
The Complete ServiceNow System Administrator Course

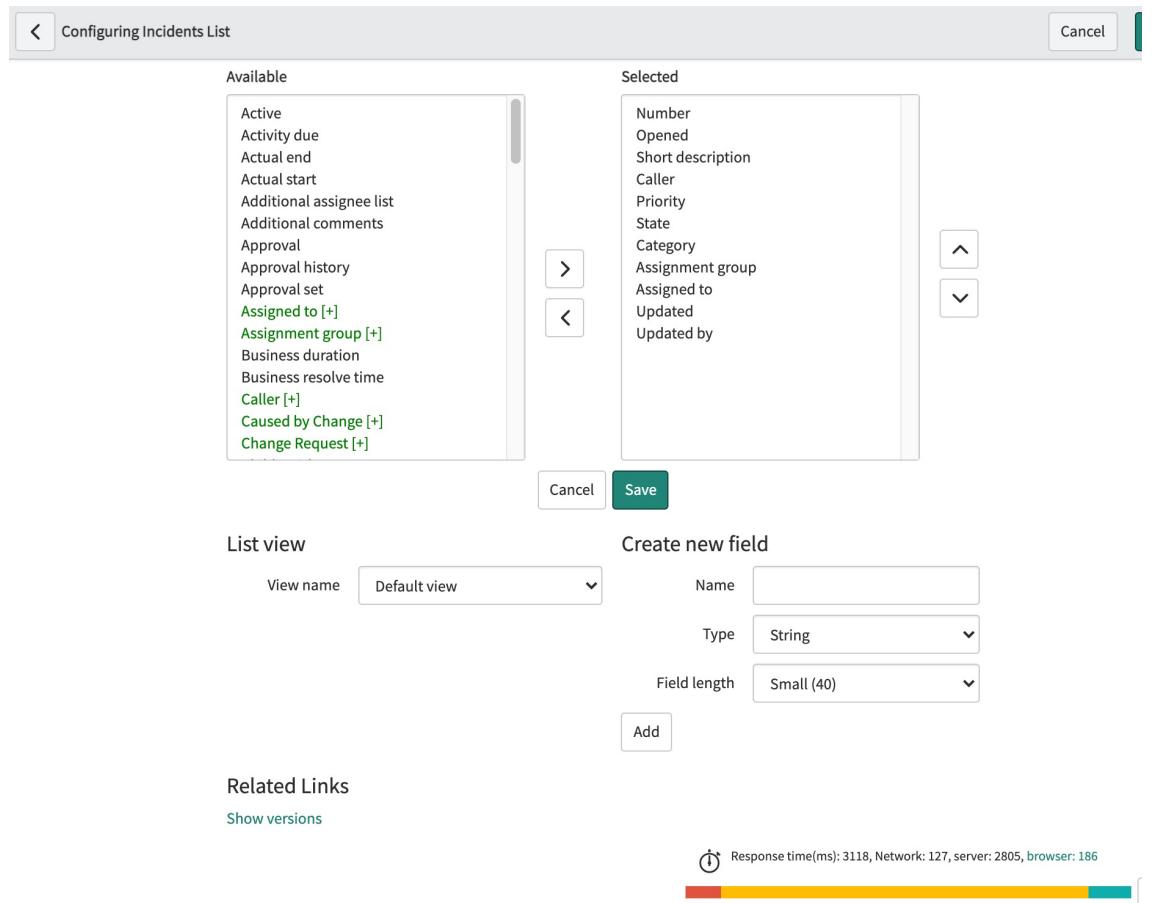
WORKBOOK ANSWERS

Section – 3 – Exercises



1. Configure the Incident list view layout to include the following fields:
Number, Caller, Category, Priority

1. Navigate to Incidents -> Open or Incidents -> All
2. Right click on any of the column headers and go to Configure -> List Layout
3. Add Number, Caller, Category, and Priority to the Selected list of fields
4. Re-order fields as required on the Selected list and click save.



2. Create a filter for all incidents that were opened today and are unassigned

1. Navigate to Incidents -> All
2. Set the filter as Opened on Today and Assignment Group is empty
3. Click Run to apply the filter

The screenshot shows a software interface for managing incidents. At the top, there's a navigation bar with 'Incidents' and 'New' buttons, followed by search fields for 'Number' and 'Search'. Below the navigation bar, there's a large green 'All' button with a funnel icon. Underneath the green button, there are several action buttons: 'Run', 'Save...', 'AND', 'OR', 'Add Sort', and a refresh icon. To the right of these buttons is a 'Save as:' field containing 'Today's unassigned incidents', a 'Visible to:' section with radio buttons for 'Me' (selected), 'Everyone', and 'Group', and a 'Save' button. A horizontal line separates this from the filter conditions. Below the line, the text 'All of these conditions must be met' is displayed. There are two main condition rows. The first row contains three dropdowns: 'Opened' (set to 'on' 'Today'), 'AND', and 'OR'. The second row contains 'Assignment group' (set to 'is empty') and 'AND'. To the right of the second row are 'OR' and a delete ('X') button. A long blue horizontal bar is at the bottom of the screen.

3. Save the filter as Today's unassigned incidents

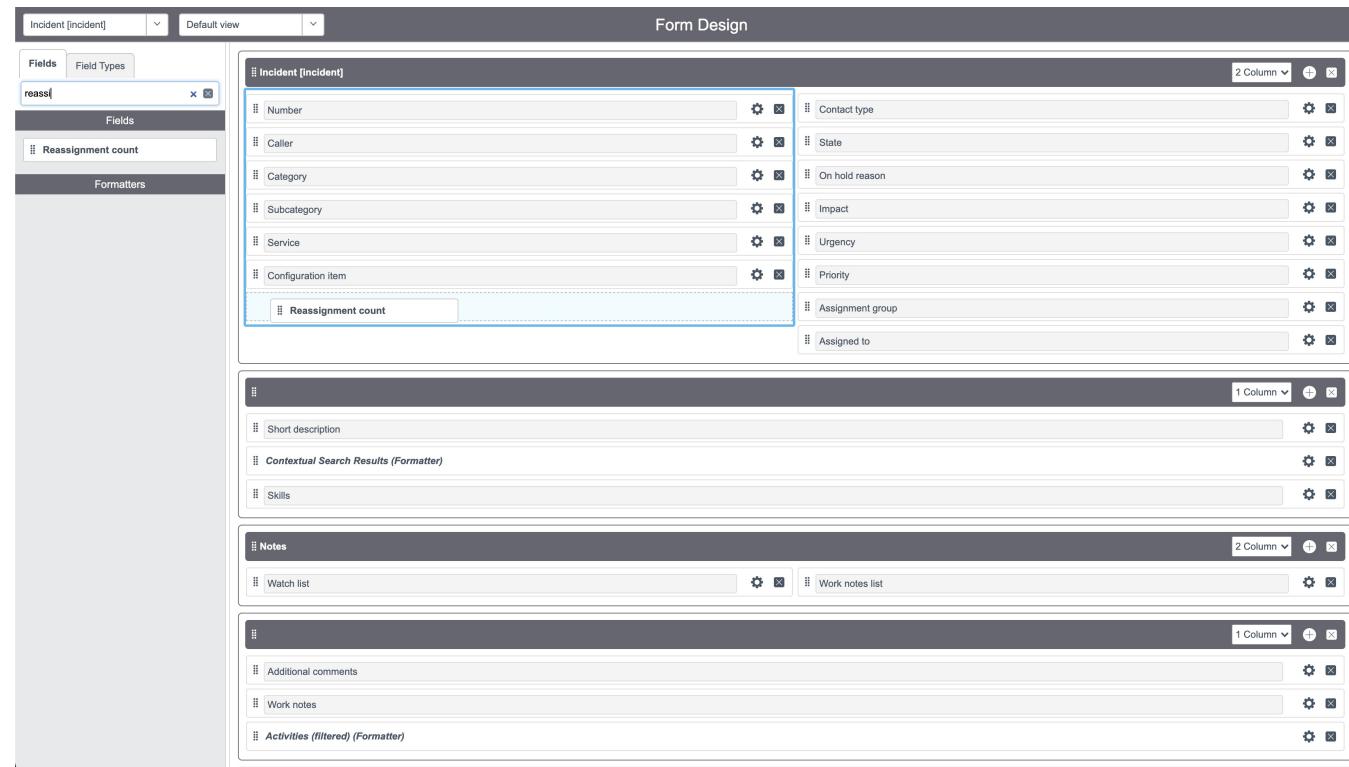
1. Navigate to Incidents -> All
2. Set the filter as Opened on Today and Assignment Group is empty
3. Click Save button and give the filter a name and save the filter. You can choose to make the filter visible for everyone or a group or just you
4. To re-use a saved filter, click on the triple-bar or the hamburger icon and Filters and then select your saved filter

The screenshot shows the Jira Incidents interface. At the top, there are buttons for 'Incidents' (highlighted), 'New', 'Search', and a search bar. Below the search bar is a filter icon labeled 'All'. A toolbar contains 'Run', 'Save...', 'AND', 'OR', 'Add Sort', and a refresh icon. In the bottom right corner of the toolbar, there is a 'Save' button. Below the toolbar, a 'Save as:' field contains the text 'Today's unassigned incidents'. Next to it is a 'Visible to:' dropdown with options: 'Me' (radio button), 'Everyone' (radio button, selected), 'Group', and a 'Save' button. A note below says 'All of these conditions must be met'. Underneath, there are two condition rows: 'Opened on Today' and 'Assignment group is empty'. Each row has dropdowns for 'Opened' (set to 'on Today'), 'Assignment group' (set to 'is empty'), and logic buttons 'AND', 'OR', and 'X'.

The screenshot shows the Jira navigation menu. The 'Incidents' and 'New' buttons are at the top. Below them is a 'View' dropdown with 'Filters' selected. A dropdown menu is open under 'Filters', listing various filter categories: 'Edit personal filters', '-- None --', 'Active', 'Refresh List', 'Create Favorite', 'Active - Unassigned', 'Assigned to me', 'Closed', 'Incident Mobile', 'Incident Portal', 'My Open Incidents', and 'Today's unassigned incidents' (which is highlighted in green).

4. Add the reassignment count field to the default incident form view

1. Navigate to Incidents and open any record
2. Click on the triple-bar or the hamburger icon and click Configure – Form Design. You can also do this using the Form Layout
3. Make sure the view is "Default View" at the top
4. Search for the Reassignment Count and add it below Configuration Item field
5. Click Save and the field should appear on the form



Section – 4 – Exercises



1. Create a business rule which sets a problem's assignment group to the Administrators group, if a group is not selected on insert

1. You can create a new business rule from the Business Rule table or from the Problem table record
2. Make sure the business rule runs on the 'Problem' table and give a name to the business rule
3. Select the 'Insert' checkbox
4. Set the filter condition to "Assignment group is empty"
5. On the actions section, set field value "Assignment Group To Administrators"
6. If the desired group does not exist, create the group.

The screenshot shows the 'Business Rule' configuration interface. At the top, there are fields for 'Name' (Assign to Admins if unassigned), 'Table' (Problem [problem]), 'Application' (Global), 'Active' (checked), and 'Advanced' (unchecked). Below this, the 'When to run' tab is selected, showing options for 'Insert' (checked) and 'Update' (unchecked). Under 'Filter Conditions', there is a dropdown for 'Assignment group' and a dropdown for 'is empty'. Below this, the 'Actions' tab is selected, showing instructions for setting field values. It includes a 'Set field values' section where 'Assignment group' is set to 'To' 'Administrators', and other options like '-- choose field --' and 'Add message' are available. There is also an 'Abort action' checkbox.

2. Create a UI policy which will hide the priority field if impact and urgency are set to 3 - Low

1. You can create a new UI Policy from the UI Policies table or from the Problem table record
2. Make sure the UI Policy runs on the 'Problem' table and give the UI policy a name
3. Set the filter condition to "Impact is 3-Low AND Urgency is 3-Low" and save the UI policy
4. Now, add UI policy action on the UI policy to hide the Priority field (Set Visible to False)
5. Save the record

Table: Problem [problem] Application: Global Active:

* Short description: Hide the priority field if impact and urgency are set to 3 - Low Order: 100

When to Apply: Script

Conditions: Add Filter Condition | Add "OR" Clause
All of these conditions must be met

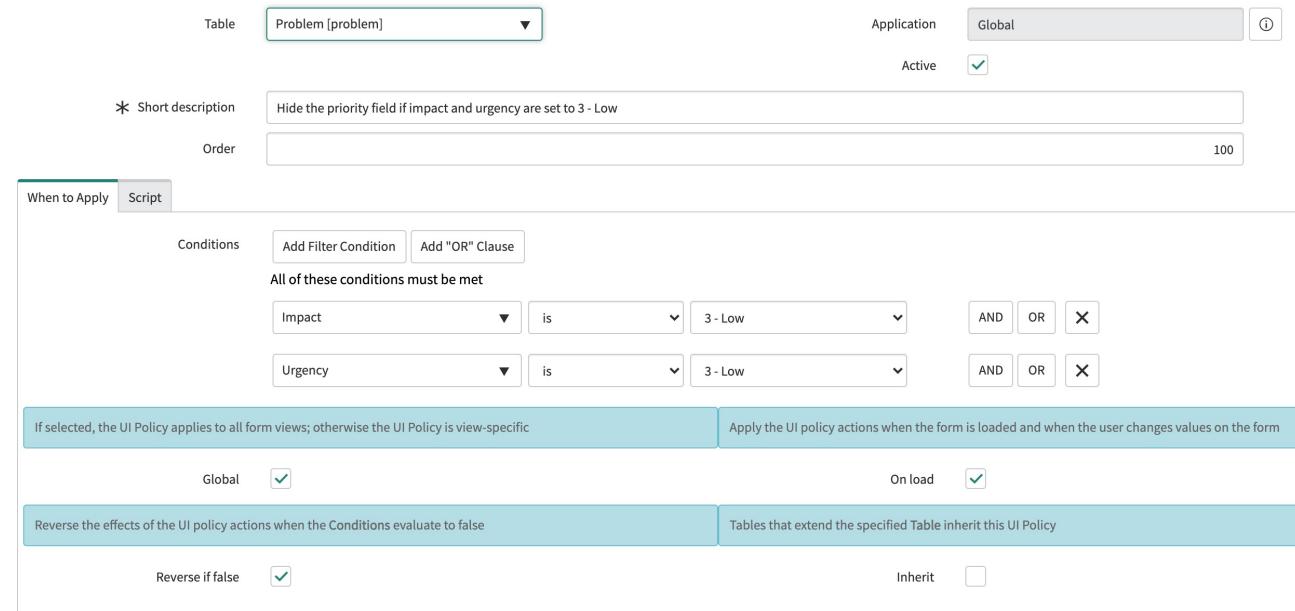
Impact: is 3 - Low AND
Urgency: is 3 - Low AND

If selected, the UI Policy applies to all form views; otherwise the UI Policy is view-specific | Apply the UI policy actions when the form is loaded and when the user changes values on the form

Global: On load:

Reverse the effects of the UI policy actions when the Conditions evaluate to false | Tables that extend the specified Table inherit this UI Policy

Reverse if false: Inherit:



UI policy: Hide the priority field if impact and urgency are set to 3 - Low Application: Global

Table: Problem [problem] Mandatory: Leave alone

* Field name: Priority Visible: False

Read only: Leave alone Clear the field value:



3. Activate the Save UI action only for the Incident form view

1. There are out-of-the-box (OOB) 'Save' UI actions that on the Global table (i.e, it runs on every table)
2. Specifically look for two 'Save' UI actions that have Form Button set to true
3. Open each of these UI actions and change the table to Incident and do an 'Insert' or "Insert and Stay"
4. Come back to the Incident table and create a record or open a record and you should see the Save button

<u>Save</u>	Global [global]	Saves a new record and redirects back to self (button version, advanced mode)
<u>Save</u>	Global [global]	Saves an existing record and redirects back to current screen (button version, advanced mode)

Section – 4 – Advanced Exercises



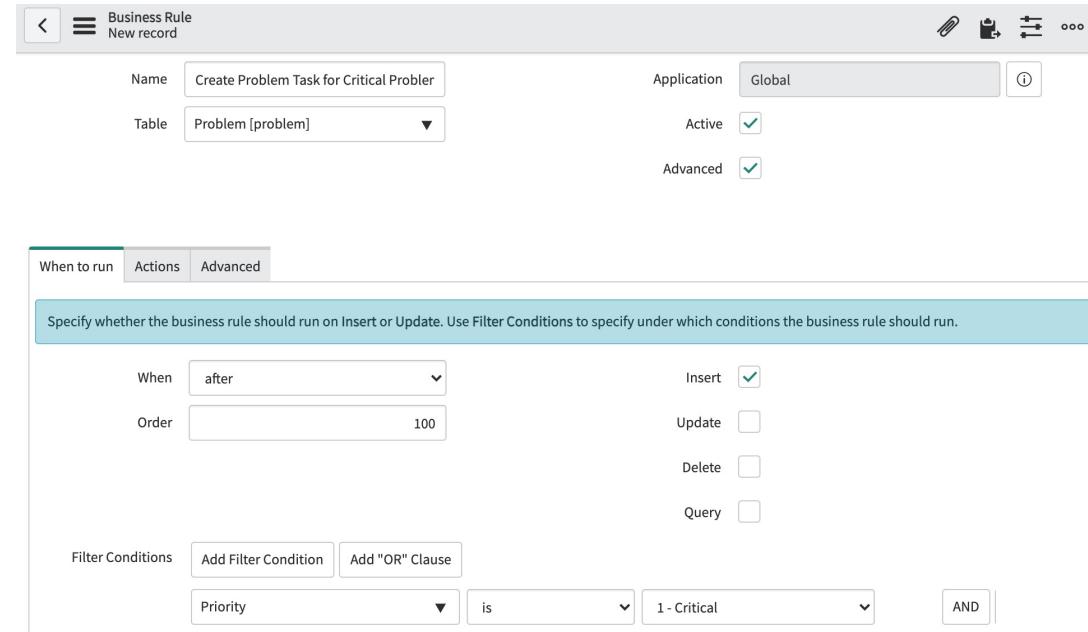
1. Create a problem task record if a problem record is saved with a priority of 1 - Critical, and relate it to the problem record

1. Create a business rule on Problem table to run on Insert when Priority is 1-Critical
2. Add below script on the advanced tab:

```
(function executeRule(current, previous /*null  
when async*/){
```

```
    var gr = new GlideRecord("problem_task");  
    gr.newRecord();  
    gr.problem = current.sys_id.toString();  
    gr.short_description = "Critical Problem  
Task";  
    gr.insert();
```

```
})(current, previous);
```

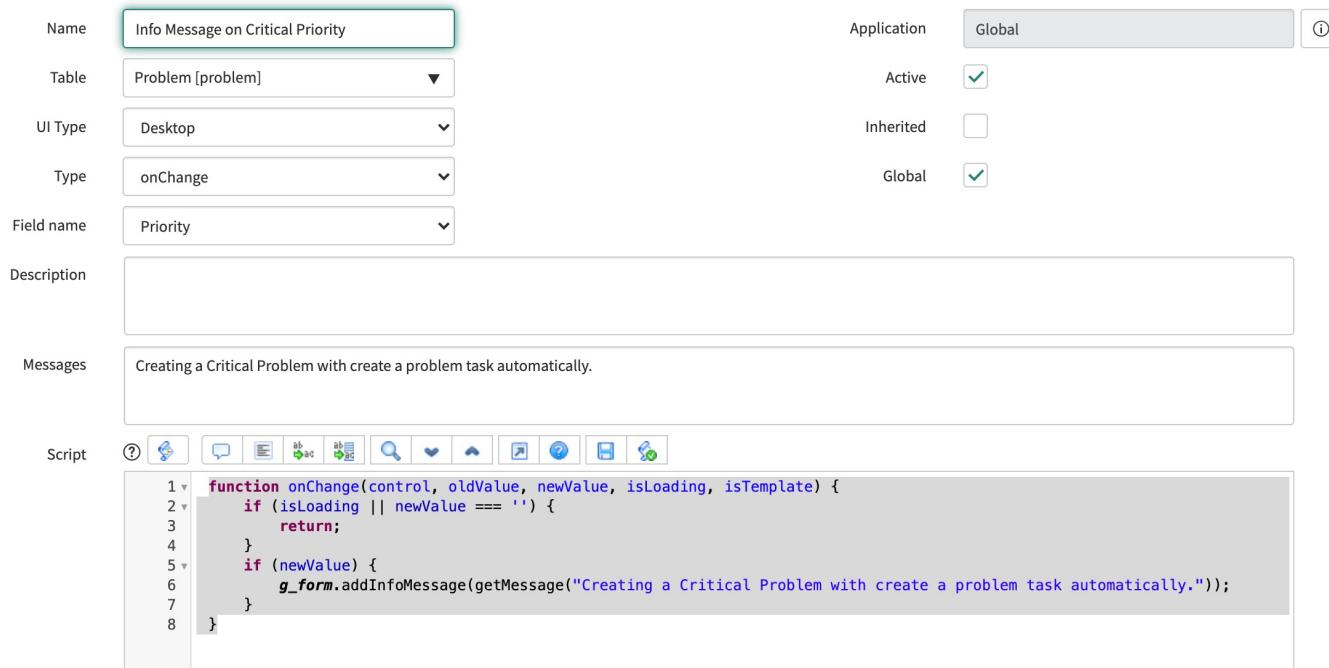


2. Create a client script which informs the user that a problem with a priority of 1 - Critical will create a problem task

1. Create a client script on Problem table with onChange type and field name as Priority
2. Add below script:

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

```
        g_form.addInfoMessage(getMessage("Creating a
Critical Problem with create a problem task
automatically."));
    }
}
```



Section – 5 – Exercises



1. Create a car table with the following fields:
make (choice) model (string) release_date (date) cost (currency)

1. Create a new table and name is 'Car'
2. Optionally create a module
3. Add the required fields
4. Remember, when you save the table,
it automatically creates the core
fields such as Sys ID, Created,
Created By, Updated, and Updated By
and some additional core fields.

The screenshot shows a table configuration interface with the following details:

Table Configuration:

- * Label: Car
- * Name: u_car
- Extends table: (empty)
- Application: Global
- Create module: (checkbox)

Table Columns:

Action	Column label	Type	Reference	Max length	Default value	Display
X	Make	Choice				false
X	Model	String				false
X	Release Date	Date				false
X	Cost	Currency				false
+	Insert a new row...					

Buttons:

- Submit
- Cancel

2. Create 5 choices for the make field

1. Open the 'Car' table and open the 'Make' field/dictionary
 2. Add choices on the Choices related list at the bottom of the form
 3. Optionally, set Choice List Specification to include --None-- as one of the choices

Car [u_car]

Type: Choice

Column label: Make

Column name: u_make

Application: Global

Active:

Function field:

Read only:

Mandatory:

Display:

Choice List Specification | **Default Value**

Displays a list of suggested values in a Choice list. In the Advanced view you can select the Choice table and the Choice field to take choice values from, plus a Dependent field.

Choice: Dropdown with -- None --

Create Choice List | **Delete Column** | **Update**

Related Links

Show Table
Advanced view

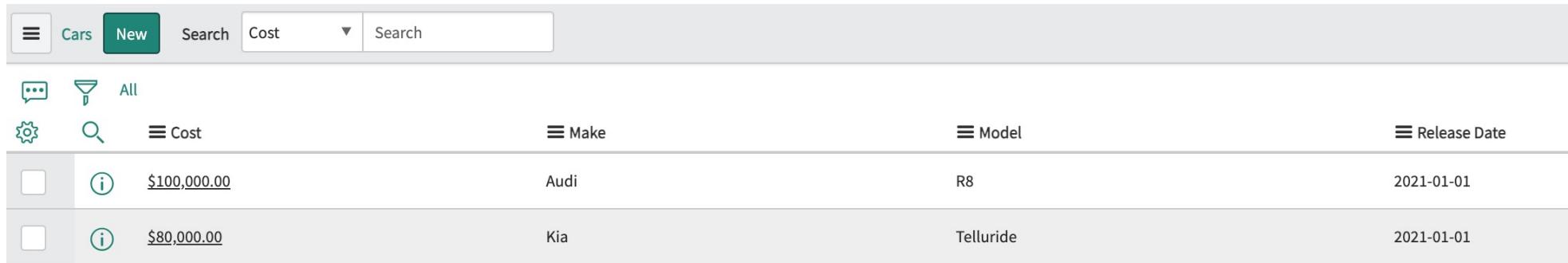
Access Controls | **Choices** | **Attributes** | **Labels (1)**

	Choices	New	Search	Label	Search	Search
	Choices					
		New		Search	Label	

	Label	Value	Language	Sequence	Inactive	Updated
<input type="checkbox"/>	Audi	audi	en		false	2021-06-01 07:38:12
<input type="checkbox"/>	BMW	bmw	en		false	2021-06-01 07:38:23
<input type="checkbox"/>	Honda	honda	en		false	2021-06-01 07:38:32
<input type="checkbox"/>	Lexus	lexus	en		false	2021-06-01 07:38:47
<input type="checkbox"/>	Kia	kia	en		false	2021-06-01 07:38:59

3. Create 2 car records

1. Open the 'Car' table
2. Create two new records with your own values



The screenshot shows a table interface for managing car records. At the top, there's a navigation bar with 'Cars' (highlighted in green), 'New', 'Search' (with dropdown), and a search input field. Below the navigation is a toolbar with icons for 'All' (filter), 'Cost' (sort), 'Make' (sort), 'Model' (sort), and 'Release Date' (sort). The main area displays two rows of data:

	Cost	Make	Model	Release Date
<input type="checkbox"/>	(i) \$100,000.00	Audi	R8	2021-01-01
<input type="checkbox"/>	(i) \$80,000.00	Kia	Telluride	2021-01-01

4. Configure the form to your liking and add the created field to the form

1. Open the 'Car' table
2. Open any record and configure the fields on the form using Form Designer or Form Layout
3. Make sure to add the Created field to the form

The screenshot shows a 'Car' record form with the following details:

- Make:** Audi (dropdown menu)
- Model:** R8 (text input)
- Release Date:** 2021-01-01 (date picker)
- Cost:** \$ 100,000.00 (dropdown menu)
- Created:** 2021-06-01 07:48:09 (text input)

At the top right, there are buttons for **Update** and **Delete**. At the bottom left, there are buttons for **Update** and **Delete**.

Section – 6 – Exercises



4. Configure the form to your liking and add the created field to the form

1. Open the 'Car' table
2. Open any record and configure the fields on the form using Form Designer or Form Layout
3. Make sure to add the Created field to the form

The screenshot shows a 'Car' record form with the following details:

- Make:** Audi (dropdown menu)
- Model:** R8 (text input)
- Release Date:** 2021-01-01 (date picker)
- Cost:** \$ 100,000.00 (dropdown menu)
- Created:** 2021-06-01 07:48:09 (text input)

At the top right, there are buttons for **Update** and **Delete**. At the bottom left, there are buttons for **Update** and **Delete**.

Continued...

1. Go to System Security -> Access Control (ACL)
2. Create two ACLs each, one for 'create' and 'write' operation on the Incident table for 'caller' and 'contact type' fields to allow users with 'incident_team' role (Remember, you must elevate yourself as 'security_admin' to be able to create, update or delete ACLs)

The screenshot shows the 'Access Control' configuration page. The main form includes fields for Type (record), Operation (create), Application (Global), Active status (checked), Admin overrides (checked), Protection policy (None), Name (Incident [incident]), Caller, and Description. Below the main form is a 'Definition' section with a detailed description of access control rules and a 'Requires role' table.

Access Control Rules allow access to the specified resource if *all three* of these checks evaluate to true:

1. The user has one of the roles specified in the Role list, or the list is empty.
2. Conditions in the Condition field evaluate to true, or conditions are empty.
3. The script in the Script field (advanced) evaluates to true, or sets the variable "answer" to true, or is empty.

The three checks are evaluated independently in the order displayed above.

[More Info](#)

Requires role	
	Role
X	incident_team
+	Insert a new row...

2. Create a new access control rule which gives permissions to change the last name of any user record

1. Go to System Security -> Access Control (ACL)
2. Remember, you must elevate yourself as 'security_admin' to be able to create, update or delete ACLs
3. Create an ACL on User (sys_user) table and the "Last Name" column for 'write' operation and add any role that should get access to edit the users' last names
4. 'incident_team' role is added to the ACL as an example

The screenshot shows the configuration of an Access Control Rule (ACR) for the 'User [sys_user]' table. The 'Last name' column is selected for the 'Name' field. The 'Type' is set to 'record' and the 'Operation' is 'write'. The 'Admin overrides' checkbox is checked. The 'Protection policy' is set to '-- None --'. The 'Active' checkbox is checked. The 'Advanced' checkbox is unchecked. The 'Description' field is empty.

Definition

Access Control Rules allow access to the specified resource if *all three* of these checks evaluate to true:

1. The user has one of the roles specified in the Role list, or the list is empty.
2. Conditions in the Condition field evaluate to true, or conditions are empty.
3. The script in the Script field (advanced) evaluates to true, or sets the variable "answer" to true, or is empty.

The three checks are evaluated independently in the order displayed above.

[More Info](#)

Requires role

Role
incident_team

Section - 7 - Exercises

1. Create an incident record, incident A

1. Go to Incident -> Create New
2. Create a new incident and save the incident after filling in all mandatory fields
3. Note down the incident number

Incident
Preview this record

Number: INC0010001

Caller: Abel Tuter

Category: Inquiry / Help

Subcategory: -- None --

Service:

Service offering:

Configuration item:

Contact type: -- None --

State: New

Impact: 3 - Low

Urgency: 3 - Low

Priority: 5 - Planning

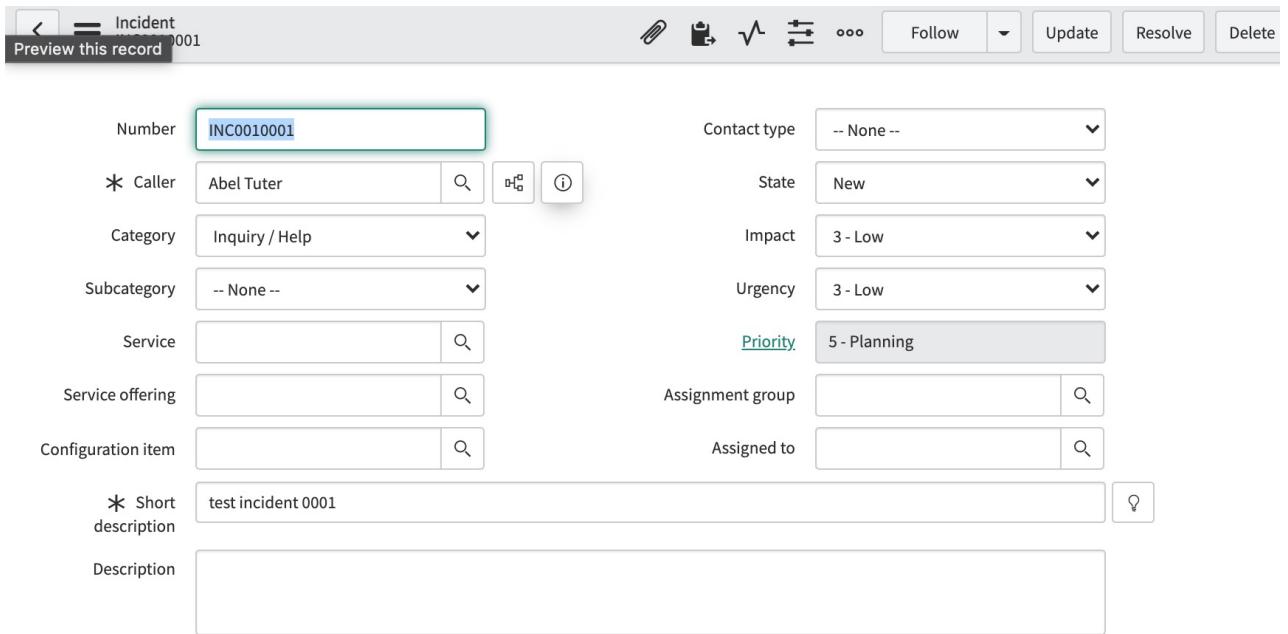
Assignment group:

Assigned to:

Short description: test incident 0001

Description:

Follow Update Resolve Delete



2. Create another incident record, incident B, and link incident A to incident B

1. Go to Incident -> Create New
2. Create a new incident and save the incident after filling in all mandatory fields
3. Note down the incident number
4. Now, add the incident number from the previous slide as the parent incident in the "Related Records" section

The screenshot shows the creation of a new incident record (INC0010002) in a software application. The main form displays various incident details:

- Number:** INC0010002
- Caller:** Abel Tuter
- Category:** Inquiry / Help
- Subcategory:** -- None --
- Service:** (empty)
- Service offering:** (empty)
- Configuration item:** (empty)
- Contact type:** -- None --
- State:** New
- Impact:** 3 - Low
- Urgency:** 3 - Low
- Priority:** 5 - Planning
- Assignment group:** (empty)
- Assigned to:** (empty)
- Short description:** test incident 0002
- Description:** (empty)

At the bottom, the **Related Records** tab is active, showing:

- Parent Incident:** INC0010001
- Change Request:** (empty)
- Problem:** (empty)
- Caused by Change:** (empty)

3. Create a new emergency change request

1. Go to Change -> Create New
2. Click on All and select 'Emergency'
3. Save the record and note the change number

Create a change request

Pinned Models Preapproved All

Emergency ITIL Mode 1 Emergency Change State model

Normal ITIL Mode 1 Normal Change State model

New Assess Authorize Scheduled Implement Review Closed Canceled

Number	CHG0030001	Model	Emergency
Requested by	System Administrator	Type	Emergency
Category	Other	State	New
Service		Conflict status	Not Run
Service offering		Conflict last run	
Configuration item		Assignment group	
Priority	4 - Low	Assigned to	
Risk	Moderate		
Impact	3 - Low		
Short description	test change 0001		
Description			

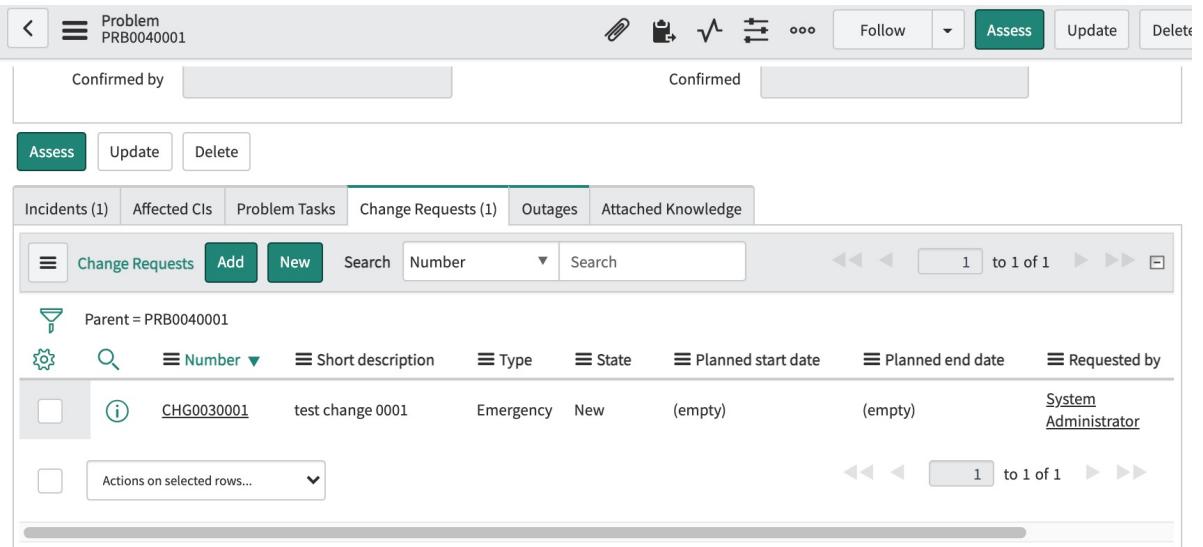
4. From incident A, create a new problem

1. From the global search, enter the incident number of incident A created earlier in this exercise
2. Right click on the form header and click "Create Problem"
3. This will create a problem record and relate it to incident A
4. Note the problem number
5. Look at the form to see where you can see the related incident number
(Hint: look at the related lists)

The screenshot shows the ServiceNow interface for creating a new problem from an incident. At the top, there is a context menu for an incident record titled 'INC0010001'. The 'Create Problem' option is highlighted in green. Below the menu, the 'Problem' form is displayed with the number 'PRB0040001'. The form includes fields for Number, State, First reported by, Impact, Category, Urgency, Service, Priority, Service offering, Configuration item, Assignment group, Assigned to, Problem statement, and Description. The 'New' tab is selected at the bottom of the form.

5. Link the emergency change request to the problem record

1. Open the problem record created on the previous slide
2. Scroll down to the related lists and on the Change Requests related list, click 'Add'
3. Search for the change number and click on "Add selected"
4. Notice that the change is now related to the problem record



6. Create a new iPhone X catalog item

1. Go to Service Catalog -> My Items (or Maintain Items) and click 'New'
2. Enter values for Name, Catalogs, Category and save the catalog item record
3. Note, you still have to add variables/variable sets to collect data from the requestor and a workflow/flow to fulfill the request
4. Alternatively, you can use the Catalog Builder feature to quickly create catalog items
5. Also, you can publish the iPhone X hardware model straight to the hardware catalog

The screenshot shows the 'Catalog Item' creation screen for an item named 'Order iPhone X'. The item is categorized under 'Hardware' and is marked as 'Active'. The 'Description' field contains the text 'Order iPhone X'. The interface includes tabs for Item Details, Process Engine, Picture, Pricing, and Portal Settings.

Catalog Item
Order iPhone X

Name: Order iPhone X

Catalogs: Service Catalog

Category: Hardware

State: -- None --

Checked out: -- None --

Owner: System Administrator

Application: Global

Active:

Short description: Order iPhone X

Description:

Order iPhone X

Item Details | Process Engine | Picture | Pricing | Portal Settings

7. Create a new knowledge article and publish it

1. Go to Knowledge -> Articles -> Create New

2. Save the knowledge article and click on 'Publish'

3. Open the article and check the Approvals related list

4. Impersonate the approver and approve the knowledge article from My Approvals

5. Check the article workflow after approval. The article should now be published

The screenshot shows the 'Knowledge' creation form with the following details:

- Number: KB0010001
- * Knowledge base: IT
- Category: (empty)
- Language: English
- Valid to: 2100-01-01
- Parent: (empty)
- Article type: HTML
- Workflow: Draft
- Source Task: (empty)
- Attachment link: (checkbox)
- Display attachments: (checkbox)
- * Short description: Sample knowledge 0001

The screenshot shows the 'Knowledge' edit form with the following details, indicating the article has been published:

- Number: KB0010001
- * Knowledge base: IT
- Category: (empty)
- Language: English
- Published: 2021-06-02
- Valid to: 2100-01-01
- Parent: (empty)
- Article type: HTML
- Workflow: Published
- Source Task: (empty)
- Attachment link: (checkbox)
- Display attachments: (checkbox)
- * Short description: Sample knowledge 0001

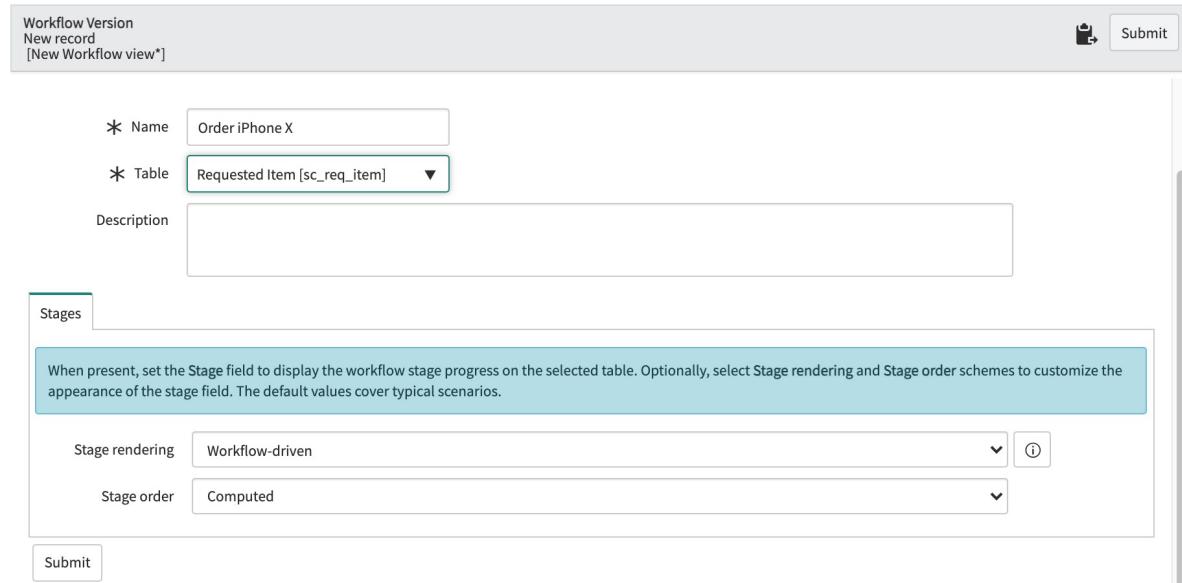
Section – 8 – Exercises



1. Create a new workflow for the catalog item you created in section 7

1. Go to Workflow -> Workflow Editor
2. Click "New Workflow"
3. Give the workflow a name

Workflow Version
New record
[New Workflow view*]

Submit

* Name: Order iPhone X

* Table: Requested Item [sc_req_item]

Description:

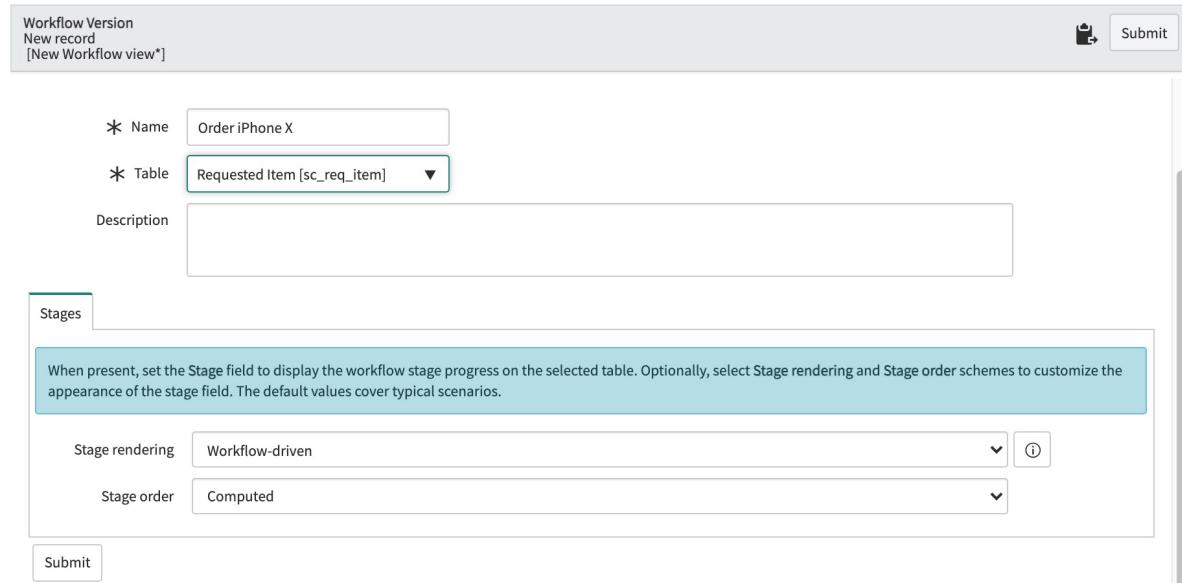
Stages

When present, set the Stage field to display the workflow stage progress on the selected table. Optionally, select Stage rendering and Stage order schemes to customize the appearance of the stage field. The default values cover typical scenarios.

Stage rendering: Workflow-driven

Stage order: Computed

Submit



2. Add the sc_req_item table to the workflow

1. Select Requested Item (sc_req_item) as the table on the workflow

Workflow Version
New record
[New Workflow view*]

* Name Order iPhone X

* Table Requested Item [sc_req_item] ▾

Description

Stages

When present, set the Stage field to display the workflow stage progress on the selected table. Optionally, select Stage rendering and Stage order schemes to customize the appearance of the stage field. The default values cover typical scenarios.

Stage rendering Workflow-driven

Stage order Computed



3. Add a condition so that this workflow is only ran when this item is selected

1. This question is incorrect
2. To use a workflow for a catalog item, the workflow should be published and added to the catalog item on the "Process Engine" section, 'Workflow' field

Catalog Item Order iPhone X

Name: Order iPhone X Application: Global Active:

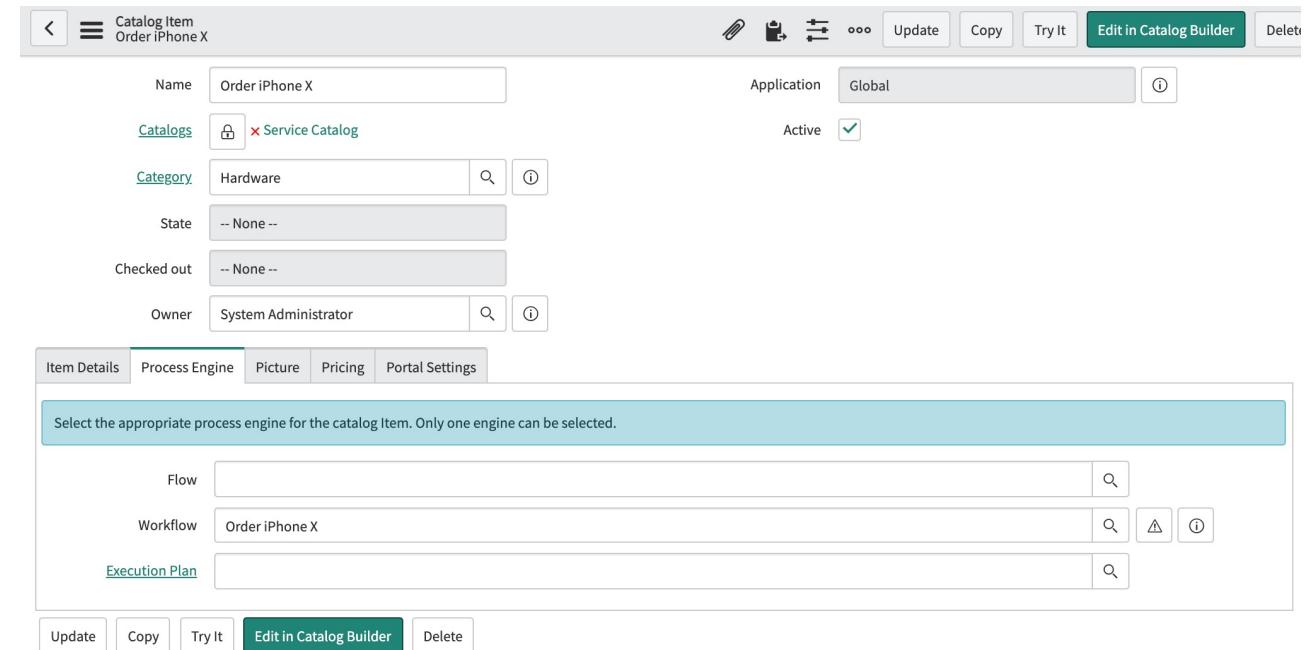
Catalogs: Service Catalog Category: Hardware State: -- None -- Checked out: -- None -- Owner: System Administrator

Item Details Process Engine Picture Pricing Portal Settings

Select the appropriate process engine for the catalog item. Only one engine can be selected.

Flow: Workflow: Order iPhone X Execution Plan:

Update Copy Try It Edit in Catalog Builder Delete



4. Add an approval and notification to the workflow

1. Add the "Approval – User" or "Approval – Group" activity to the canvas
2. Add "System Administrator" to the 'Users' field as the approver and click Submit
3. Add Approval Action activities to mark the request as approved or rejected
4. Make sure to connect the Begin activity with the new approval activity

Approvers

Specify the users whose approval will be requested. To edit this field, click the lock icon. To select specific users by name, use the lookup list record at runtime, click the tree icon. Each user will be assigned an individual approval record.

Users  System Administrator

Groups 

Check 'Advanced' to use a script for creating additional user approvals. When you check Advanced, a text box appears where you can enter a comma-separated list or an array of user and/or group ids.

Advanced

To   \${opened_by}

To (groups) 

Advanced

Message

In 'Subject' specify the text to appear in the message's subject line. In 'Message' specify the text of the message itself. To include the value of a field in the message body, place the cursor in the text where you want the field's value inserted. Then click the + icon next to Fields and select the field you want.

Subject Requested Item \${number} has been \${approval}

Message 

Requested Item \${number} has been \${approval}

- + Select variables:

 Fields

Continued...



5. Request the item and view the notification in the emails log

1. Search for the "Order iPhone X" catalog item from the platform or from the Service Portal and submit a request
2. As System Administrator, approve or reject the request from "My Approvals"
3. Go to System Logs -> Emails to find the email record

The top screenshot shows a confirmation message: "Thank you, your request has been submitted". It includes details: Order Placed: 2021-06-02 07:49:29, Request Number: [REQ0010001](#), Estimated Delivery Date of Complete Order: 2021-06-02.

Description	Delivery Date	Stage	Price (ea.)	Quantity	Total
Order iPhone X	2021-06-02	▶ ✓		1	

The bottom screenshot shows the "Email Requests" log. A preview window displays the email content: "Requested Item RITM0010001 has been Requested" and "Ref:MSC0000221_OwQMnirPfLfxLJxtmMRx". Below it, the "Originating Event and Notification" table shows a single event named "workflow.notification" with no notifications.

Event	Notification
workflow.notification	(empty)

Details for the email:
Subject: Requested Item RITM0010001 has been Approved
Recipients: admin@example.com

Section – 9 – Exercises

1. Create the Rotten Potatoes application in your personal instance

1. Use the Studio to quickly create a custom application
2. Name the application and add a new table named Game and add the columns as per Section 9.pdf
3. Also create the Game Review table with the columns as per Section 9.pdf

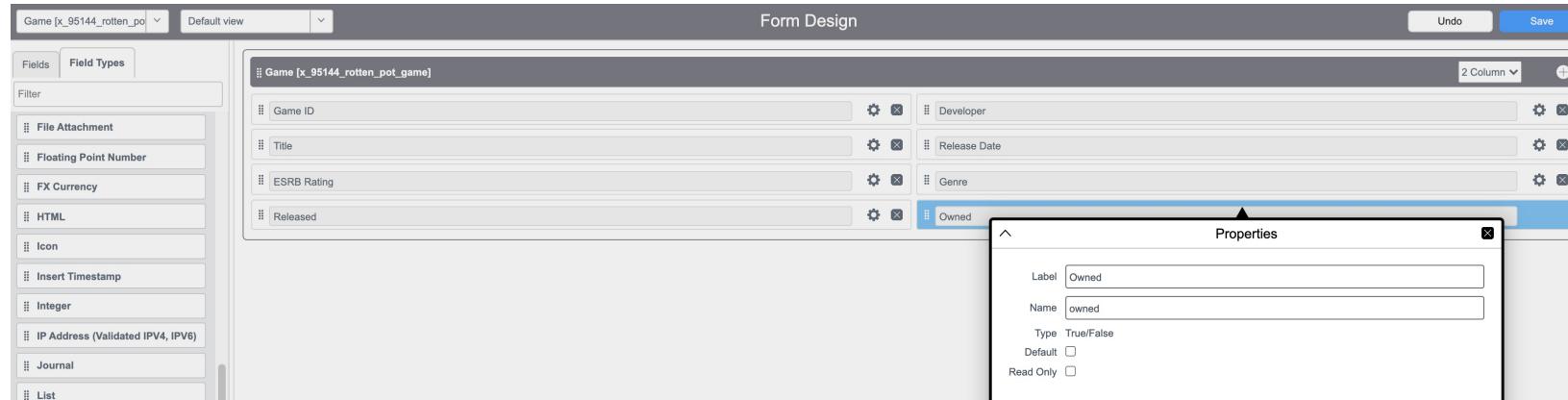
The screenshot shows the Studio interface with the following details:

- Top Bar:** STUDIO, File, Source Control, Window, Search.
- Left Sidebar (Application Explorer):** Filter navigator (Groups / Files[,Files]), Link VS Code via sn-scriptsync, Data Model, Tables (Game), Forms & UI, Forms (Game [Default view]), List Layouts (Game [Default view]), Access Control, Roles (x_95144_rotten_pot.game_user), Access Controls (x_95144_rotten_pot_game (read), x_95144_rotten_pot_game (create), x_95144_rotten_pot_game (delete), x_95144_rotten_pot_game (write)), Navigation, Application Menus (Game), Modules (Games), Application Menus (Mobile) (Game), Modules (Mobile) (Games).
- Central Area:** Data Model > Table > Game. A tooltip says: "A table is a collection of records in the database. Each record corresponds to a row in a table, and each field".
 - Table Definition:** Label: Game, Name: x_95144_rotten_pot_game.
 - Table Columns:** New, Search for text, Search.
 - Dictionary Entries:**

Column label	Type	Reference
Developer	String	(empty)
ESRB Rating	Choice	(empty)
Game ID	String	(empty)

2. Add an Owned true/false field to the video game table

1. You can add the field either through the Studio or Form Designer or Form Layout
2. Add the Owned (true/false) field and add it to the form layout



3. If a review is given 1 or 2 stars, require the comments field

1. Create a UI Policy on the Game Review table
2. Add condition as Stars is one of 1 Star or 2 Stars
3. Add a UI Policy action to make the Comments mandatory

Table: Game Review [x_95144_rotten_pot_game_review] Application: Rotten Potatoes Active:

* Short description: Make comments mandatory on 1 star or 2 stars Order: 100

When to Apply: Script

Conditions: Add Filter Condition, Add "OR" Clause
Stars: is one of 1 Star, 2 Stars, 3 Stars, 4 Stars

AND OR X

If selected, the UI Policy applies to all form views; otherwise the UI Policy is view-specific

Apply the UI policy actions when the form is loaded and when the user changes values on the form

Global: On load:

Reverse the effects of the UI policy actions when the Conditions evaluate to false

Tables that extend the specified Table inherit this UI Policy

Inherit:

Submit

UI policy: Make comments mandatory on 1 star or 2 stars Application: Rotten Potatoes

Table: Game Review [x_95144_rotten_pot_game_review] Mandatory: True

* Field name: Comments Visible: Leave alone

Read only: Leave alone

Clear the field value:

4. Add an Image field to the video game table

1. Open the Game table config
2. Add a new field/dictionary of type 'Image'
3. Add the field to the form layout

Game
GAME00000105
No file chosen

Game ID: GAME00000105

Title:

ESRB Rating: -- None --

Released:

Developer:

Release Date:

Genre: -- None --

Owned:

Image: [Update][Delete]

Update Delete

Section – 9 – Advanced – Exercises

1. Create a workflow for the video game table which will execute only for video games that have not yet been released, and send a notification to a review group once the video game release date is 1 week after today's date

1. Open the workflow editor and create a new workflow on the Game table
2. Add condition so that workflow only runs on games that have not yet been released (Released is False)
3. Add a wait for condition activity to wait until 7 days before the release date (Condition: Release date – relative – before – 7 days – from now)
4. Add a notification activity to send an email to the review group
5. Remember to publish the workflow

