1. **What is Global Business?**

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?**

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

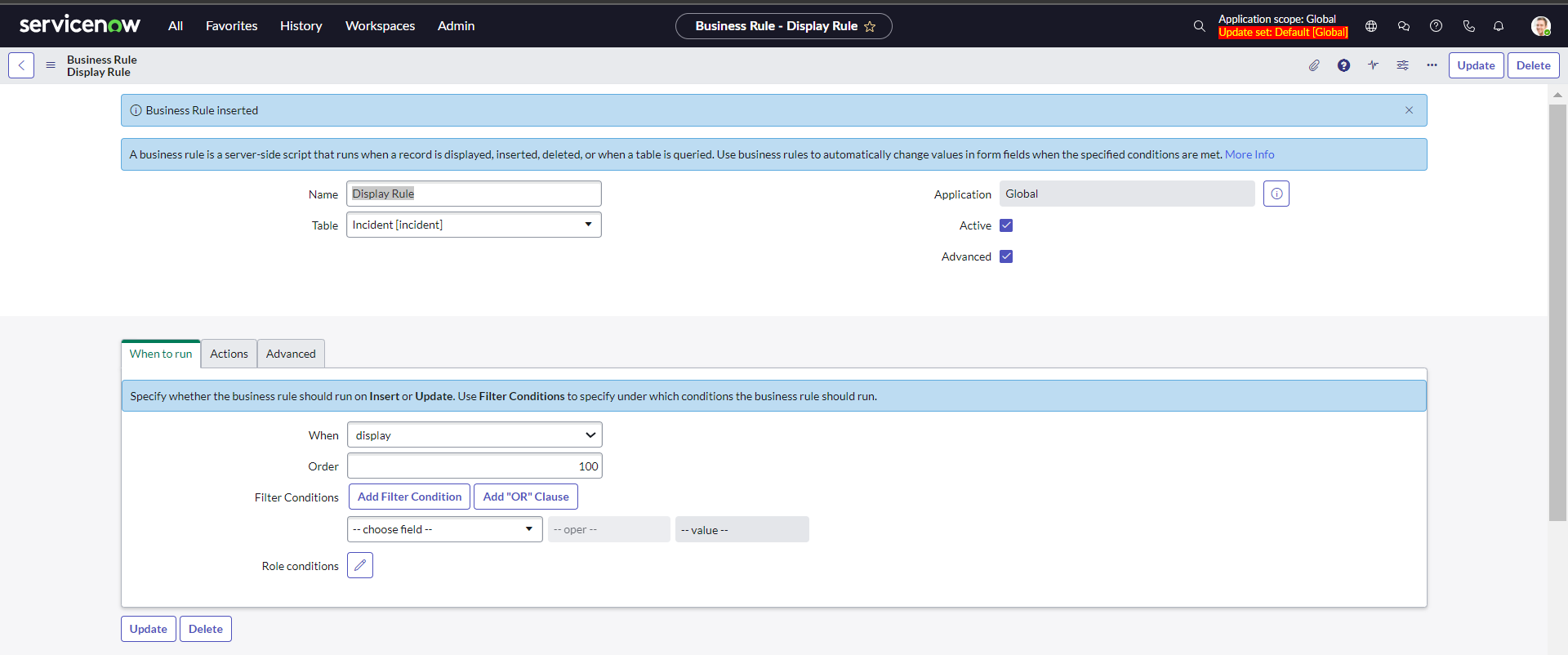
1. **What is Display Business Rule?**

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

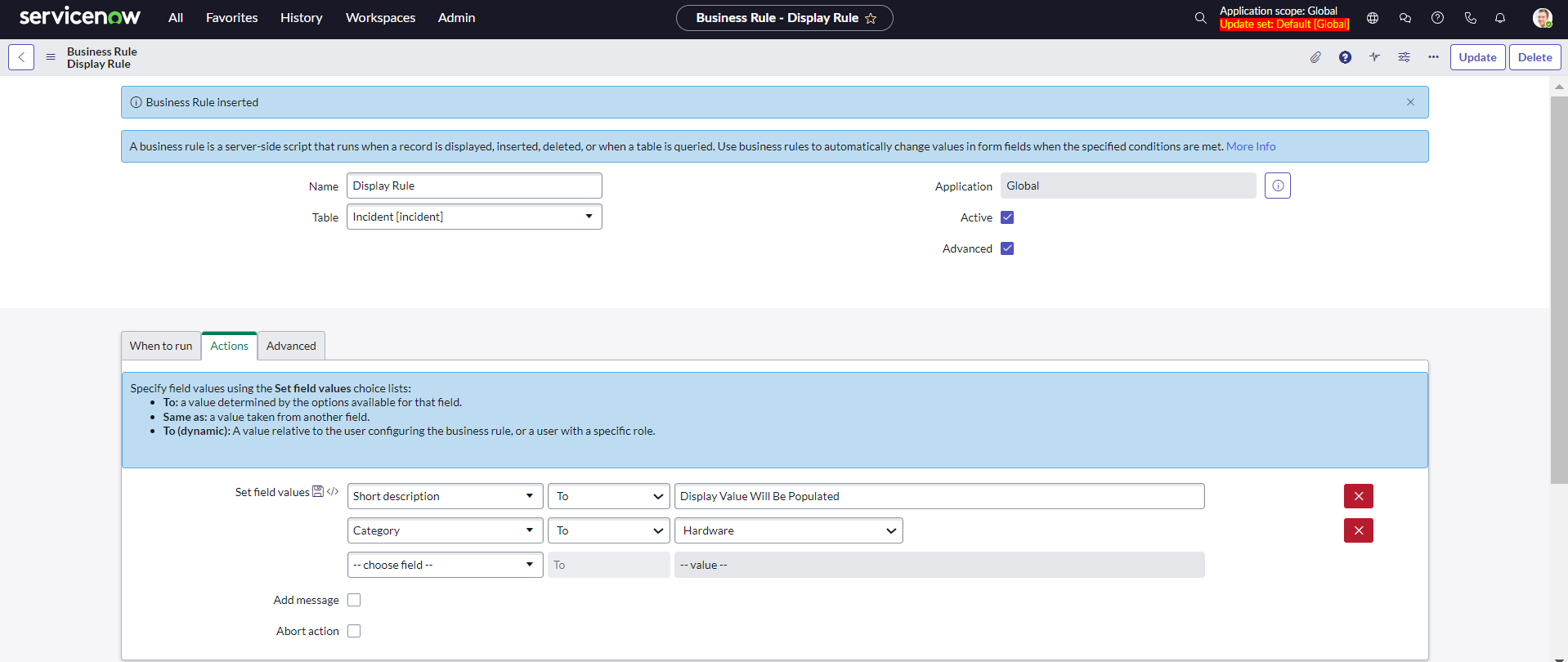
It can be used to pre-populate some fields like category, short description and etc, before the form is displayed to the user.

To create a Display Business Rule: -

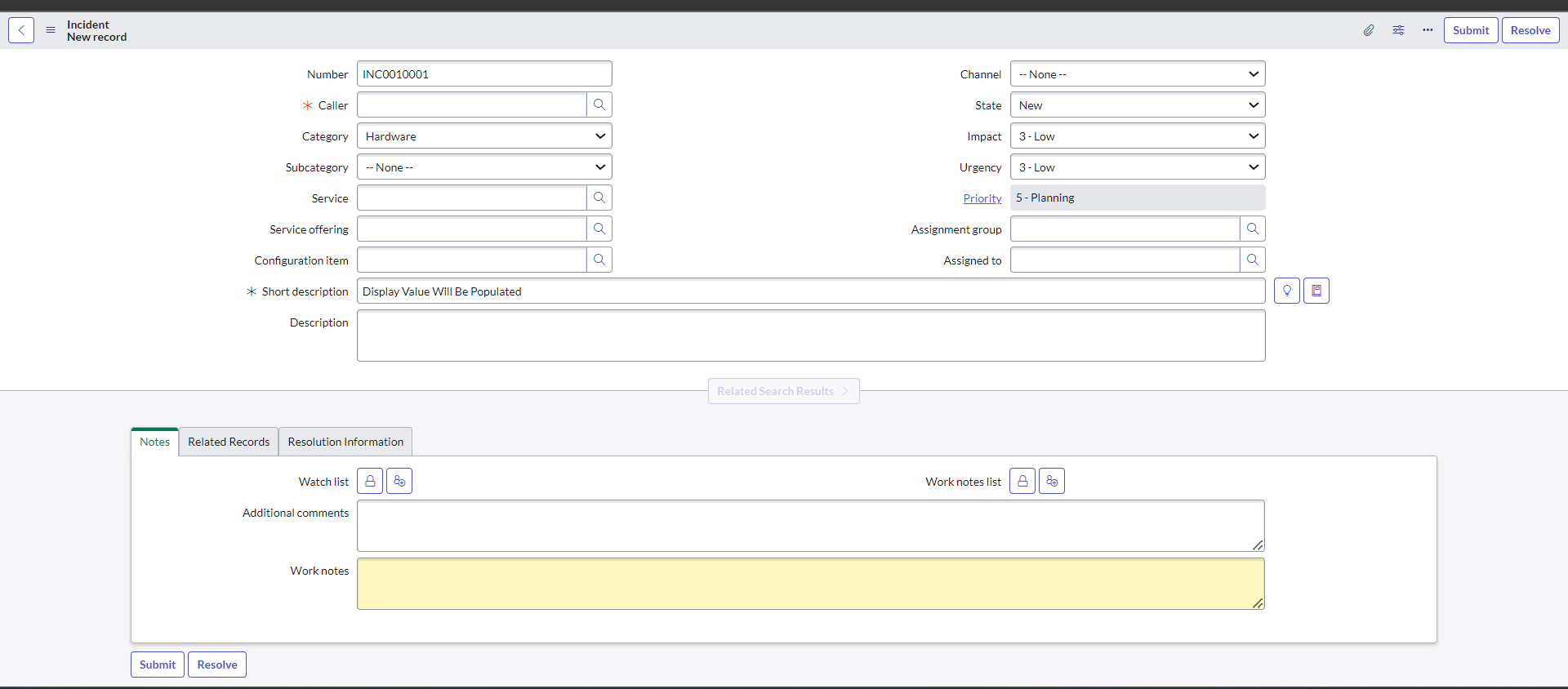
1. Create a new Business Rule and Select and Table.
2. In When to Run Section, select when to Display and select filter conditions if required.



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.

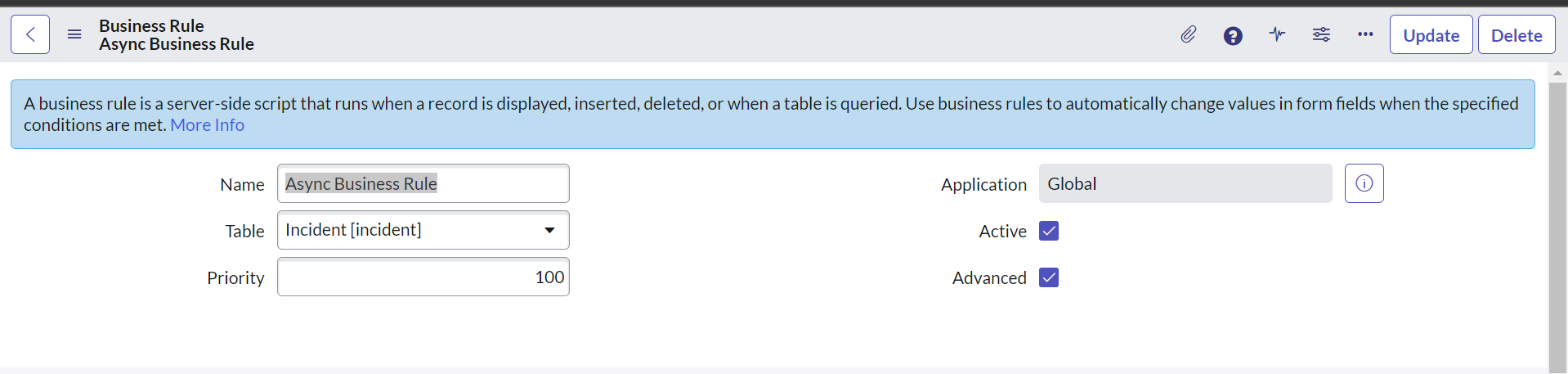


1. **What is Async Business Rule?**

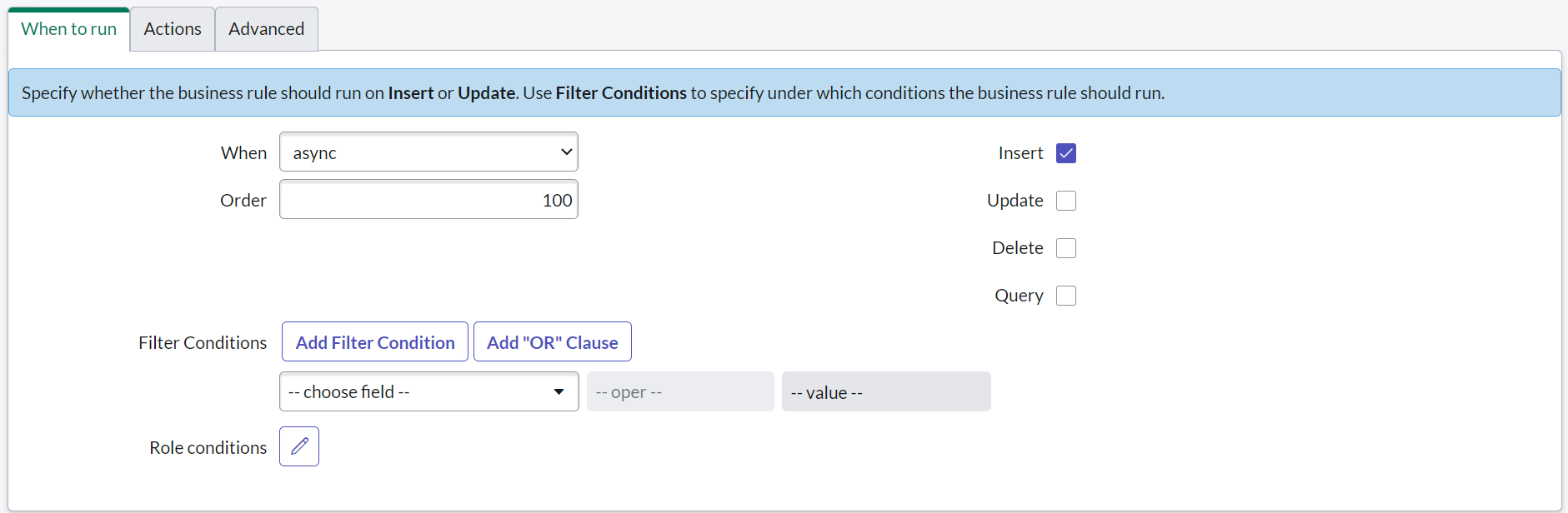
Asynchronous Business Rules (ABRs) run after database commits in the background, simultaneously with other processes. This allows ServiceNow to return control to the user sooner, but may take longer to update related objects. ABRs are similar to **After Business rules**. The only difference is that **After** 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:

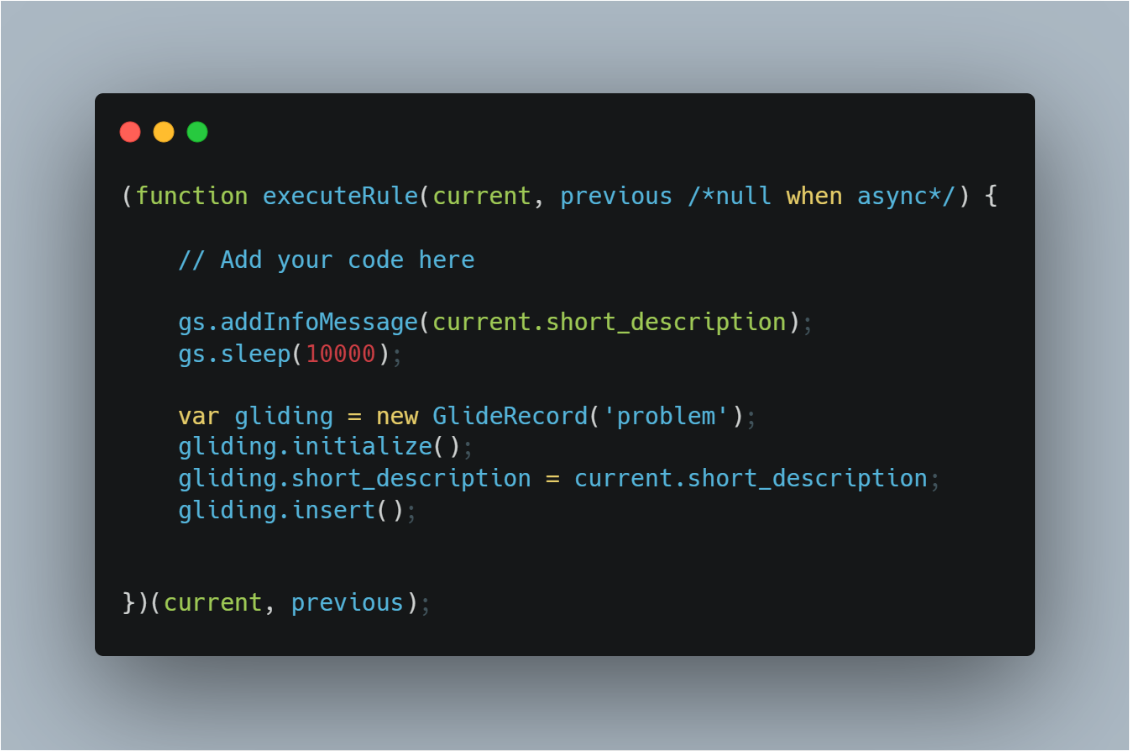
1. Create a new Business rule on Any Table and check the Advanced checkbox to bring the when section



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 gliderecord



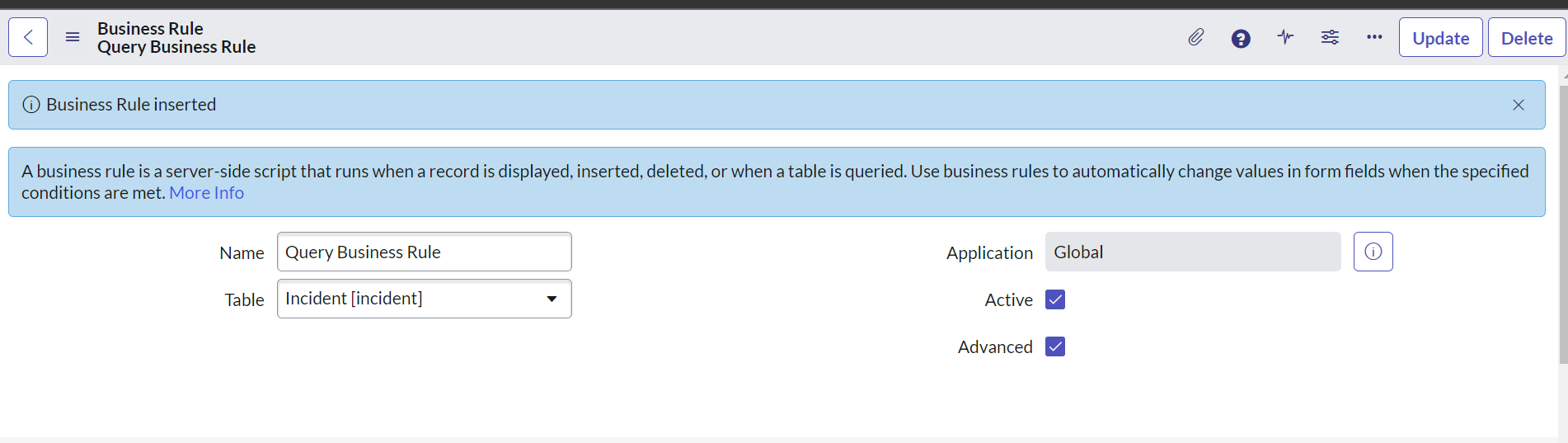
1. Create a new incident. You will see that it never sleeps for 10 seconds because of the async BR, the transaction is processing in the background.
2. **What is Query Business Rule?**

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.

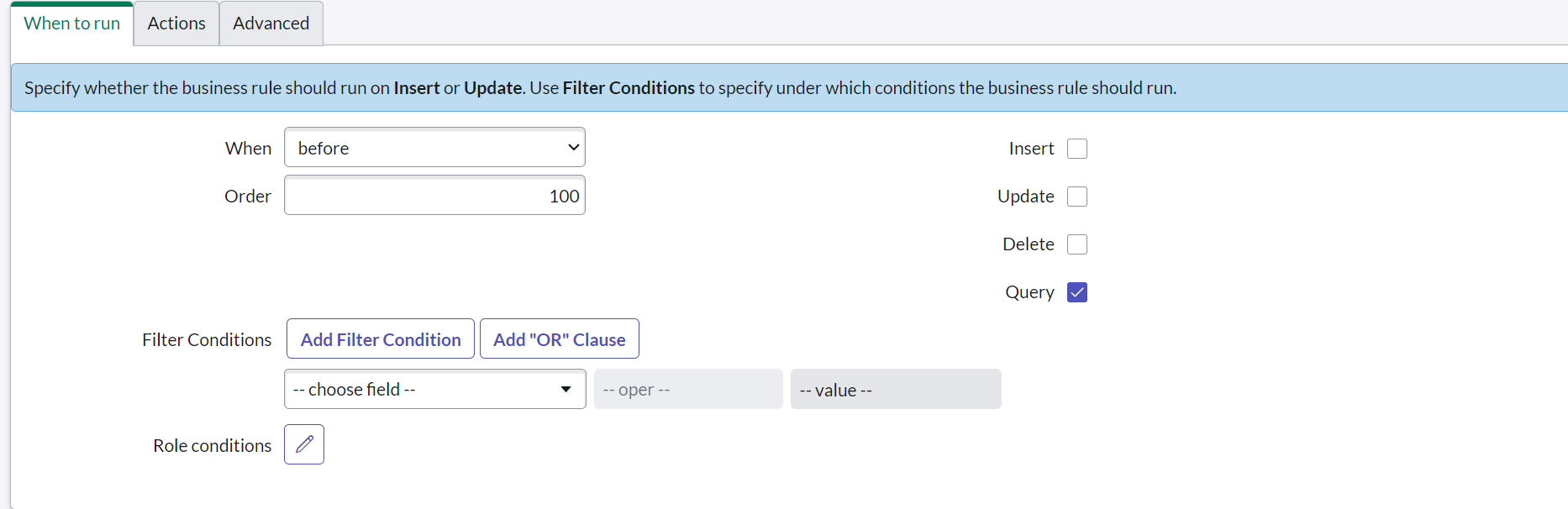
Example: If you open incident list view, then ServiceNow is querying behind the scene to fetch all the incident records and display it in list view.

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.



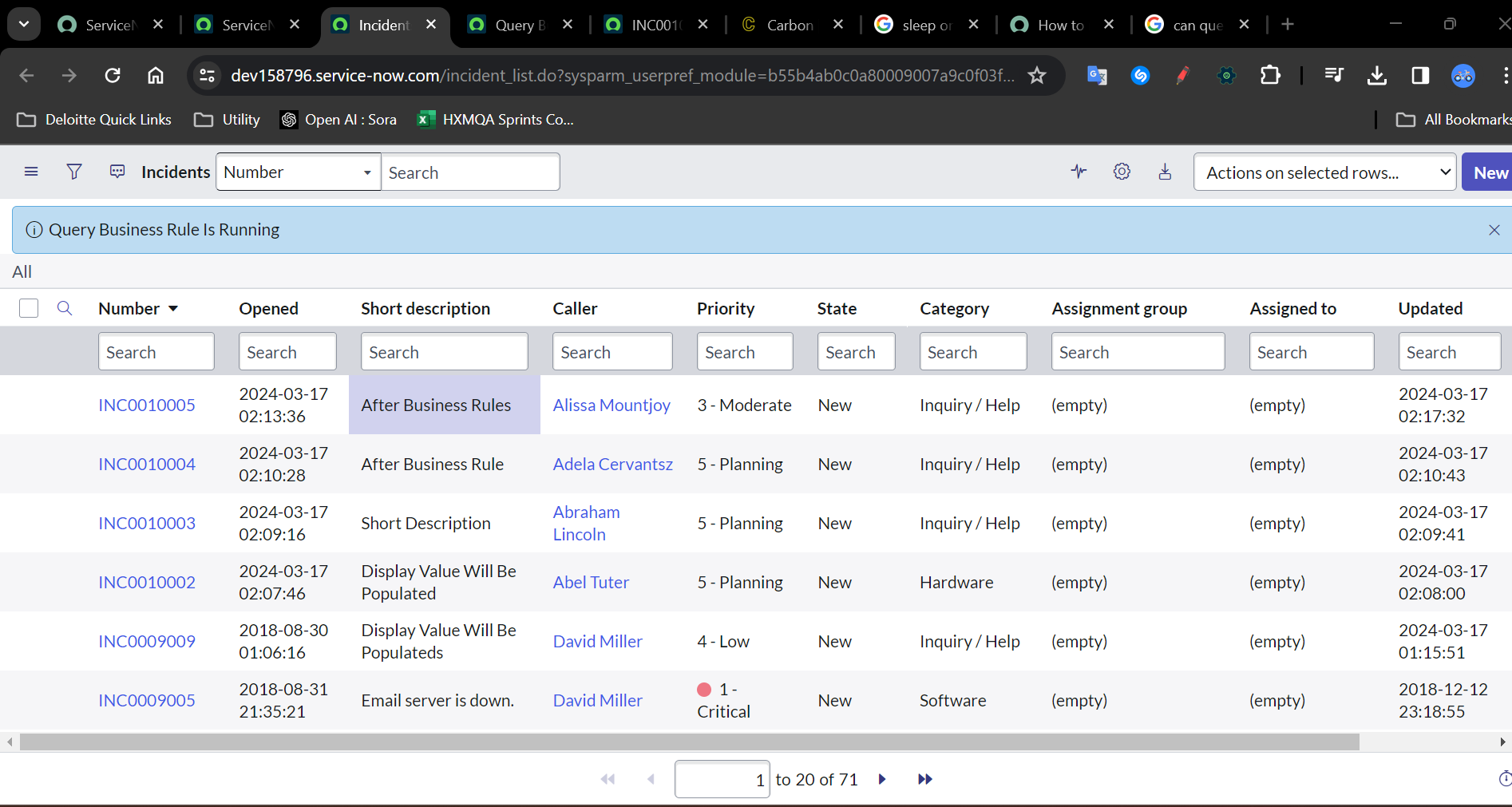
1. In when to run section, select when and check the query checkbox.



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



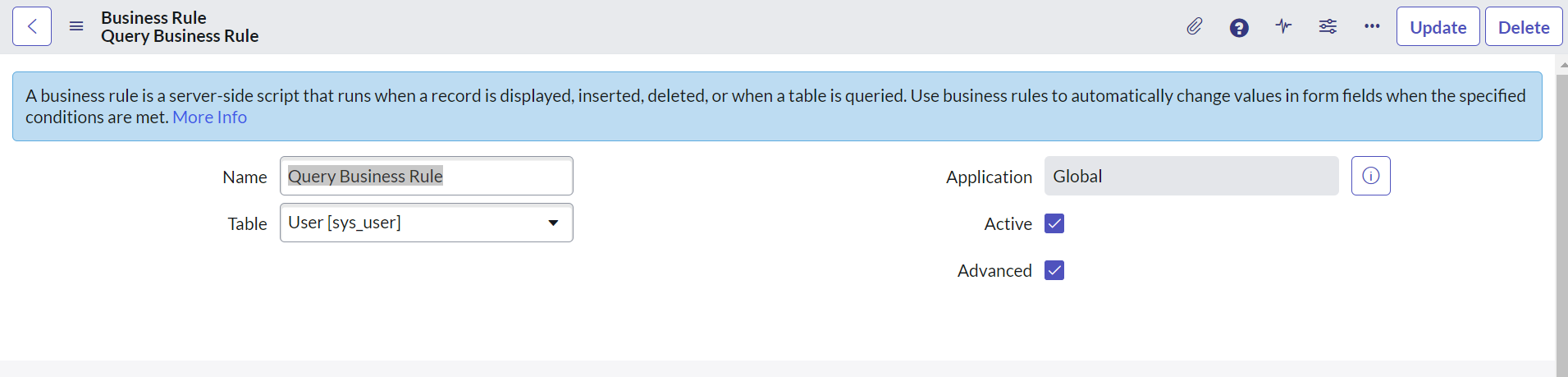
1. Save the record and navigate to the list view of the Incident, you will see the info message there.



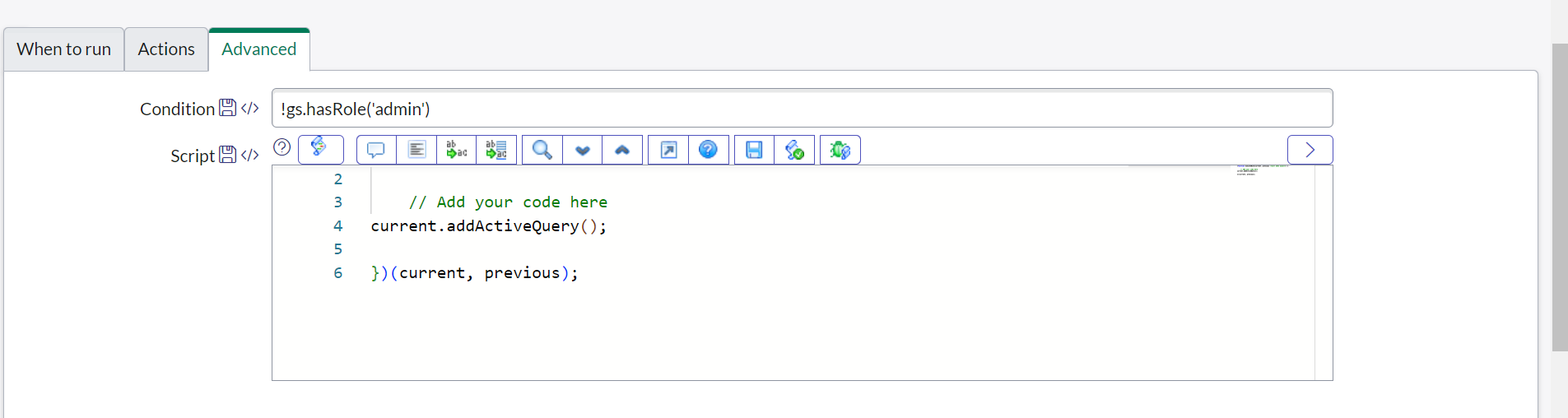
1. Try to make any change In the list view and see that the BR runs on every change.

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

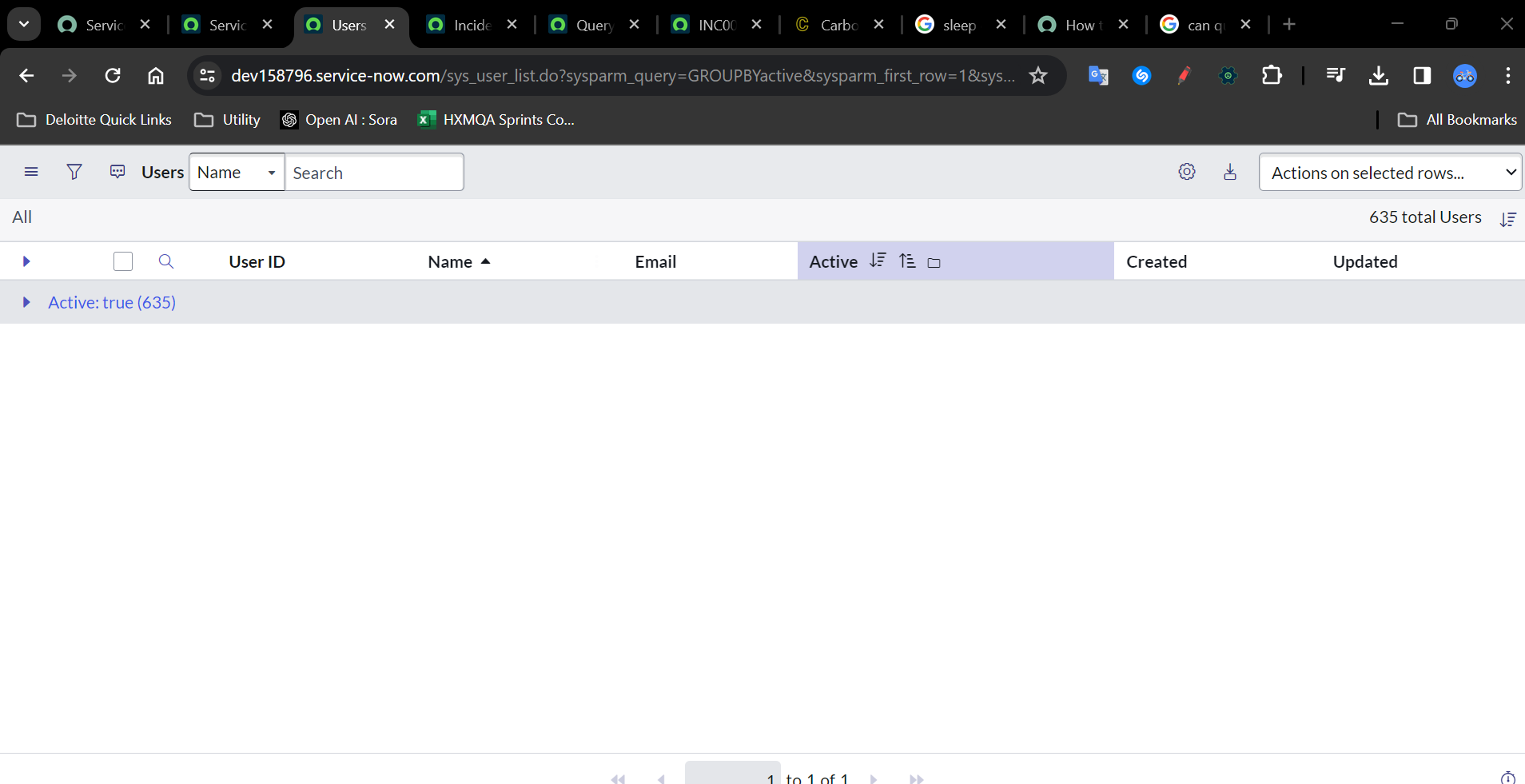
1. Change the table in the Query BR to user.



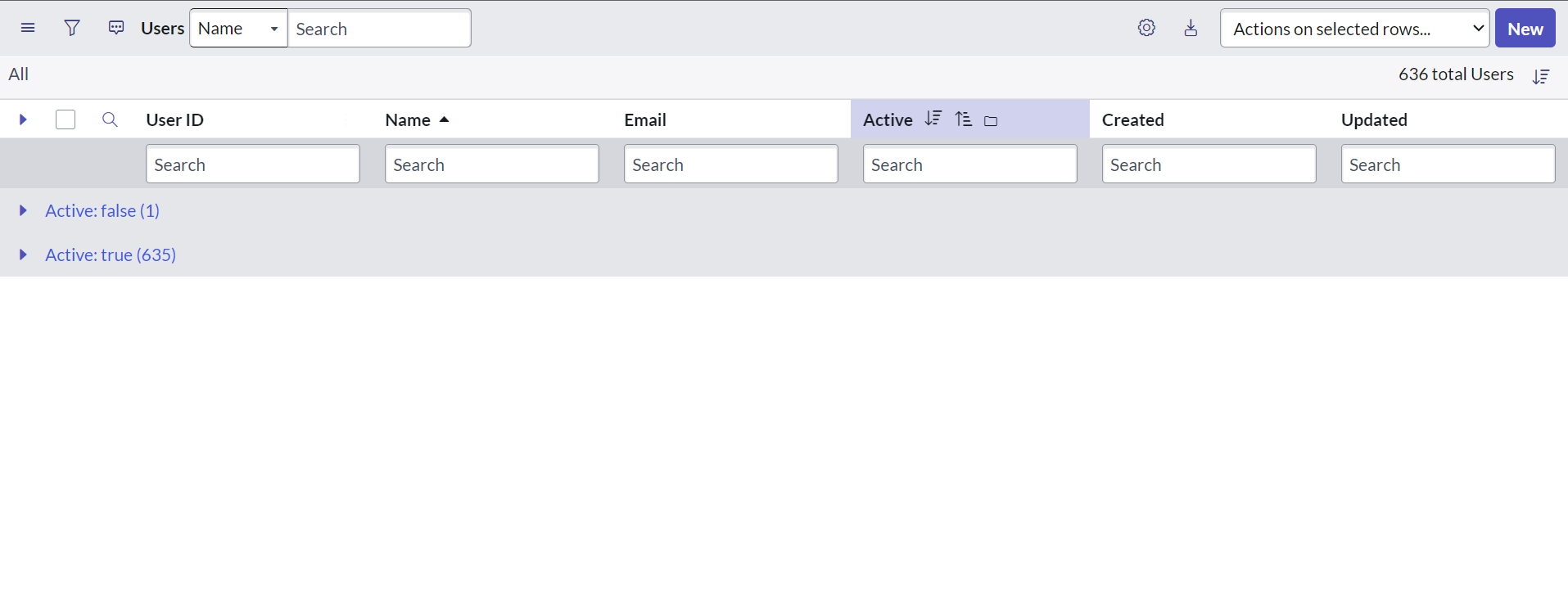
1. Navigate to Advanced section and add condition and modify the script.



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.

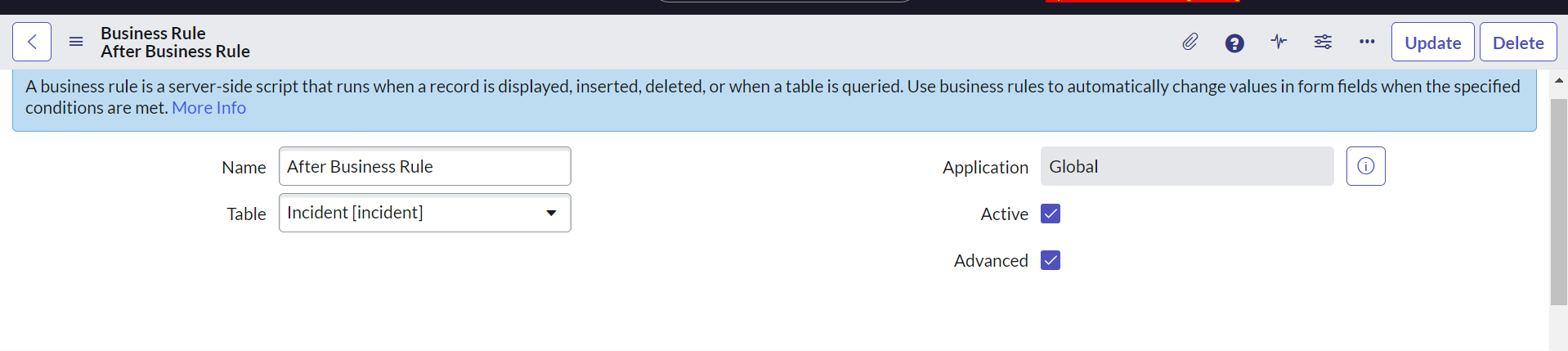


1. **What is After Business Rule?**

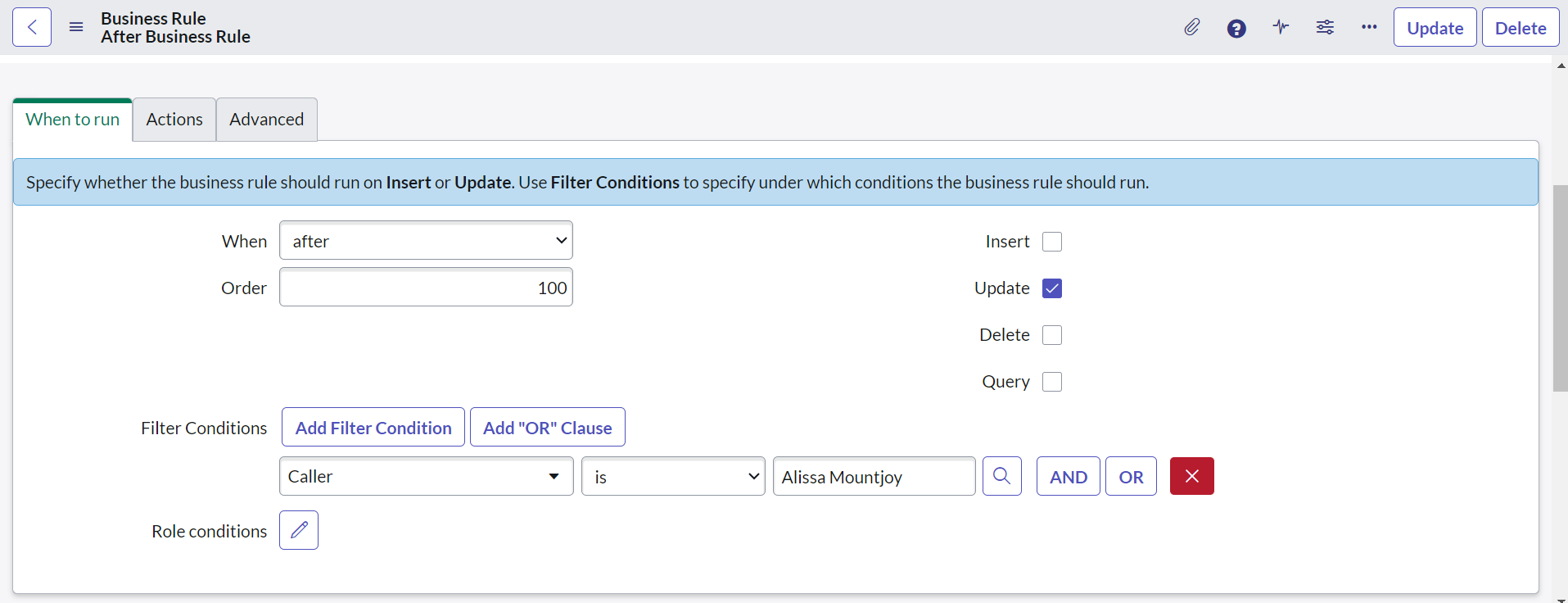
After Business Rule runs after the data is inserted to database. It is similar to Async BR. The only difference is Async BR runs in the background and gives the control back to the user, meanwhile the control will not be transferred back to user until the After BR has executed successfully.

To Create an After Business Rule, follow these steps:

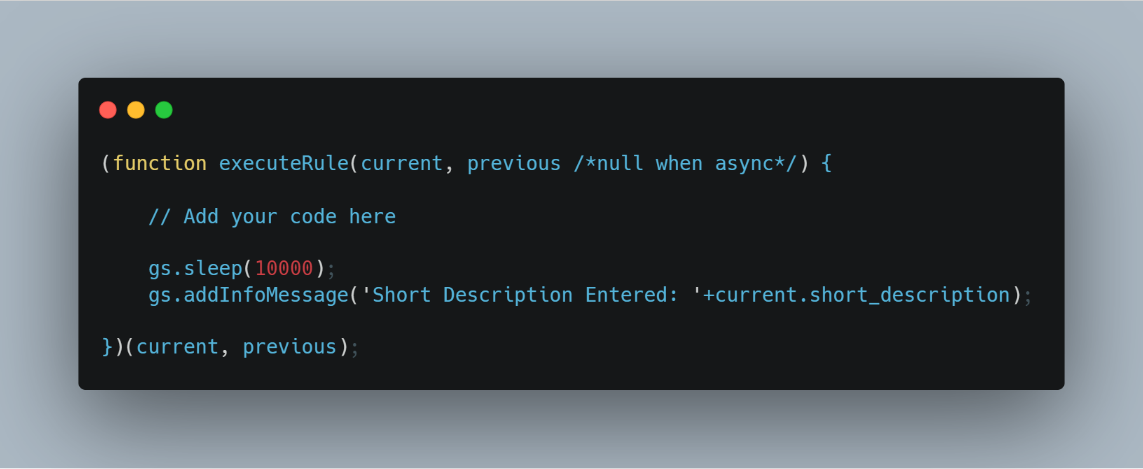
1. Create a new Business rule on Any Table and check the Advanced checkbox to bring the when section



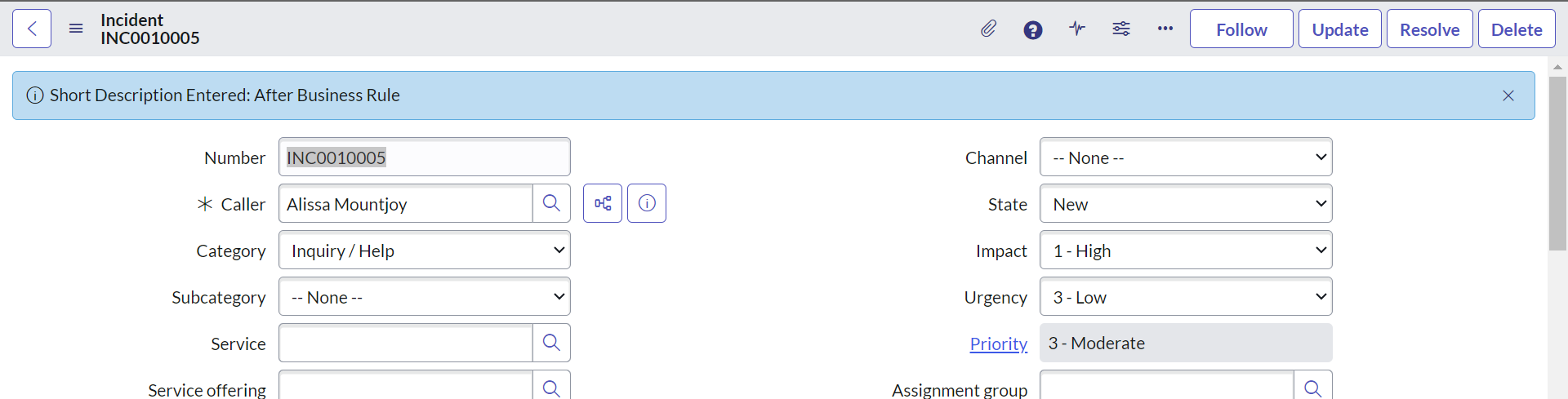
1. Select After from the When Dropdown and check Insert or Update or any checkbox to select the action timing. Also populate any filter condition if you want



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.

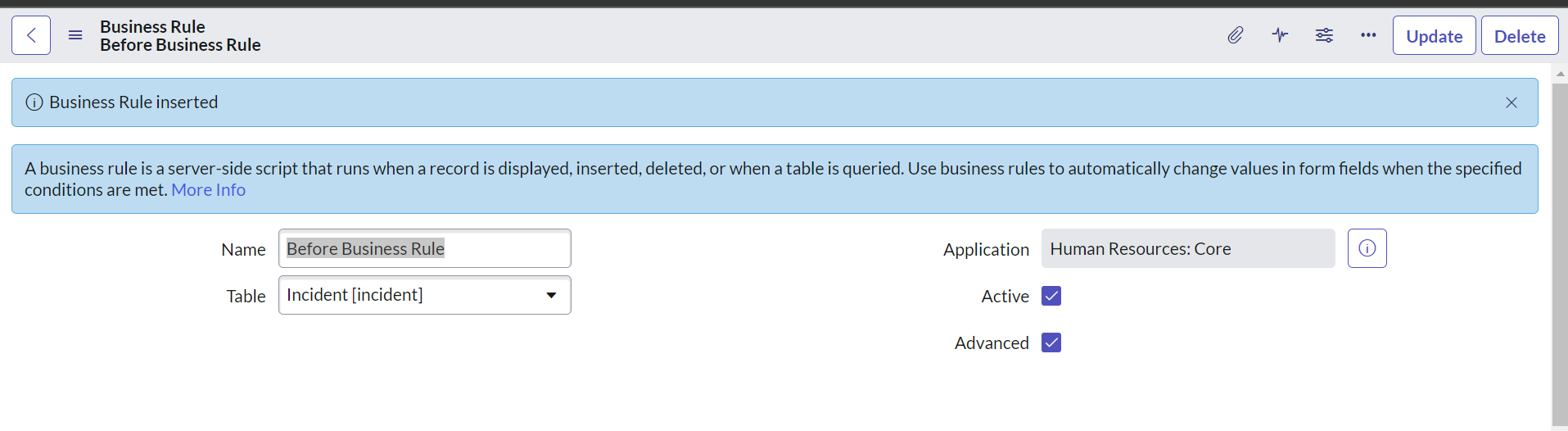


1. **What is Before Business Rule?**

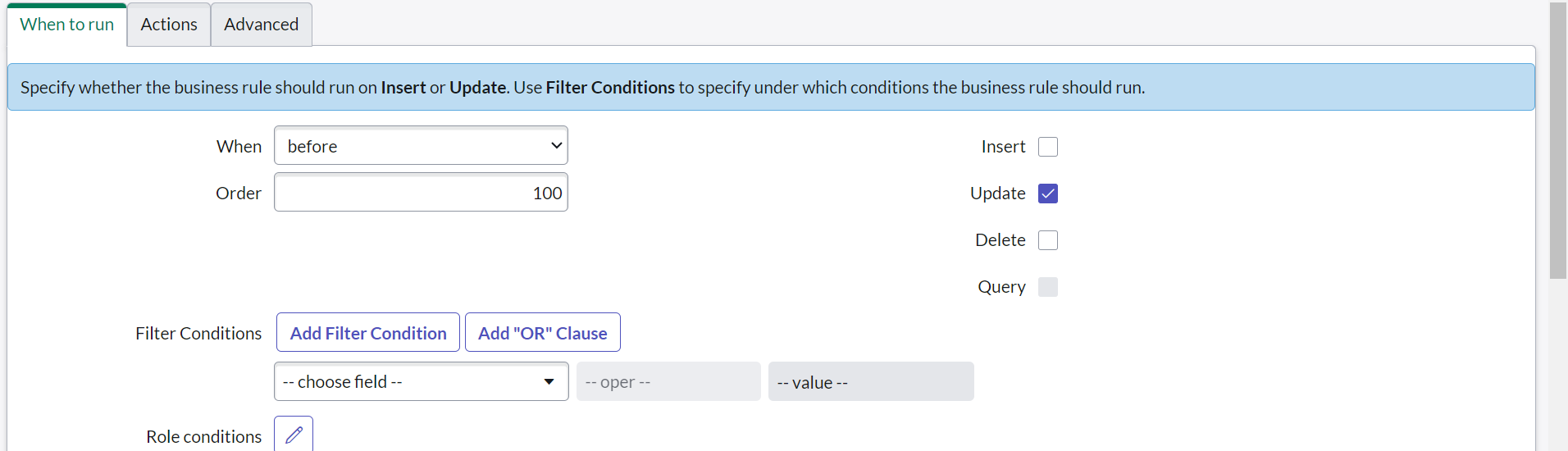
Before Business Rule runs before the data is inserted or updated into database. It can be used to create a validation on Server-Side end like validating if user has sufficient balance in the account before making transaction.

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



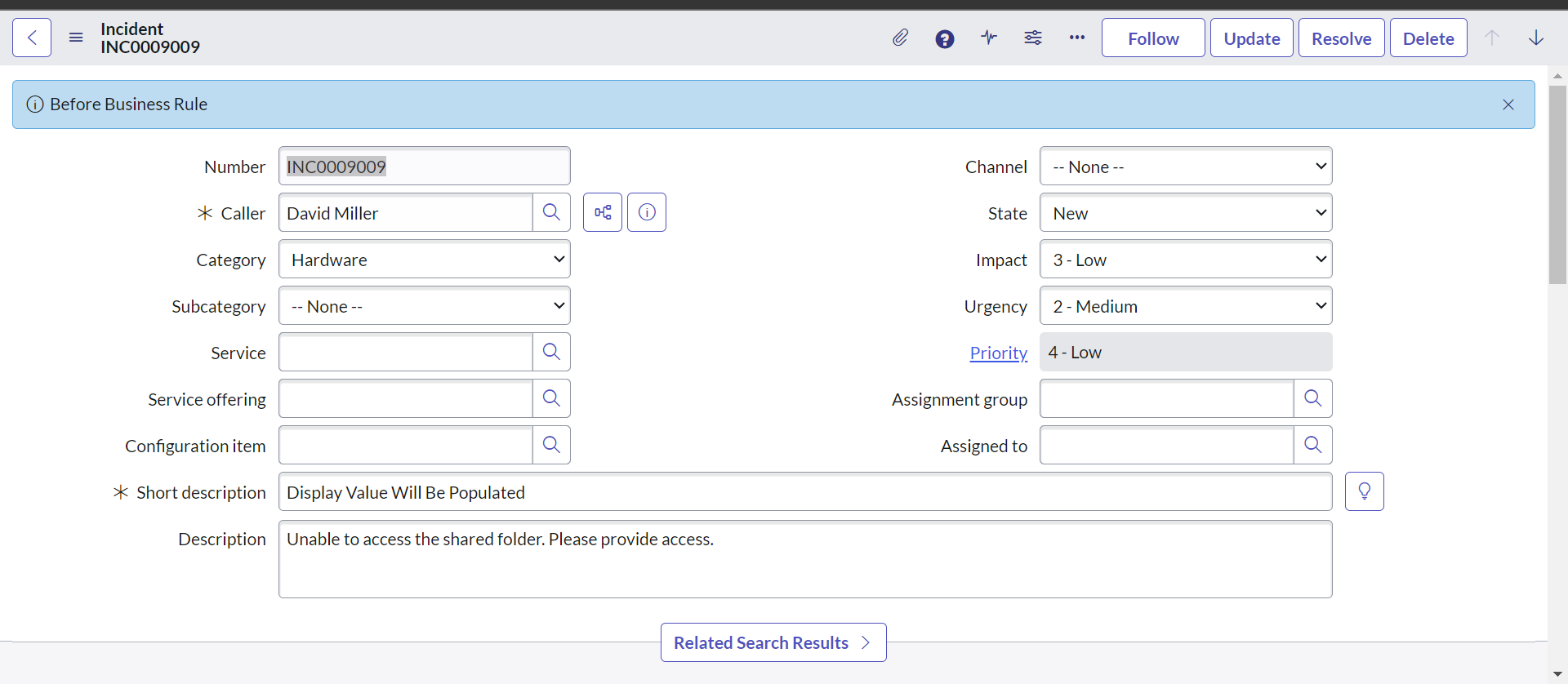
* 1. Select Before from The When Dropdown and check Insert or Update or any checkbox to select the action timing.



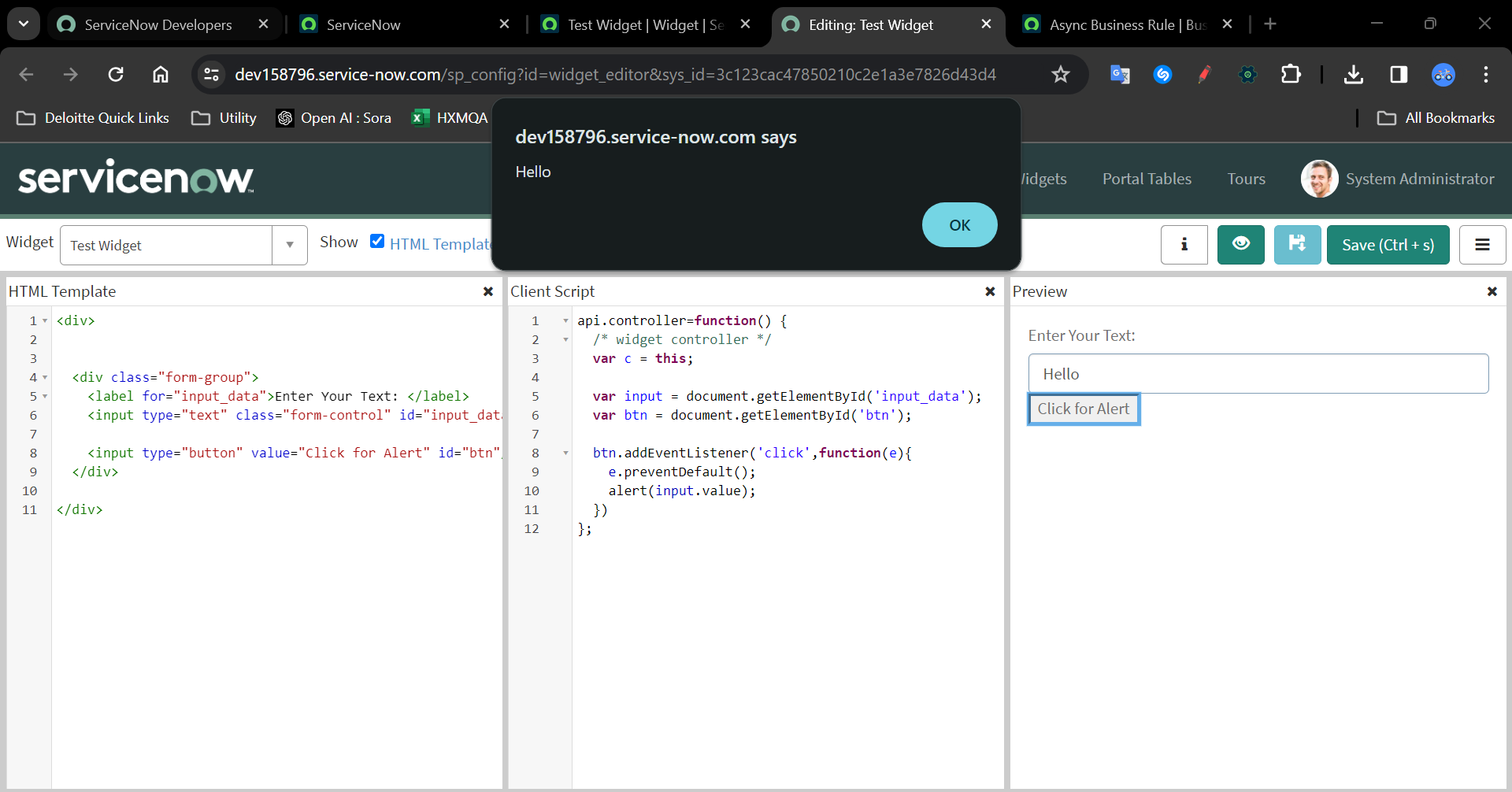
* 1. Add any script in the advanced section



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



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

****

**HTML:**

****

**Client Script:**

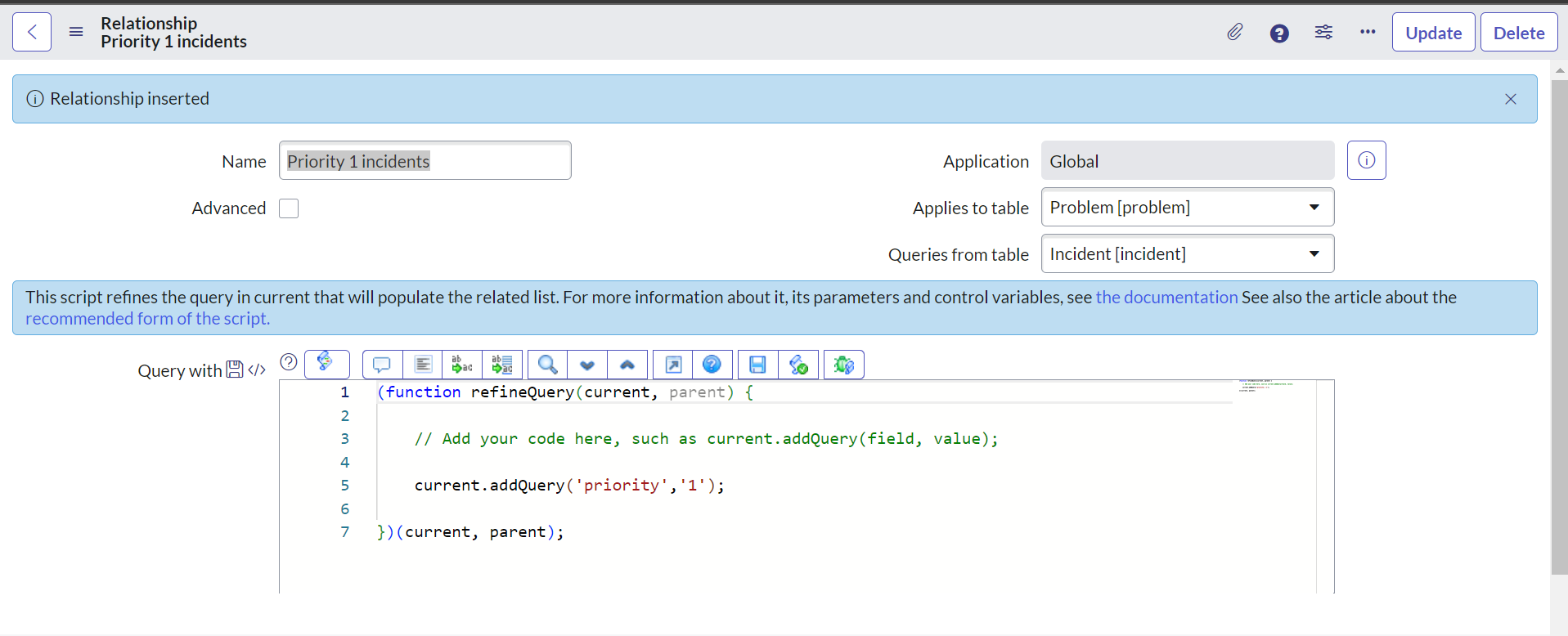
****

1. **How to create a related list and add it on a form?**

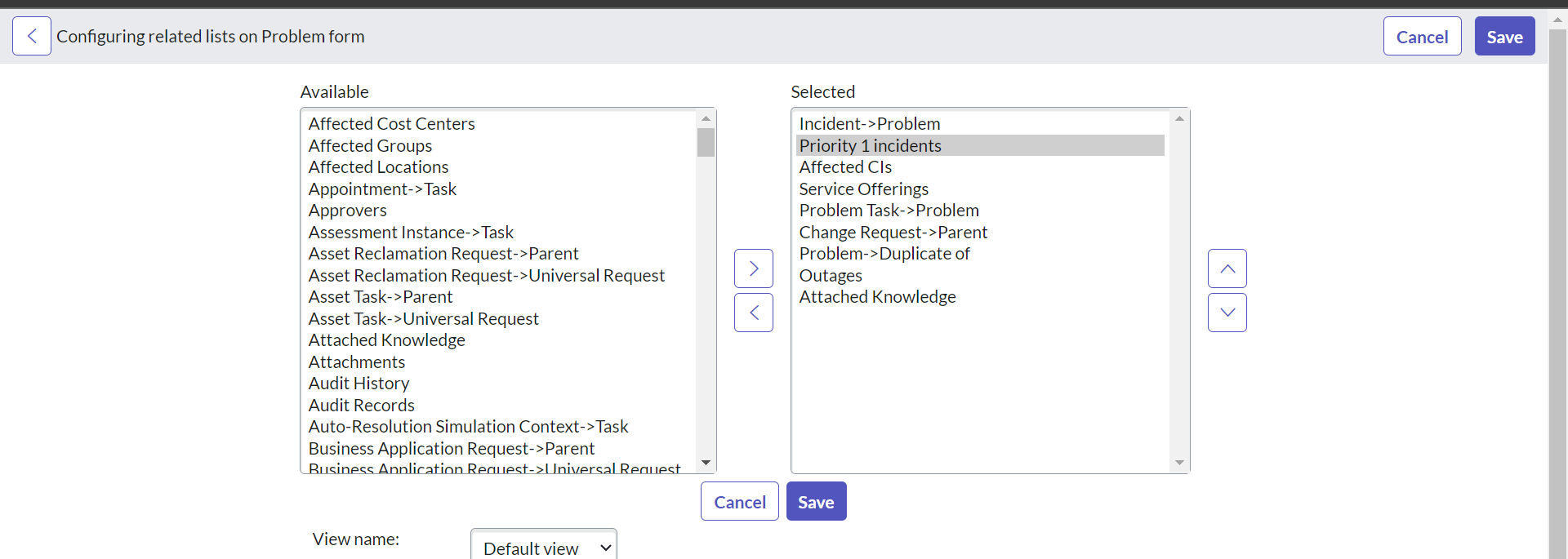
Before creating a related list, lets understand what is a related list.  
As the name suggest, related list is a list that holds data from a table that is related to the current table.  
  
Example: A case form can have Knowledge Articles or HR Task Related list, meanwhile an incident form can have problem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

Steps to create a related List:

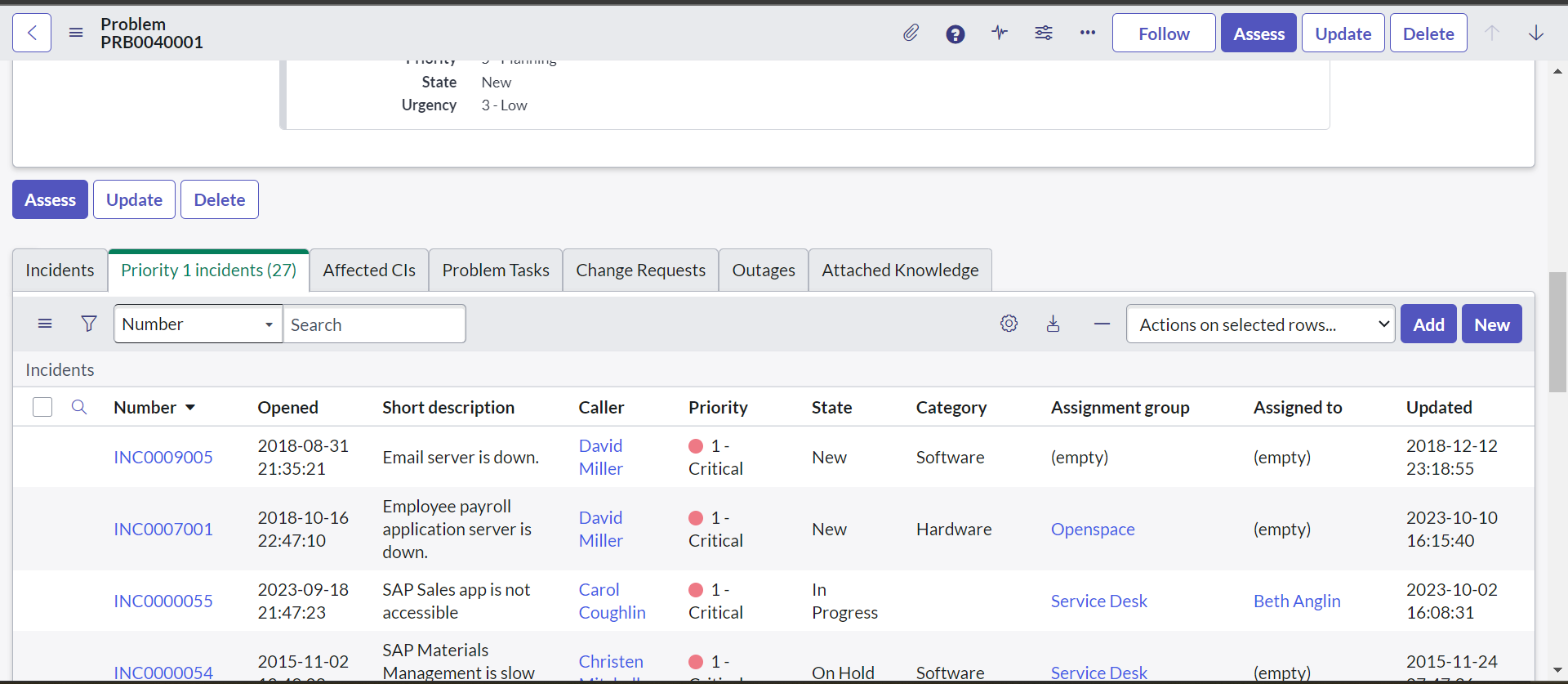
1. Navigate to System Definition -> Relationships and create a new record.
2. Fill in the fields like Applies to Table means in which table you want to show it
3. Queries from table means from which table you want the related data to be shown.
4. Query with section can handle any additional filter queries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



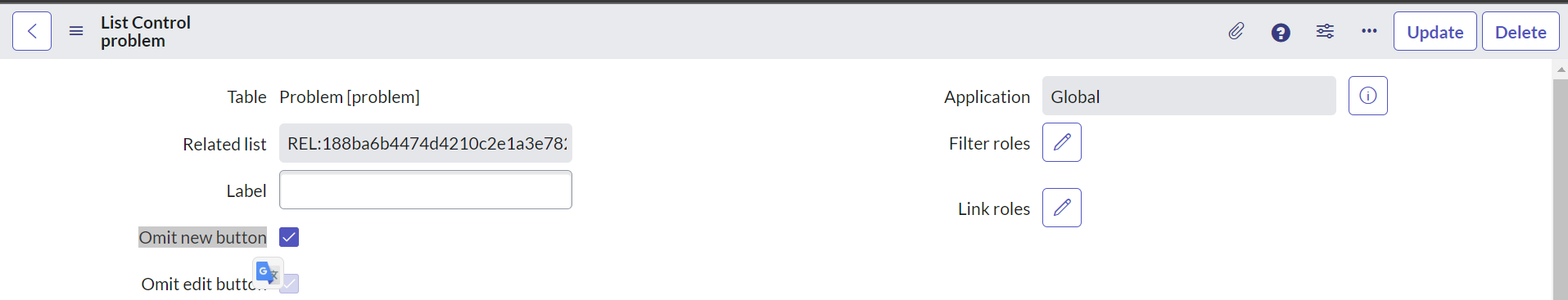
1. Check If the related list is visible.



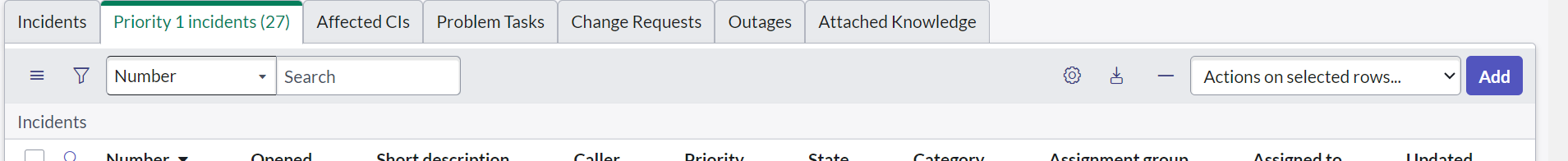
1. **How to hide UI Actions from a Related List?**

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?**

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 and UI Policies can be used to make a field mandatory and put up some validations.

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.

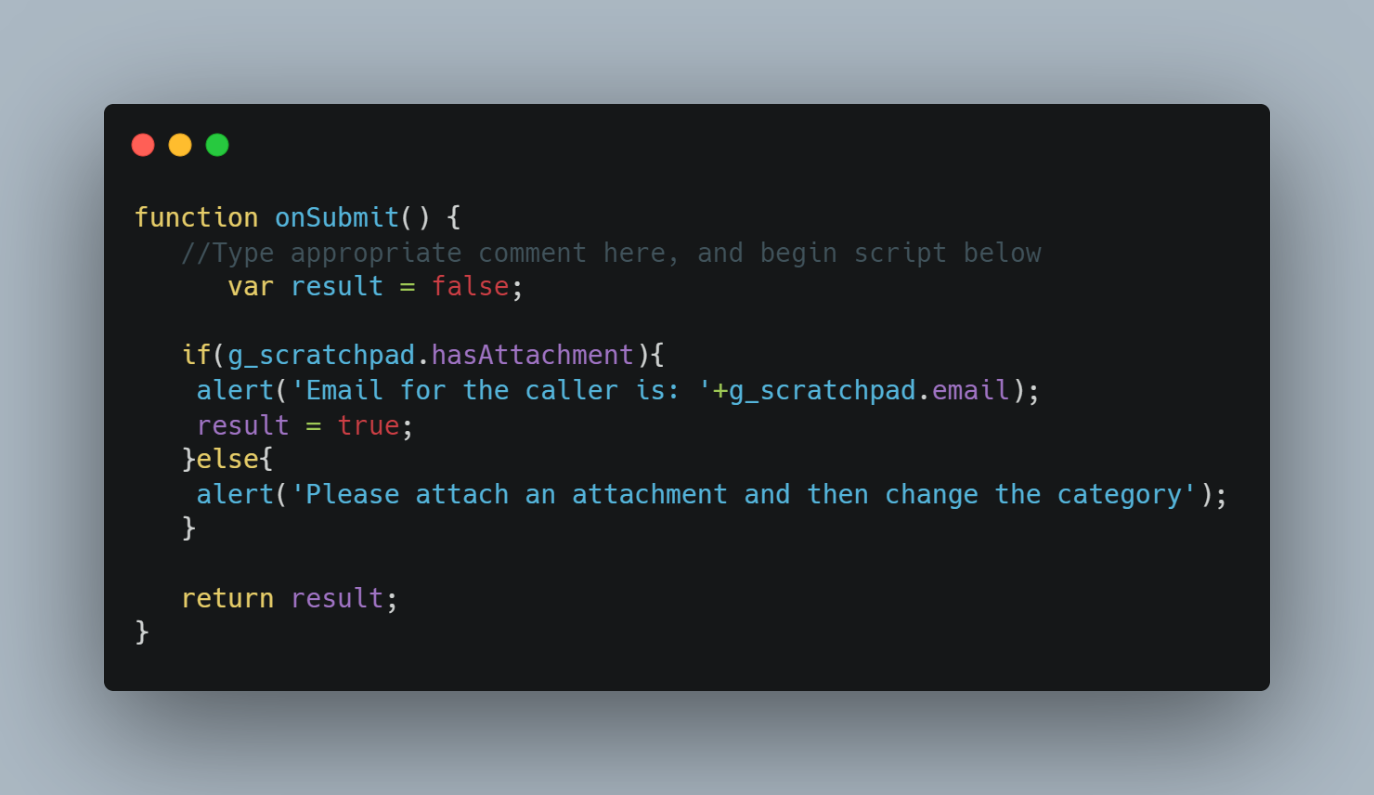
If we know that what information the client need from the server before the form loads, then in that case, once can use a display business rule with g\_scratchpad object to hold the data.

Steps for implementing it:

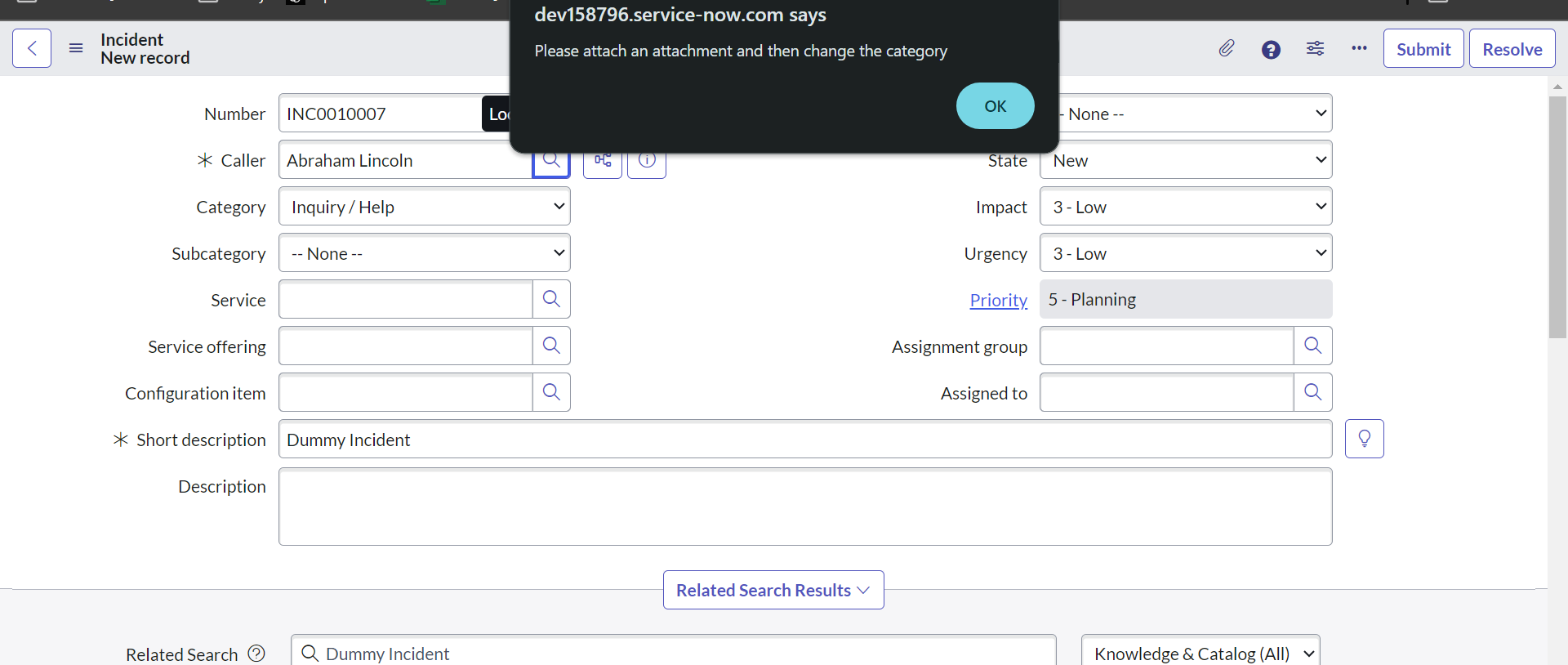
1. Create a display business rule and add the following script
2. Write the following script to check if incident has attachment.



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()**
2. **ACLs**
3. **SLAs**
4. **Data Policies**
5. **Transform Maps**
6. **Workflows**
7. **Why to use Scripted Rest APIs?**
8. **MID Server**
9. **Rest Messages**
10. **Order Guide and Catalog Items**
11. **Domain Separation?**
12. **Custom Applications**
13. **What is current.update() and why should we avoid it?**
14. **What is Class.Create() method used for in Script include?**
15. **What is initialize:function(){} Is used for in Script Include?**
16. **Difference between Display, Before and After Business Rule**

|  |  |  |  |
| --- | --- | --- | --- |
| **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. |