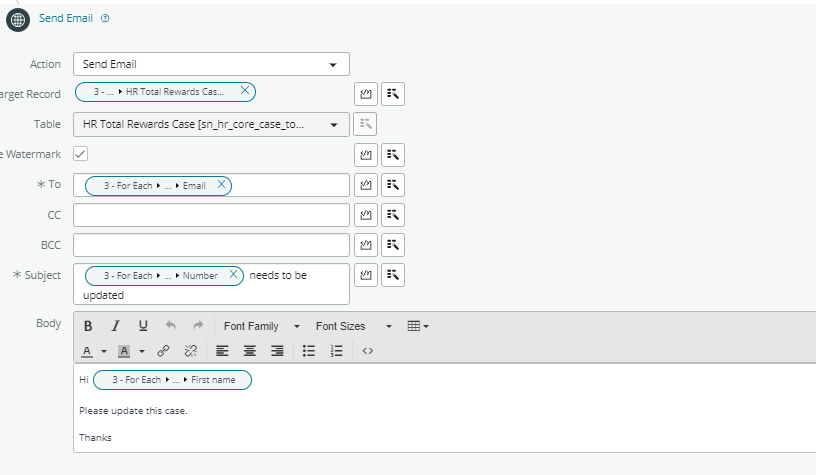
1. Check if such records exist



1. Add a loop to go send email for each case.

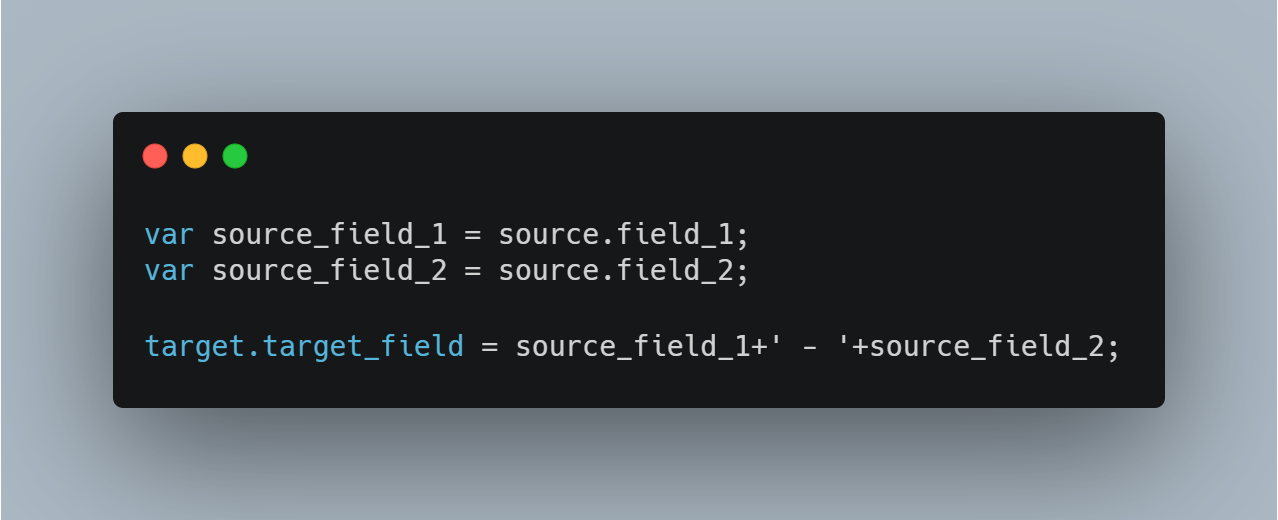


1. Set To, Subject and Body of the email



1. Remember, the receiving (To) will be assigned to of the case and the subject should contain the case number
2. **How can we add two values from source field into one field in target table. The target field is a string field.**

Given that the target field is a string field, one can concatenate the values from 2 field and merge into a single field. If required one can separate the two values with some symbol to be more precise.



1. **On an inbound integration, when will the business rule, on insert or on update**

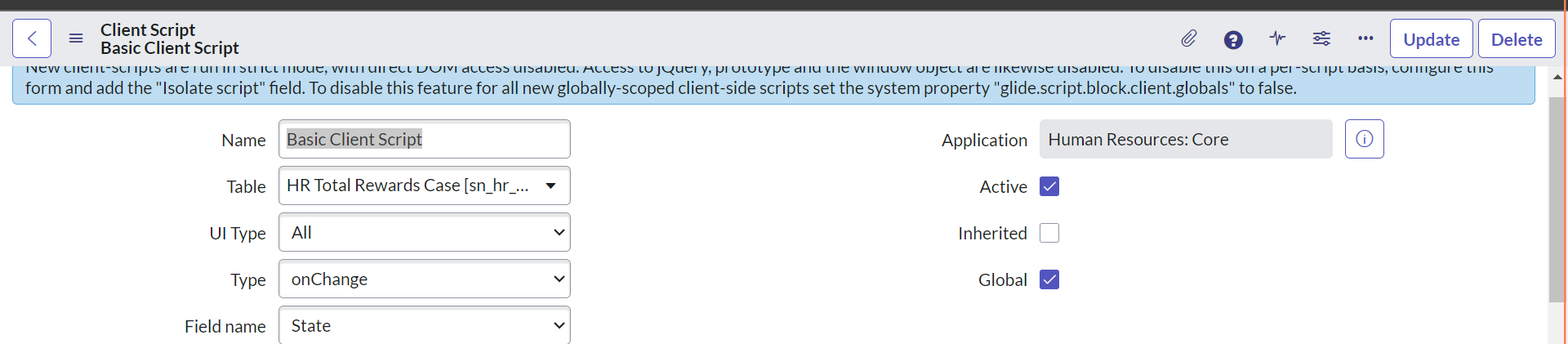
It depends whether you are updating a record or inserting a new record using inbound integration. So, in an inbound integration, ServiceNow is accepting the third-party system data and manipulating its record. When manipulating the data, we will see the business rule will on the specified trigger.

1. **Given that we must write an onload and onchange script, can we write both these script in one place to avoid 2 scripts?**

Yes, we can actually write an onload script within and onchange script. This can reduce our effort and we don’t have to write another new script for onload.

Here is how it will work:

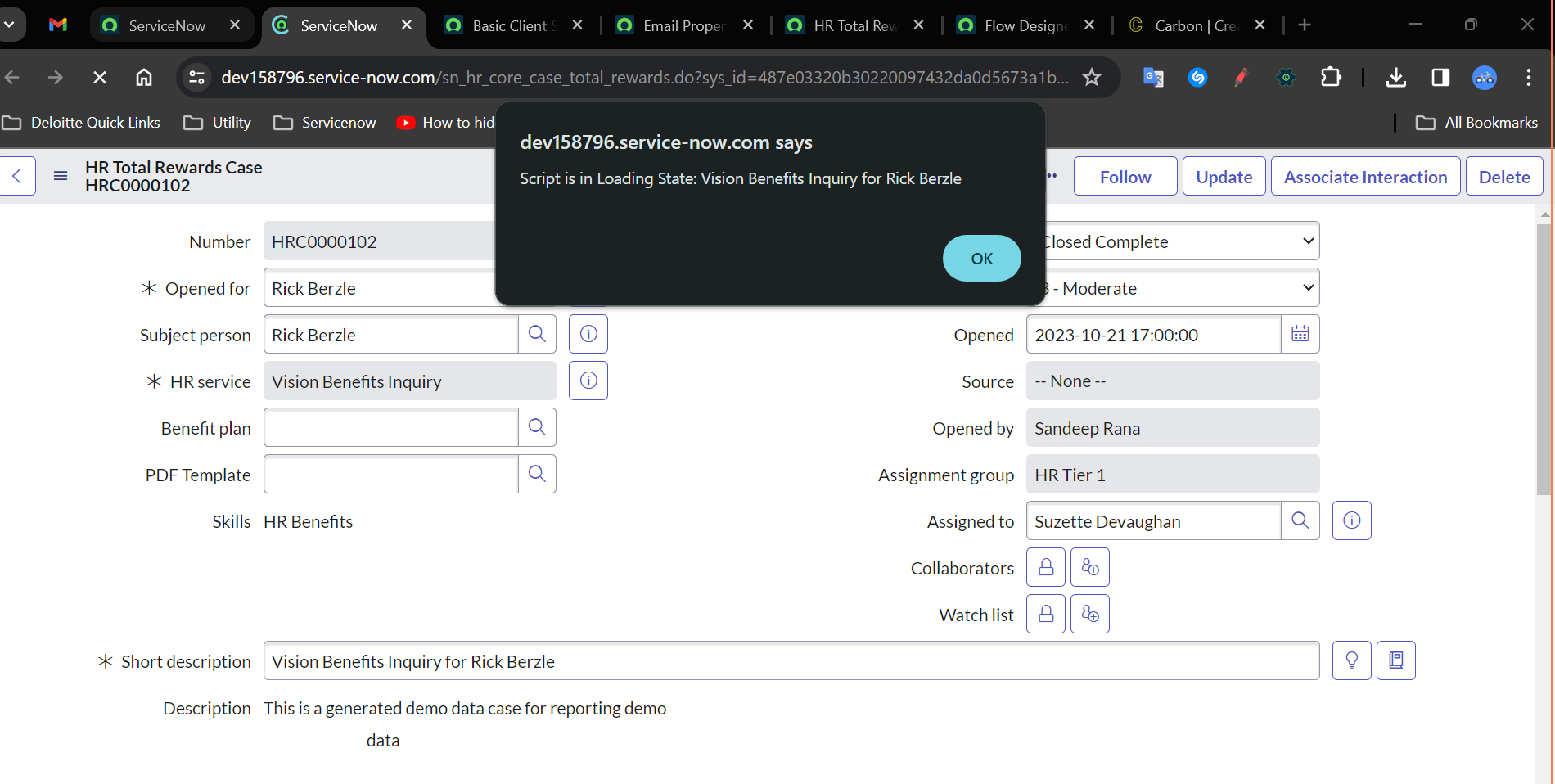
1. Create a new client script
2. Select Table Name. (Total rewards table in this example)
3. Select a UI Type (All in this example)
4. Select Type as OnChange Script
5. Select a field that’s modification will trigger the onchange script (State field in this example)



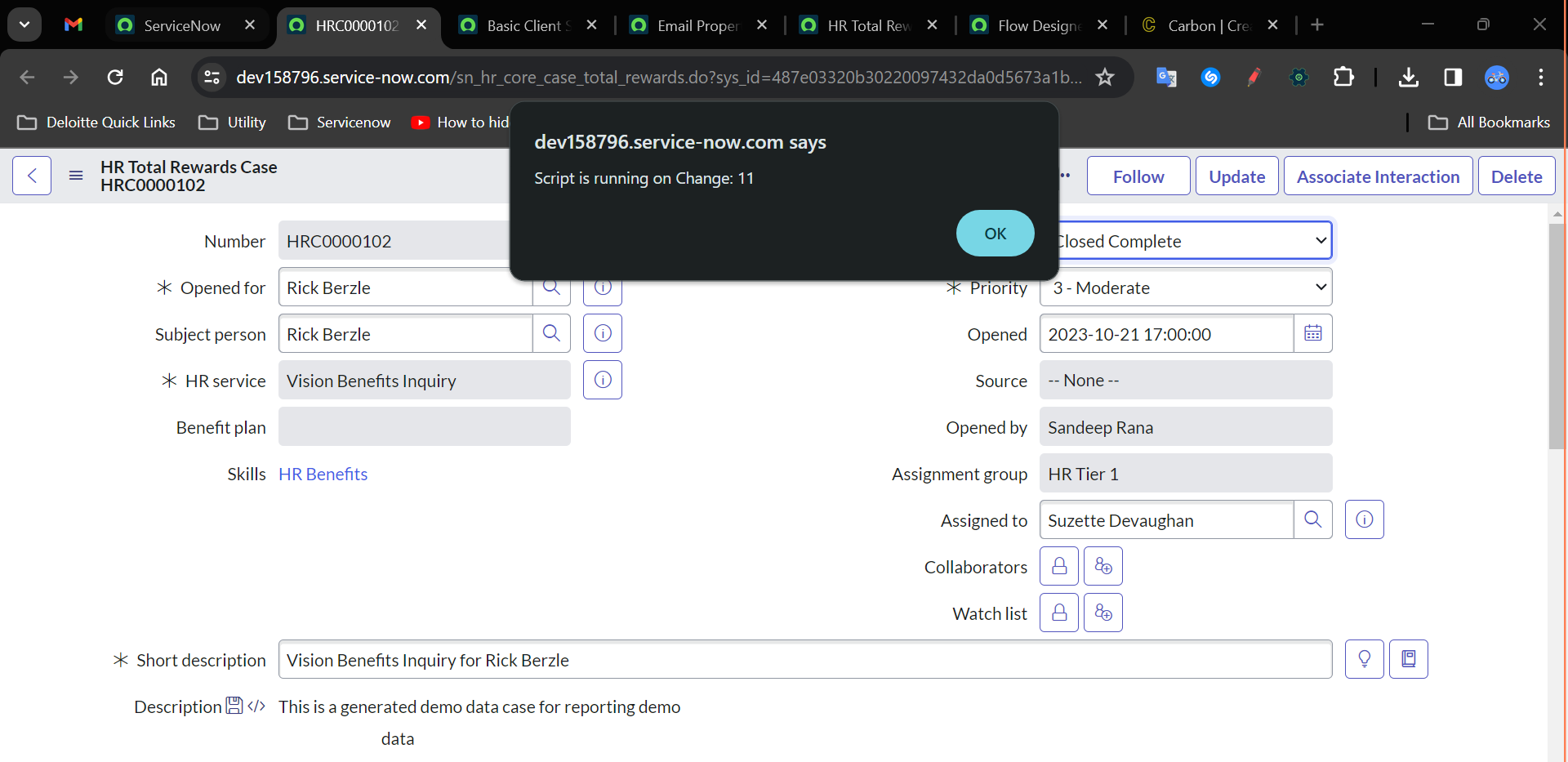
1. Write the below code.



1. Load an HR Case and see the first alert to run as the form loads.

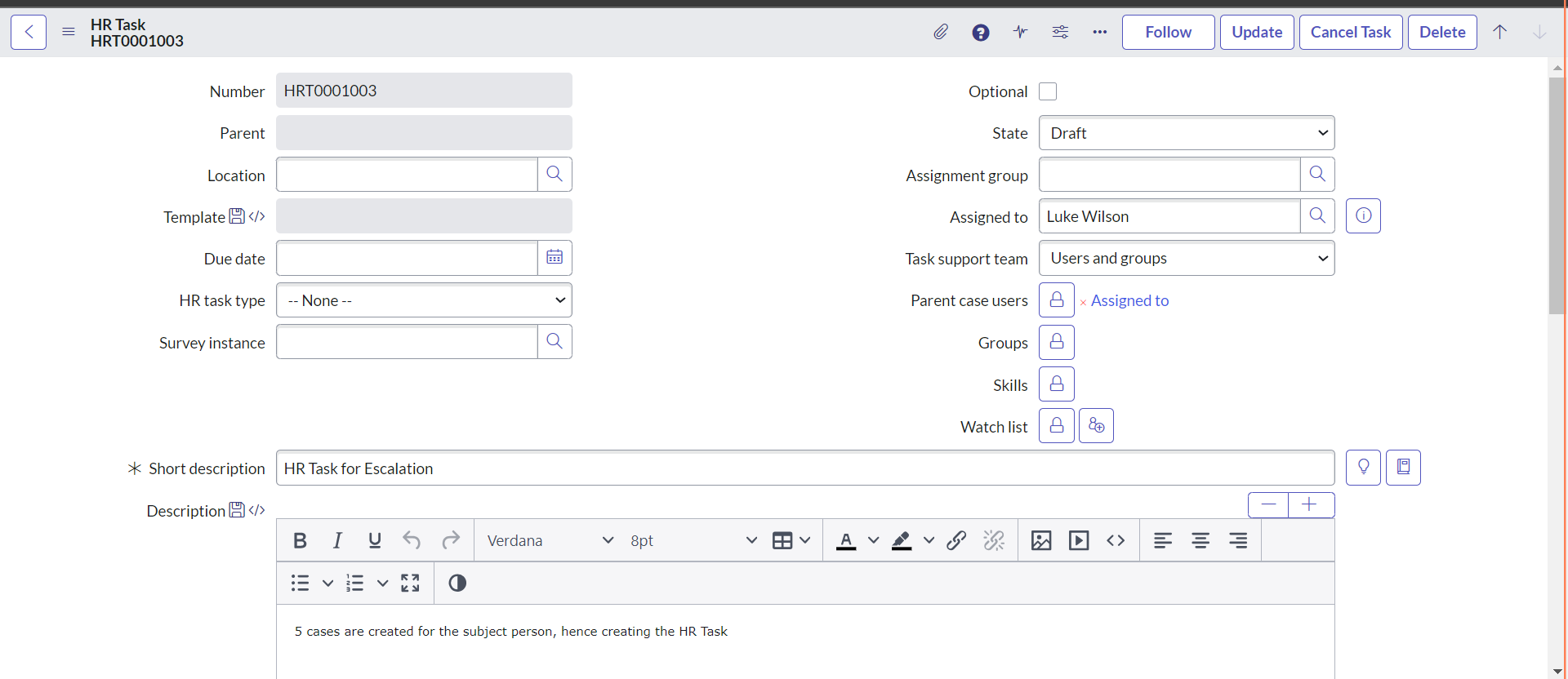


1. Change the state field and see the onchange script running.



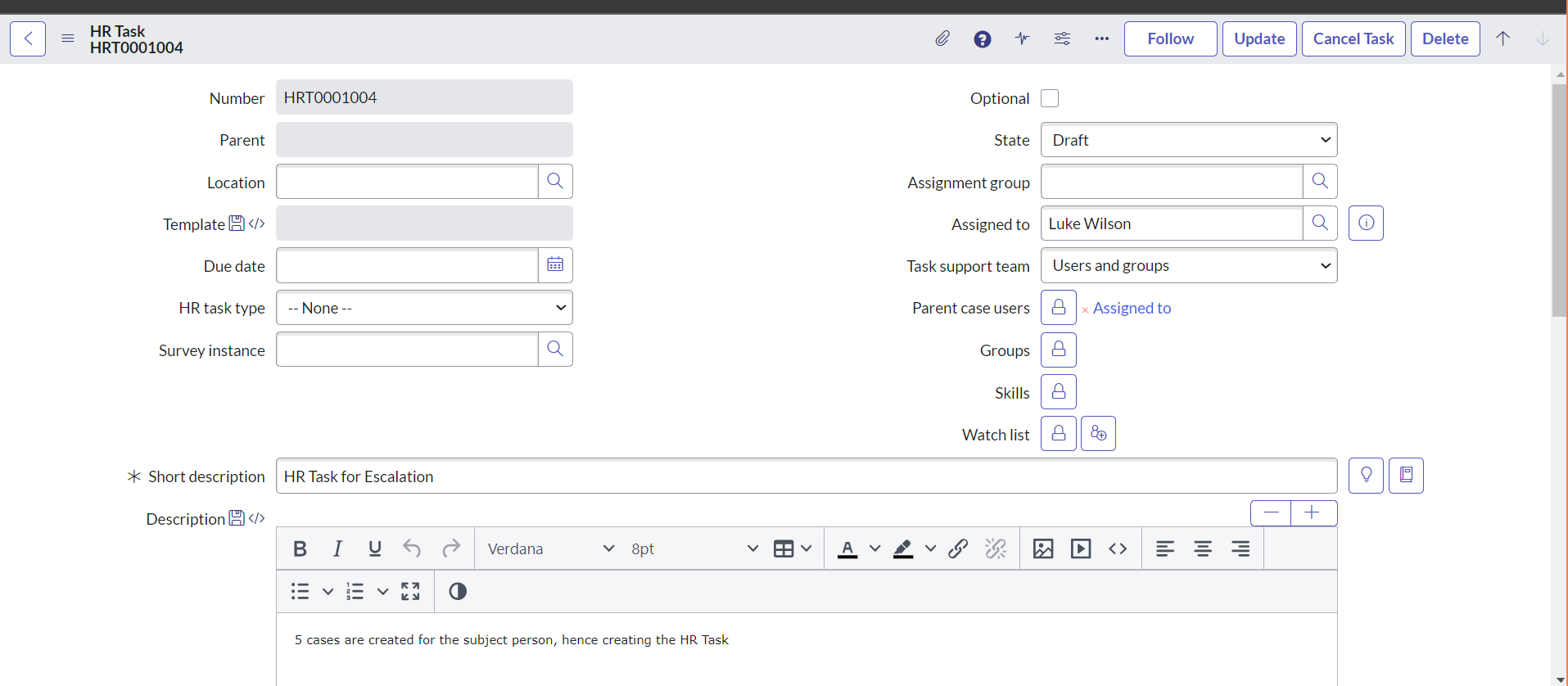
1. **GlideAggregate**
2. **Use GlideAggregate to create a HR Task if 5 HR Cases are created for a subject person. Assign the task to the subject person.**



****

1. **Without GlideAggregrate, create a HR Task if 5 HR Cases are created for that subject person. Assign the task to the subject person.**

****

****

1. **Update a record without changing the updated date or updated by**
2. **In ServiceNow how can I check if what data I have sent to third party?**

In ServiceNow, tracking and verifying the data sent to third parties requires a combination of configuration, monitoring, and auditing. Here's how you can check what data has been sent to third parties in ServiceNow:

1. \*\*IntegrationHub and API Logs:\*\*

- ServiceNow's IntegrationHub maintains logs of integration activities, including API requests and responses.

- You can review the IntegrationHub logs to see details of data payloads sent to third-party systems through integrations.

- Navigate to ` syslog\_transaction\_list.do` to view transaction logs, or check `System Logs` under `System Logs > All` for API-related activities.

2. \*\*Data Export and Import Logs:\*\*

- ServiceNow tracks data exports and imports through its data import/export logs.

- You can access these logs by navigating to `System Import Sets > Data Sources` or `System Import Sets > History`.

- Review the logs to identify any data exports or transfers to third-party systems.

3. \*\*Audit Trails and Change History:\*\*

- ServiceNow maintains an audit trail of data changes, which can help you track data sent to third parties.

- Navigate to individual records (e.g., incident records, task records, etc.) and check the `Audit` or `History` related links to view the change history.

- Look for changes indicating data sent to external systems or third parties.

4. \*\*Scheduled Data Exports and Reports:\*\*

- If you have scheduled data exports or reports that are sent to third parties, you can review these configurations.

- Navigate to `System Scheduler > Scheduled Jobs` to view scheduled data exports or reports.

- Review the configurations to identify what data is being exported and where it is being sent.

5. \*\*Custom Applications and Workflows:\*\*

- If you have custom applications or workflows that involve sending data to third parties, you can check these configurations.

- Review the workflows, business rules, and scripts associated with the custom applications to identify data transfers to third parties.

- Check any logging or auditing mechanisms implemented within the custom applications to track data sent to third parties.

6. \*\*Third-Party Integration Documentation:\*\*

- Refer to the documentation or configuration details of each third-party integration within ServiceNow.

- Review the integration settings, mappings, and transformations to understand what data is being sent to third parties.

- Ensure that data sharing agreements and compliance requirements are documented and accessible for reference.

By following these steps and leveraging ServiceNow's logging, auditing, and monitoring capabilities, you can check and verify what data has been sent to third parties. It's important to regularly review and monitor these activities to ensure compliance with regulations, data privacy policies, and security requirements.

1. **What is the use of the Message field on the client script?**

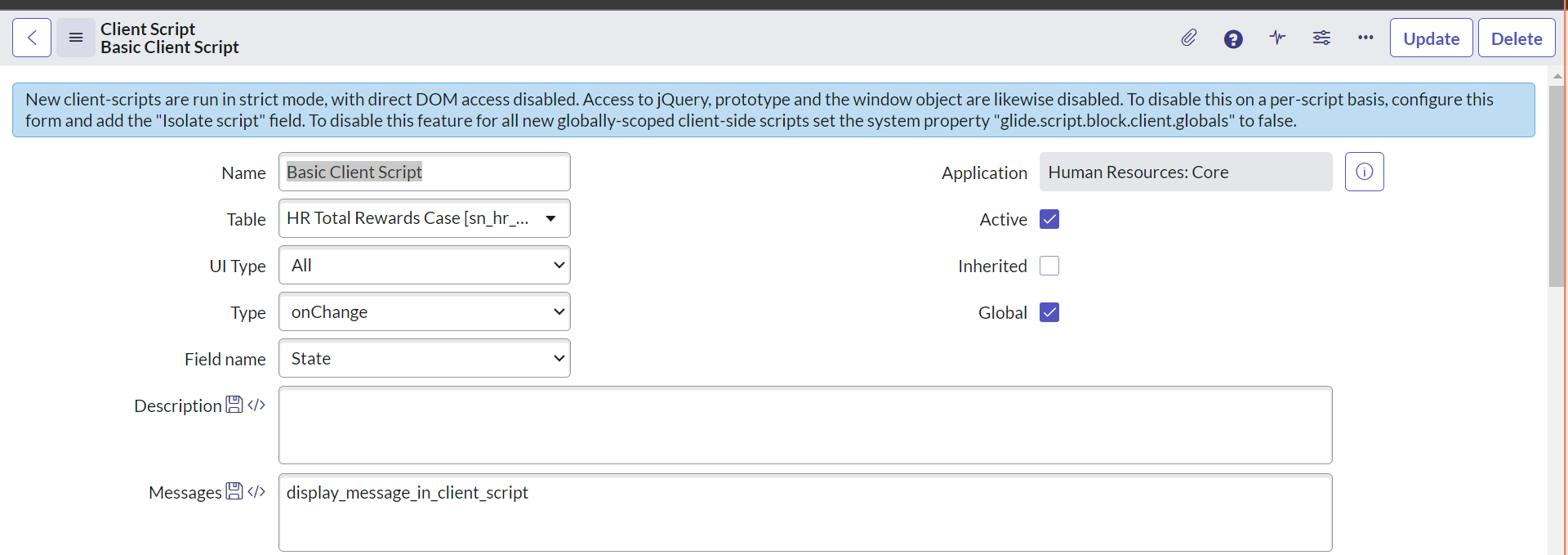
The message field is a multiline field in the client script and it is used to display messages on the form. You can use this field for Internationalization or to not hardcode messages within scripting (could be used for English language as well).

Within this field, you would mention the sys\_ui\_message record (the key). In the Client Script, you could reference that message by using getMessage()

1. Navigate to system UI -> Messages and create a new message



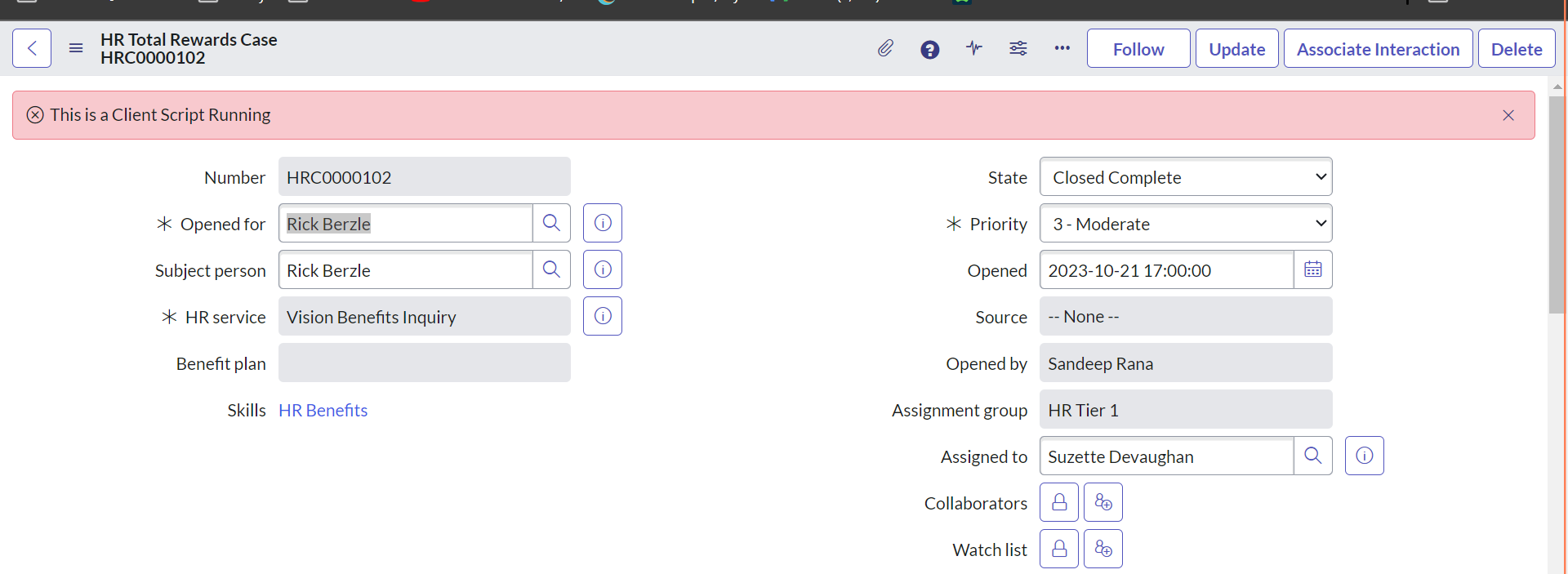
1. Create a client script
2. In the message field , add the key i.e display\_message\_in\_client\_script



1. Use the below script



1. Load the form and see the error message box displaying because we have used addErrorMessage Function.



**(Virtusa First Technical Round Questions [33 – 36]**

1. **Transform Scripts**
2. **Scheduled Data Imports**
3. **Server-Side UI Action**
4. **Use Client and Server Script both in One UI Action**